IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 31 May 2023, accepted 21 June 2023, date of publication 26 June 2023, date of current version 3 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3289751

RESEARCH ARTICLE

An Intelligent Hybrid Neural Collaborative Filtering Approach for True Recommendations

MUHAMMAD IBRAHIM[1], IMRAN SARWAR BAJWA[1], NADEEM SARWAR[2], FAHIMA HAJJEJ[3], AND HESHAM A. SAKR[4]

[1]Department of Computer Science, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[2]Department of Computer Science, Bahria University Lahore Campus, Lahore 54600, Pakistan
[3]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[4]Communications and Electronics Department, Nile Higher Institute for Engineering and Technology, Mansoura 35511, Egypt

Corresponding author: Nadeem Sarwar (Nsarwar.bulc@Bahria.edu.pk)

This work was supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R236), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT Recommendation services become a critical and hot research topic for researchers. A recommendation agent that automatically suggests products to users according to their tastes or preferences instead of wandering in a huge corpus for a product. Social data such as Reviews play an important role in the recommendation of products. Improvement was achieved by neural network methods for capturing user and product information from a short text. However, such approaches do not fairly and efficiently incorporate users' preferences and product characteristics. We are proposing the novel Hybrid Neural Collaborative Filtering (HNCF) model that combines deep learning capabilities and deep interaction modelling for recommender systems with a rating matrix. To overcome the cold start problem, we use the new overall rating by aggregating the Deep Multivariate Rating DMR (votes, likes, stars and sentiment scores of reviews) from different external data sources because different sites have different rating scores about the same product. The propose novel model consists of four major modules HUAPA-DCF+NSC+DMR (Hierarchical User Attention and Hierarchical Product Attention, Deep Collaborative Filtering, Neural Sentiment Classifier, and Deep Multivariant Rating) to solve the addressed problems. Initially, the HUAPA module is based on BiLSTM's hierarchical user attention (HUA) and hierarchical product attention (HPA) to embed the user preferences and product characteristics respectively. Further, these are combined nonlinear representations and fed as input to the interaction module. Secondly, the deep collaborative filtering module is implemented to find the explicit interaction between user and product. Thirdly, NSC module will extract the user's semantic about products by incorporating the user's preferences and product characteristics. Finally, the module uses explicit information (Deep Multivariant Rating) to the maximum extent for final classification. Experimental results demonstration that our novel model is outperforming than state-of-the-art at IMDb, Yelp2013 and Yelp2014 datasets for the true recommendation of top-n products using HNCF (HUAPA+DCF+NSC+DMR) to increase the accuracy, confidence, and trust on recommendation services.

INDEX TERMS Hierarchal user attention product attention (HUAPA), deep collaborative filtering (DCF), neural sentiment classification (NSC), deep multivariate rating (DMR), recommendation system (RS).

I. INTRODUCTION

"*Recommendation systems*" (RSs) are the software agents that automatically suggest the products according to user interests or preferences [1]. Recommendation system services are based on "*Arterial Intelligence*"(AI), "*Natural Language Processing*"(NLP) and "*Machine Leering*" (ML) or "*Deep Learning*" (DL) that provide the empirical solutions and got great importance in facilitating the decision-making process across the various real-world

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

applications such as e-commerce [2], e-healthcare [3], e-purchasing [4], e-cinema [5], e-game [6] and so many others.

Users may express their opinions in the form of ratings, votes, likes, and reviews about products on social media. Several user behaviours such as searching, spending time, watching [7] and history sequence of interaction with products are "*Implicit Feedback*" while likes, votes and stars to a product or features of products are "*Explicit Feedback*" [8] [9]. So, the rating that is explicitly given by users is more reliable than implicitly. That is the reason we use explicit feedback. Some of the recommenders use "*Univariant*" (IMDB, Rotten Tomatoes, Netflix) while another's using "*bi-variant*" (Facebook, Fandango) ratings are not suitable and significant for measuring the popularity scores of products for ranking because different sites have different ratings about the same product.[1] therefore, there is a need to evaluate the true weight of products for top-n ranking by using the multivariate (likes, votes, stars and sentiment scores) from different sites [10], [11].

Usually, recommendation systems are based on "*Content-based filtering*" (CBF), "*collaborative based filtering"* (CF) or "*hybrid-based filtering"* (HF) [12], [13]. CF techniques have been extensively researched for personalized recommendations based on the interaction between users and products instead of using the previous history or knowledge about users and products. CF techniques simple and efficient but suffer with cold start problem, accuracy of prediction and do not have maturity to capture the complex relationship (interaction) between user and product. Most of conventional CF techniques are based on the Factorization of Matrix [13] in which users and products are characterized by Latent Factors (LF) derived from the rating matrix for user product. Traditionally, LFM models in CF [14], user's choices for a product is frequently predicted with a linear kernel, like a "*dot product*" of its Latent Factors, but the complex structure of user product interaction is not handled effectively. currently deep learning-based recommendation systems [15] overcome the limitations of traditional recommendation system approaches, such as complex user preferences, product features and their relationships itself to gain high performance in recommendation. Mostly auxiliary information is explored by deep learning for recommendation, such as modeling the product features [16], [17]. Recent research is mainly continuing to use conventional MF-based methods for the user-product rating matrix. Firstly, "*Restricted Boltz Mann Machines*" (RBMs) [18] using the two-layer neural network instead of deep learning for user-product rating matrix is modeled to achieve more accurate results over conventional approaches. Another recent method, "*Collaborative Denoising Auto-Encoder*" (CDAE) [19] is seemed to be primarily design the rating prediction model by implementing the one hidden layer between the input and output layer. while in "*Neural Collaborative Filtering*" (NCF) [20] is designed the interaction model by employing the multi-layer perceptron's, However, this does not analyze any preferences of users and characteristics of products which appear to be useful in maximizing CF performance in recommendation. rather than explicit feedback on results, NCF and CDAE use only implicit feedback for recommendations. "*Deep Matrix Factorization*" (DMF) [21] is designed for the user-product rating matrix in which features of users and products is mapped into a low-dimensional with nonlinear representations by using the deep neural networks in which user-product interactions is computed by an inner product and as the LFM [14], apply in the similar way as linear kernel (i.e., dot function). It is assumed that the effectiveness can be gained by capturing the users and product information as well as both non-linear and non-trivial user-product relationship with multi-layer projection [22].

Recommendation system based on "*Convolutional Neural Networks*" (CNNs) is used to extract the product features from short text or context information using ontologies as axillary information [16], On the other hand, the recommender system based on "*Recurrent Neural Networks*" (RNNs) is used to explore the sequential features and temporal dynamics of ratings [23]. but they only focus on the content of text typically while users and products have crucial influences for sentiment classification. For example, different users may express their different emotional intensity by the same words, lenient users use "good" words for ordinary products while potential users may use this word for excellent behavior about the product. Likewise, review ratings may also affect a product's characteristics. Such that higher quality products lead to higher ratings while low-quality product leads to low rating. Reference [24] enables the Sentiments Classification by incorporating user and product details at word level to reflect preferential matrix and vector distribution for each user and product of CNN Sentiment Classification. It has some improvement but it represents the information of users and product at the word-level instead of the document-level.

User and Product information are incorporated to overcome the issues [25] by using the attention mechanism in a better way. In which user and product embedding are used to present the user's preferences and product characteristics respectively after that user-product specific document representation is achieved by using the joint "*User Product Attention*" (UPA) mechanism in hierarchically manner. But user and product influence on reviews differently. So joint user product attention mechanism seems to unreasonable for example the review about the movie "The story of the movie is very lovely and romantic, I like that movie.", "like" represent the user semantics about movie while the words "lovely and romantic" represents the features of movie, it means reviews are subjective when they are user-centered and reviews are objective when they are product-centered. So,

[1] https://www.freecodecamp.org/news/whose-reviews-should-you-trust-IMDb-rotten-tomatoes-Metacritic-or-fandango-7d1010c6cf19/

TABLE 1. Differences and shortcomings of different famous models.

References	HUAPA	HAN	DCF	NSC	Discrete Rating	Continuous Rating	DMR	Favorability Status	Emotion	Popularity Status	Medals
[21] DMF	×	×	✓	×	✓	×	×	×	×	×	×
[20] NCF	×	×	✓	×	✓	×	×	×	×	×	×
[28] J-NCF	✓	×	✓	×	✓	×	×	×	×	×	×
[10] CF+SC+MR	×	×	×	×	✓	✓	×	✓	×	✓	×
[11]CF+NSC+MR	×	✓	×	✓	✓	✓	×	✓	×	✓	×
[29] HDCF	×	✓	✓	✓	✓	✓	✓	✓	×	✓	×
[Proposed-Model] HNCF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

to overcome the issues, HUAPA is introduced by [26] that represents user attention and product mechanism in hierarchical way separately to incorporate the information of user and product at the word and sentence level in a more accurate way. while HUAPA outstanding performance than [27] but it does not implement the user-product interaction model for collaborative filtering in recommendations. J-NCF [28] state-of-the-art proposed collaborative model that is based on the extraction of features and user-product interaction modules using the MLP that would be jointly trained to form a unified deep learning structure consisting of two neural networks for the recommendation. But this model "*Joint Neural Collaborative Filtering*" (J-NCF) does not use the Hierarchical User Attention and Product Attention mechanism in which user's preferences and product characteristics are incorporated for document representation that further is used for interaction modelling between user and product for more accurate prediction.

As we previously addressed and shown in the **Table 1**, the major short coming and issues in different recommendation systems, the true popularity of the product, users and product documentation representation, and sentiment class extraction from reviews by using the incorporation of user's preferences and product characteristics are the shortcoming of J-NCF, NCF while collaborative filtering is a shortcoming of HUAPA. So, our novel proposed model HNCF provides the empirical solution to overcome the limitations of HUAPA, NCF, J-NCF and HDCF by integrating HUAPA, DCF, NSC and DMR. the proposed model will learn the user's presence and product characteristics from reviews and encode the reviews into two document representation using the Hierarchal attention of user and product, further these are concatenated and further fed as input to interaction model and sentiment classifier. sentiment classifier predicts the favorability class of reviews. fussing the sentiment class of reviews that are given by a user about a product with other rating votes, likes, and stars from other sites are given by others users to that product to achieve significant accuracy in the popularity of the product. On the other hand, "Comments" are helpful for providing more detailed feedback and insights about a particular product, service, or post instead of voted, likes or stars. rating likes, stars, and votes provide a quick and simple way for users to express their overall opinion or preference that may be biased, while comments provide a more detailed and nuanced form of feedback. The multivariant rating is used instead of uni (Discrete) or bi (Continuous) rating because some site has rating about a product very high, some sites have rating for the same product very low or some sites have a normal rating for the same product, so these ratings may lead to confusion for users to make decision either this product is good, bad or normal. That is the reason to introduce a new overall rating called Multivariant Rating, the multiple rating from different sites for the same product to generate the multivariant rating for True Ranking to make a true recommendation. Obliviously system will suggest the top product. If one site has a high rating for low standard product, this site will be placed in top ranked product and obviously will recommend the low product and if another site has a low rating for high standard product obviously high standard will not be placed in top rank product, obviously, it will not be recommended. The existing may recommend the product according to preferences but they did not control the standard and fairness. So that is the reason to choose the multivariant. user document and product document representation will be frequent used for Training and classification that is the reason we used a combined strategy. So, we can say that our contribution to the state-of-the-art recommendation systems by better utilization of NSC, NCF, HUAPA and DMR.

- **RC1:** Our innovative propose HNCF model shows outstanding results as compared to state-of-the-art baselines CF techniques for recommender systems.
- **RC2:** HUA and HPA units in combined form fed into deep interaction networks (NCF) helpful to increase the optimization of collaborative filtering for recommendation.
- **RC3:** fussing continuous rating (reviews) with discrete rating (stars, votes, likes) to generate the new overall rating is called as Deep Multivariant Rating (DMR) for improving the accuracy and generate the true ranking of the product popularity in recommendation system.

The document is divided into 7 sections, the first section introduces and raises the problem related to our work, the second section represents the background related to deep machine learning, the third section represents the related work to recommendation techniques, the fourth section briefly explores our proposed model, the fifth section represent the experimental setup, the sixth section represents and evaluates the results, the seventh section represent the conclusion and discussion and the last section represent the references.

II. BACKGROUND

A. ARTIFICIAL NEURAL NETWORKS

ANNs is encouraged by the human nervous system in which computation is performed by a model that consists of numerous computational unit that take inputs and generate outputs based on their well-defined activation functions. $y_i = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right)$ in which different inputs $(x_1, x_2 \ldots x_n)$ to the single perceptron is fed and multiplied by their respective weights $(w_1, w_2, \ldots w_n)$ that indicate the strength of the specific unit. Whereas b is the bias and σ is Activation function that decides whether to activate or not neurons are applied on the weighted sum to generate output y that may be o or 1. The activation function may be the threshold, sigmoid, hyperbolic Tangent or Rectified Linear, etc. here some important machine learning techniques are discussed for natural language processing that is used in the recommendation system [30].

B. MULTI-LAYER PERCEPTRON

ANNs are encouraged by the human nervous system in which computation is performed by a model that consists of numerous computational units that take inputs and generate outputs based on their well-defined activation functions. $y_i = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right)$ in which different inputs $(x_1, x_2 \ldots x_n)$ to the single perceptron is fed and multiplied by their respective weights $(w_1, w_2, \ldots w_n)$ that indicate the strength of the specific unit. Whereas b is the bias and σ is Activation function that decides whether to activate or not neurons is applied on the weighted sum to generate output y that may be o or 1. The activation function may be the threshold, sigmoid, hyperbolic Tangent or Rectified Linear, etc. here some important machine learning techniques are discussed for natural language processing that is used in the recommendation system [30].

C. RECURRENT NEURAL NETWORK

Typically, in brought, RNN is a chain of numerous copies of the equivalent static network from the input, sequencing every copy of Networks working on the single time step. CNN's deals better with grids, local or short patterns but ignored the long-term dependencies while RNNS specially deals with the sequences and dependencies. At high-level RNN is fed with a set of input vectors $x = (x_1, x_2, \ldots x_t)$ and initialized with a hidden state h_0 to all zeroes and returns ordered list of hidden states $h = (h_1, h_2, \ldots h_{t-1}, h_t)$. And also produce an ordered list of output vectors y $= (y_1, y_2, y_3 \ldots y_T)$. Let us each timestep is represented as $(x_i \in x_1, x_2, \ldots x_n)$. the nonlinear function f is used to calculate the hidden state h_t at timestep i In simplest form represented as: $h_t = f(h_{t-1} + x_t)$. The output vectors may input other RNN units in deep RNN that are stacked vertically [31].

D. LONG SHORT-TERM MEMORY

In practicality whenever researcher uses the RNNs they usually use the LSTM [32], LSTM is the variation of RNN that deals such long-term dependencies such as time series or in the natural language processing sequence of the word, sentence and documents because of its outstanding performance on sequence modelling, particularly for long documents. The cell state is as explicit memory and hidden state that is computed by interacting layers these layers have the capability to the memorized state over a long period or not for specified information about the previous elements of the sequence to overcome the long-term dependencies. These make the major difference between the RNN unit and the LSTM units. The state flow in the LSTM unit is controlled and protected by three gates. At each "t" time step, x_t as given input vector, c_t the current state and h_t as the hidden state can be changed with c_{t-1} previous cell state and h_{t-1} hidden state.

$$\mathrm{X} = \left(\frac{h_{t-1}}{x_t}\right) \tag{1}$$

$$\text{Input gate at time } t: i_t = \sigma(W_t.X + b_t) \tag{2}$$

$$\text{Forget gate at time } t: f_t = \sigma(W_f.X + b_f) \tag{3}$$

$$\text{Candidate state at time } t: \tilde{C}_t = \tanh(W_c[h_{t-1}; X] + b_c) \tag{4}$$

$$\text{Final memory cell}: C_t = f_t \odot c_{t-1} + i_t \odot \tilde{C}_t \tag{5}$$

$$\text{Output gate}: o_t = \sigma(W_o.X + b_o) \tag{6}$$

$$\text{Hidden state}: h_t = o_t \odot \tanh(C_t) \tag{7}$$

$$\begin{bmatrix} i_t \\ f_t \\ o_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix} (W.[h_{t-1}; X_t] + b) \tag{8}$$

Activation function as σ in [0,1] and gate activations are represented as i, f, o. While element-wise multiplication is represented as $\odot$. At i_t as input gate, which information should remember or get ride that decision made by sigmoid function σ and produces 1 means should remember while 0 means should forget in the cell state. Which value should update is decided by Sigmoid function at the input gate and Tanh function signifies $\tilde{C}_t$ the new candidate value information? which part of information should produce is decided by sigmoid function at the output gate.

E. GATED RECURRENT UNIT

GRU is a simplified form of LSTM that consist of "*Reset Gate*" and "*Update Gate*" without using the separate memory to track the state of sequences [33]. The new state is computed at time t by GRU as follows.

$$h_t = 1 - z_t \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{9}$$

The linear interpolation between the current new state $\tilde{h}_t$ and previous state h_{t-1} is computed by using new sequence information. How much information is kept or new information added is decided by gate z_t as:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{10}$$

$$\tilde{h}_t = \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \tag{11}$$

$$r_t = \sigma(W_h x_t + U_h h_{t-1} + b_r) \tag{12}$$

Here the sequence of the vector at time t is represented by x_t while candidate state $\tilde{h}_t$ is computed traditionally RNN way. Past state's contribution to the candidate state is managed by gate reset r_t.

F. ATTENTION MECHANISM

The self-attention mechanism derives into play twofold: at the word-level and the sentence-level. The reasons behind this technique are: the first reason, it matches the natural hierarchical structure of documents (words, sentences and documents). The second reason is in computing the encoding of the document, allows the model to first determine which words are important in each sentence, and then, which sentences are important overall. Through being able to reweigh the word attentional coefficients by the sentence attentional coefficients, the model captures the fact that a given instance of the word may be very important when found in a given sentence, but another instance of the same word may not be that important when founding in another sentence. References [27] and [34] explore hierarchical attention mechanisms to select informative words or sentences for the semantics of the document.

The input sequence is embedded into a vector by encoder while some output from the input vector is generated by the decoder. Furthermore [35], to increase the capacity of input information accessible to the network bidirectional LSTM is used to model text semantic for the feedforward and backward. The LSTM forward scans the sequence from x_1 to x_t while LSTM backward scan sequence from x_t to x_1 for sequence vectors $(x_1, x_2, \ldots x_t)$. Furthermore, the forward $\overrightarrow{h_t}$ and backward $\overleftarrow{h_t}$ hidden states are concatenated $h_t = \left[\overrightarrow{h_t}\ ;\ \overleftarrow{h_t}\right]$. Here h_t is summarized information of centered around x_t while concatenation is represented as ";".

"Modeling language" or "Sequence Labeling" is a particular classification case in which the training of the model is done to predict the next character or word or sentence in the document. The output vector provides the probability distribution of y_t over these characters/words/sentence/ in the vocabulary at-every time step t, conditioned on the preceding characters/words/sentence in the sequence, i.e. $\mathrm{P}(y_t \mid y_1, y_2, y_3 \ldots y_T)$. the product of all conditional probabilities gives the probability of a complete sequence at the test time such as:

$$\mathrm{P(Y)} = \prod_{t=1}^{T} \mathrm{P}(y_t \mid y_1, y_2, y_3 \ldots y_T) \tag{13}$$

Here next word/sentence y'_t can be predicted by previously predicted word/sentence. Where $\mathrm{y} = (y_1, y_2, y_3 \ldots y_T)$ and similarly, the next word/sentence y'_t can be predicted by previously predicted word/sentence over the context word/sentence using conditional probability.

$$\mathrm{P(Y)} = \prod_{t=1}^{T} \mathrm{P}(y_t \mid \{y_1, y_2, y_3 \ldots y_T\}, c) \tag{14}$$

$$\mathrm{P(Y)} = \prod_{t=1}^{T} \mathrm{P}(y_t \mid \{y_1, y_2, y_3 \ldots y_T\}, c) = g(y_{t-1}, s_t, c) \tag{15}$$

c is a vector generated from sequence of the hidden states while Hidden state $\mathrm{h}_t \in R^n$ at time t. Ordered list of hidden states: $h = (h_1, h_2, \ldots h_{t-1}, h_t)$ and Vector of hidden state $c = \sigma(h)$ While σ is a non-linear function.

$$\mathrm{P(Y)} = \prod_{i=1}^{T} \mathrm{P}(y_i \mid \{y_1, y_2, y_3 \ldots y_i\}, c) = g(y_{i-1}, s_i, c_i) \tag{16}$$

Here s_i RNN hidden state for time t_i is computed as $s_i = g(y_{i-1}, s_{i-1}, c_i)$. here the conditional probability is computed on a distinct context vector c_i for each target word/sentence y_i.

$$c_i = \sum_{j=1}^{T} a_{ij} h_i \tag{17}$$

Input sentence mapped by encoder based on the context vector c_i is computed as a weighted sum of these annotations h_i:

whereas

$$a_{ij} = \frac{\exp\left(e\left(h_{ij}\right)\right)}{\sum_{k=1}^{T} \exp\left(\mathrm{e}\left(h_{ik}\right)\right)} \text{ and}$$
$$e_{ij} = a(s_{i-1}, h_j) \tag{18}$$

Here a_{ij} is the weight of each annotation h_j. e_{ij} is an alignment model that scores how well the inputs around position j and the output at position i match. The probability a_{ij}, or its associated energy e_{ij}, reflects the importance of the annotation h_j with respect to the previously hidden state s_{i-1} in deciding the next state s_i and generating y_i.

1) HIERARCHAL ATTENTION NETWORK

It has a hierarchal document structure and two levels of attention mechanism that enable it to attend differently to less and more important features during the construction of document representation. So, [36] "*Hierarchal Attention Networks*" (HAN) is based on two major components, one is the word sequence encoder and word-level attention whereas the other component is the sentence encoder and the attention at the sentence level.

a: WORD LEVEL ATTENTION

Given Words [37] within the sentence are encoded to vector by embedding matrix utilizing bidirectional GRU [35] which summarizes the information for words from both directions to incorporate the contextual information. Given w_{ij} embedding is achieved through hidden forward state and Hidden Backward state $\overrightarrow{h_{ij}} = \overrightarrow{GRU}(w_{ij})$ and $\overleftarrow{h_{ij}} = \overleftarrow{GRU}(w_{ij})$ respectively. The entire information of sentence centered around w_{ij} is summarized by $h_{ij} = \left[\overrightarrow{h_{ij}}\ ;\ \overleftarrow{h_{ij}}\right]$.

At word-level attention, the important meaning of words is extracted that incorporate into the representation of the sentence because not all words equally contribute. These informative words are aggregated to form the sentence vector.

$$u_{ij} = \tanh(W_w h_{ij} + b_w) \tag{19}$$

$$\alpha_{ij} = \frac{\exp\left(\mathrm{u}_{ij}^{\mathrm{T}} u_w\right)}{\sum_{j=1}^{l_i} \exp\left(\mathrm{u}_{ij}^{\mathrm{T}} u_w\right)} \tag{20}$$

$$\mathrm{s}_i = \sum\nolimits_{j=1}^{l_i} a_{ij} h_{ij} \tag{21}$$

Firstly, here u_{ij} is obtained as the hidden representation h_{ij} by feeding the annotation of word h_{ij} via one-layer. The u_{ij} Similarity with a context variable at word-level u_w and normalized weight of importance α_{ij} is gained by SoftMax function.

b: SENTENCE LEVEL ATTENTION

In a similar way to word embedding, a document vector is obtained by giving the sentence vectors s_i using Bidirectional GRU. The embedding of the sentence is obtained by concatenation of forwarding hidden state $\overrightarrow{h_i} = \overrightarrow{GRU}(s_i)$ and backward hidden state $\overleftarrow{h_i} = \overleftarrow{GRU}(s_i)$. The neighbor sentences around sentence s_i are summarized by $h_i = \left[\overrightarrow{h_i}\ ;\ \overleftarrow{h_i}\right]$ while it focuses on sentence i. Sentence attention is similar to the word attention mechanism to represent context vectors at the sentence level u_i for measuring the sentence's importance.

$$u_i = \tanh(W_s h_i + b_s) \tag{22}$$

$$\alpha_i = \frac{\exp\left(\mathrm{u}_i^{\mathrm{T}} u_s\right)}{\sum_{k=1}^{n} \exp\left(\mathrm{u}_i^{\mathrm{T}} u_s\right)} \tag{23}$$

$$v = \sum\nolimits_{i=1}^{n} a_i h_i \tag{24}$$

The information of sentences in a document is summarized by document vector representation v. The initialization of the context vector is randomized and during the training process jointly learned.

III. RELATED WORK

A. CONVENTIONAL RECOMMENDATION SYSTEMS

The goal of the RS is to predict the top-n product that is more likely to the scheme (users-products) with the highest popularity [38]. The majority of the recommender system has been focused on CF [12] in which recommendations are based on the previous history or past behavior (rating) instead of domain knowledge [39]. CF techniques categorize into two "*Neighborhoods*" based [40] and "*Latent Factor*" based [14]. Neighborhood models are based on similarities of users or products. For example, similarities are measured on the bases of a similar rating among the sets of products or users. While latent factor techniques use the vector model for users and products as latent factor space that reduce the number of hidden factors. User and product ratings are calculated by estimating the inner-product between related vectors of the latent factor. Neighborhood-based models are designed on similarities between user and userusually estimate a user's rating based on the ratings given by other identical users about that product. While the user preference for a product based on his/her ratings to similar products is computed by "*Product-Product Similarity*". The correlation between product p_i and product p_j is calculated in a similar way as users appear to rate products p_i and p_j is usually based on either the "*Cosine Similarity*", or the "*Coefficient of Pearson Correlation*" [40]. To consider the "k" products rated by the user that are most likely to the product during the predicting the r_{up} rating to enhance the neighborhood models [41] by "*K-Nearest Neighbors*" (KNN). while low correlated to target products are discarded in KNN models to improve the accuracy by decreasing the noise. So, neighborhood models are similar to KNN product-product model for user personalization that is different from the user-product model [40] therefore we concentrate on latent factor models.

Typically, most of the research-based on the factorization of the user-product rating matrix [14] with "*Singular Value Decomposition*" (SVD) which uses the "*Inner Product*" between both low rank matrices, one for user factors and another one for product factors. The user's preferences are generated as follows:

$\hat{y}_{up}$=(b^{up} + (d^u.d^p)^T). here d^u and d^p denotes the inner product of factors user and product respectively while b^{up} the bias. Traditional SVD is unable to define the unknown (missing) rating. Most of the researcher tries to overcome the problem by baseline estimation [40] but Which results in a dense matrix of user ratings and complex factorization in terms of infeasible computation. Recently objective functions are used to minimise the prediction error to learn factor vectors directly on unknown ratings for avoiding overfitting [42]. Usually, SVD approaches are used for newly users after giving their rating to particular products without rebuilding the parameters in the models to minimize the objective function by using gradient descent. So, SVD using the current rating provides recommendations for new users. "*Matrix factorization*" (MF)model performance is improved by incorporating of Explicit and implicit feedback SVD++ [14]. That is the reason for sparse rating matrices by using the typical MF methods with complex computational costs for the decomposition of the rating matrix.

Mostly conventional RSs are implemented through "*linear kernel*" to design the interactions between user and product using an "*inner product*" vector of users and products. Linear functions may not have been able to provide a significant information of the user characteristics (products) and user-product interactions: earlier work has shown that nonlinearities have potential benefits with systematic studies in maximizing the performance of RSs [19], [43], [44].

B. RECOMMENDATION SYSTEMS BASED ON DEEP LEARNING

RSs that are based on "*Deep Learning*" are classified into two brought categories as "*Single Neural Network*" (SNN) and integrational models. In a single neural network, RBM [18] is the earliest neural recommender system that models the tabular data explicit rating of movies by using the two-layer undirected graph. RBM focuses on the prediction of ratings not on top-n recommendations while its loss function is used only for known ratings [19]. Rating prediction is computed by "*Auto-Encoder*" (AutoRec) [44] which uses the loss function only for known ratings, which means it is not good for providing accuracy for top-n recommendations. Auto-encoder fails for generalization of unknown rating data, auto-encoder is prevented from learning the identity function by using the Denoising autoencoders [43] to learn from corrupted inputs intentionally. CDAE [19] uses implicit feedback that is partially observed as inputs. Unlike our model, DAEs and CDAEs use a product-product model for personalizing recommendation represented by ratings of products given by a user [40] and product values are decoded by learning the user's representation of outputs. Unlike previous models, our model is a type of user-product model in which users and product representation are learned first and then measures in the correlation.

The state-of-the-art NCF implements the multi-layer perceptron to design the uses-products interaction model that represents the non-linear relationship between users and products [20]. However, user and product representation are initialized in a limited manner by using the one-hot vector for users and products. While j-NCF [28] uses the two neural networks DF and DI for users and product representation respectively that are further concatenated for the input of the interaction model. While our proposed model uses the same sense in a better way because user attention and product attention in a hierarchical manner capture the users and product features and their relationship more accurately than NCF and JNCF. "*Cross Domain Content Boosted Collaborative Neural Networks*" (CCCFNet) [45] based on a dual network one for users and another for products using the content information to explore the user's preferences and product features, further to compute the user-product interaction with dot product in the last layer. Faultlessly in DeepFM [46], The integration of the factorization machine and multi-layer perceptron is modelled as end-to-end uses the content information for low order interactions using a factorization machine while using deep neural networks for higher-order feature interaction.

Unlike DeepFM, our proposed model adopts the rating information from reviews or short text to explore reliably both information of users and product by using the hierarchical attention mechanism. "*Collaborative Deep Learning*" (CDL) [17] proposed the deep integration model based on the "*Hierarchical Bayesian Model*" in which DAEs stack are integrated into conventional probabilistic MF. It has two deviations from our propose model, first one is the deep features representations of the product are extracted from content information and another one is to model the user-product relationships via the "*dot product*" of vectors of user and product using linear kernel. Another popular model for integration is "*Deep Cooperative Neural Networks*" (DeepCoNNs) [22] which uses convolutional neural networks to extract the behavior of users and characteristics of products from reviews. It applies the factorization machine for interactions between users and products from predictions of rating to overcome the problem of sparsity. Another integration model, "*Deep Matrix Factorization*" (DMF) [21] is based on a DNNs that transforms the users and product feature matrix into a space of low dimensional that follows the LFM which computes the interaction between user and product by using "inner product" of user and product. Unlike DMF, we adopt the multilayer perceptrons to model the user-product interaction using the concatenation of user and product features as input that is extracted by using the HUAPA to improve the accuracy more expressively. As we addressed the previous work limitations, our novel proposed work is the empirical solution for that problem.

IV. THE PROPOSED FRAMEWORK

A. OVERVIEW OF THE ARCHITECTURE

The propose novel model is based on four major components, user and product document embedding by using HUAPA, sentiment classification of reviews about product, Neural Collaborative Filtering used for User-Product Interaction using MLP (three-layer perceptron's) and ranking the users or products using multivariant rating. HUAPA is used for encoding the information of the user and product by incorporating the preferences of a user and the characteristics of product. HUAPA uses BiLSTM for word and sentence level encoding of the reviews into two views, hierarchical user attention view and hierarchical product attention view respectively, further theses are combined to get explicit document representation. Further this document representation is fed into interaction module to extract the interaction between users and products. Combined document representation is also fed into Deep User-Product Interaction module in which Sentiment classification is performed. The sentiment classifier transforms the reviews (continuous sentiment) into the real number (discrete sentiment) and classifies the discrete sentiment into five classes by labeling the favorability status. Multivariant rating is computed by combining the votes, stars, likes, and sentiment scores to generate the popularity status of product labelling with the medal for user convenience

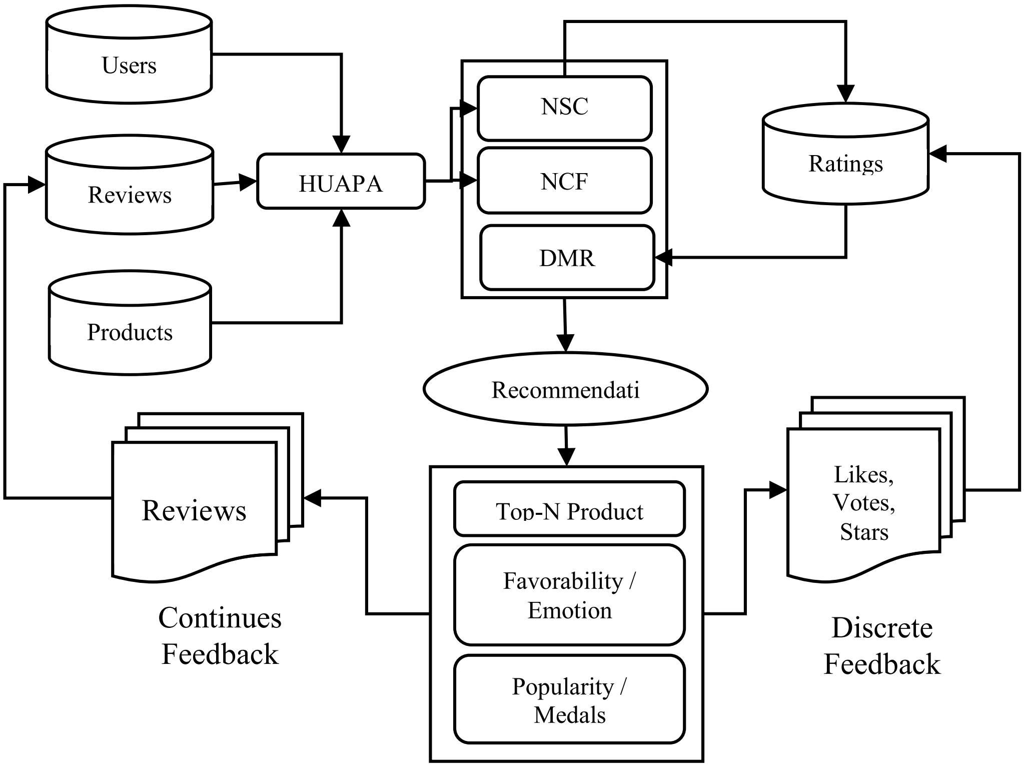

FIGURE 1. A graphical abstract of neural collaborative filtering approach for true recommendations.

to easily identify the most popular product. The following **Figure 1**shows the graphical abstract of Neural Collaborative Filtering Approach for True Recommendations and **Figure 2** illustrates the architecture of the propose HNCF.

Here Some nomenclatures are used such as a user $u \in U$ writes a review about a product $p \in P$ while the review is denoted by $d \in D$ with n sentences $(s_1, s_2, \ldots s_n)$ where l_i is the length of i^{th} sentence with l_i words $(w_{i1}, w_{i2}, \ldots w_{il_i})$. We employ a hierarchal structure with the bidirectional LSTM to obtain the document representation for modeling the user-product interaction and document level text semantics. Word embedding $\mathrm{W}_{ij} \in R^d$ are employed for each word w_{ij} at the word level. $(\mathrm{W}_{i1}, \mathrm{W}_{i2}, \ldots \mathrm{W}_{il_i})$ are received by BiLSTM and hidden states $(h_{i1}, h_{i2}, \ldots h_{il_i})$ are generated or fetch the last hidden state h_{il_1} or sentence

representation S_i is obtained by the usage of the attention mechanism. Generated sentence representation $(\mathrm{S}_1, \mathrm{S}_2, \ldots \mathrm{S}_n)$ is fed into BiLSTM and in a similar way document d^{up} representation is also obtained.

B. TEXT ANALYTICS AND NLP

NLP empowers the computing machine to communicate with people characteristically. It encourages the machine to comprehend human language and get significance from it. NLP is material in a few tricky from discourse acknowledgement, language interpretation, and grouping documents to data extraction. Breaking down movie comments is one of the great guides to show a basic NLP Bag-of-words model, on movie comments.

1) PREPROCESSING

Data cleaning is essential for text investigation activities of NLP utilizing Stanford CoreNLP,[2] for example, "Tokenization", "Normalization", "Stemming", "Lemmatization" and "POS labelling". because noise data affect the accuracy of predictions.

C. HIERARCHICAL USER ATTENTION PRODUCT ATTENTION

HUAPA is based on two main components first one HUA hierarchal user attention network and the second one is the HPA Hierarchal Product attention network. These hierarchical attention networks are used to incorporate the two information user preferences and product characteristics for document representation of user and product into low dimension. After that, these two representations are concatenated for final document level representation these are used to predict the interaction between user and product, and the overall sentiment of the review of a product given by a user.

1) HIERARCHICAL USER ATTENTION

Not all the words imitate equally User's preferences and sentiments from the user's point of view. Using this idea,

[2] https://github.com/stanfordnlp/CoreNLP

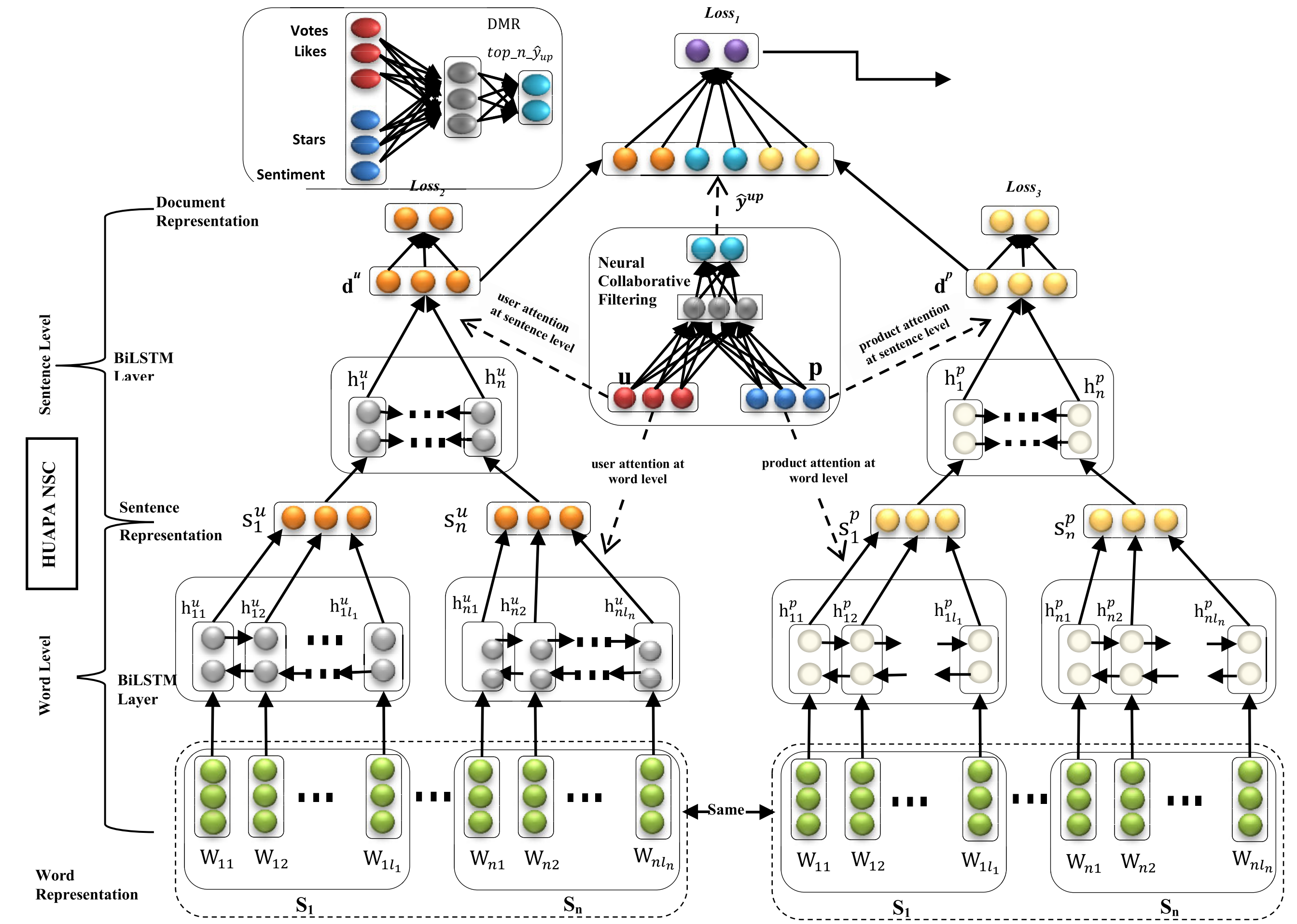

FIGURE 2. The proposed architecture of HNCF.

the user's specific words are extracted by using a user attention mechanism that is important to declare the sentence's meaning. The informative word representation generates the aggregated sentence representation. So, formally we can say that the weighted sum of words level hidden states in the user's view enhanced the sentence representation S_i^u as.

$$S_i^u = \sum\nolimits_{j=1}^{l_i} \alpha_{ij}^u h_{ij}^u \tag{25}$$

Here the hidden state of jth word in the ith sentence is presented by h_{ij}^u while the attention weight of h_{ij}^u is represented as α_{ij}^u and computes the importance of jth for the current user. Each user u is mapped into continuous and real-world value vector $U \in R^{d^u}$. The dimension of user embedding is represented as d^u. For the attention weight, α_{ij}^u of each hidden sate is defined as.

$$e(\mathrm{h}_{ij}^u, \mathrm{u}) = (\mathrm{v}_w^u)^{\mathrm{T}} \tanh(\mathrm{W}_{wh}^u \mathrm{h}_{ij}^u + \mathrm{W}_{wu}^u \mathrm{u} + \mathrm{b}_w^u) \tag{26}$$

$$\alpha_{ij}^u = \frac{\exp\left(e\left(\mathrm{h}_{ij}^u, \mathrm{u}\right)\right)}{\sum_{k=1}^{l_i} \exp\left(e\left(\mathrm{h}_{ik}^u, \mathrm{u}\right)\right)} \tag{27}$$

Here weight vector and its transpose are represented as a_{ij}^u and (v_w^{uT}) respectively while weight matrices are represented as W_{wh}^u and W_{wu}^u respectively. In the Sentence representation of current users, the importance of the word is computed by the score function e. same sense, at the sentence level, each sentence does not contribute equally to document semantic for users. User vector u with attention mechanism in word-level for document representation is also used at the sentence level. In the user view, document representation is obtained as:

$$e(\mathrm{h}_i^u, \mathrm{u}) = (\mathrm{v}_s^u)^T \tanh(\mathrm{W}_{sh}^u \mathrm{h}_i^u + \mathrm{W}_{su}^u \mathrm{u} + \mathrm{b}_s^u) \tag{28}$$

$$\beta_{ij}^u = \frac{\exp\left(e\left(\mathrm{h}_i^u, \mathrm{u}\right)\right)}{\sum_{k=1}^{n} \exp\left(e\left(\mathrm{h}_k^u, \mathrm{u}\right)\right)} \tag{29}$$

$$d^u = \sum\nolimits_{i=1}^{n} \beta_i^u \mathrm{h}_i^u \tag{30}$$

Here ith sentence hidden state is represented as h_i^u in the review document. the hidden state weight is represented as β_{ij}^u in a similar way to the word level representation.

2) HIERARCHICAL PRODUCT ATTENTION

A similar way to user document representation, each word or sentence has different information to text semantic about a different product. Using this sense, product information representation is achieved by hierarchical product attention

(HPA) in a similar way of hierarchical user attention (HUA). The sentence S_i^p and document d^p representation in the product's view is computed as.

$$S_i^p = \sum\nolimits_{j=1}^{l_i} \alpha_{ij}^p h_{ij}^p \tag{31}$$

$$d^p = \sum\nolimits_{i=1}^{n} \beta_i^p \mathrm{h}_i^p \tag{32}$$

Here the weight of hidden state h_{ij}^p at in word-level is represented as α_{ij}^p and β_i^p while h_i^p is represented at the sentence

3) COMBINED HIERARCHICAL USER ATTENTION AND HIERARCHICAL PRODUCT ATTENTION

Training and final prediction are dons at a high level by combining them both representation of d^u user document and d^p product document. So, the final document representation for classification is obtained by combining them without performing the feature engineering.

$$d^{up} = [d^u; d^p] \tag{33}$$

"Cross Entropy Error" between the ground truth distribution of documents representation and p as follows:

$$loss_1 = \sum\nolimits_{d \epsilon T} \sum\nolimits_{c=1}^{C} p_c^g \left(d^{up}\right) . \log(p_c(d^{up})) \tag{34}$$

Here T represents the training set while p_c^g represents the probability of sentiment label c or interaction of user and product with ground truth being 1 and others being 0. d^u and d^p both have the certain predictive capability to make the review representation, integrate the SoftMax classifier to d^u and d^p to increase the accuracy of user-product interaction and sentiment classification of reviews. The corresponding losses are determined as follows:

$$p^u = \mathrm{SoftMax}(W^u d^u + b^u) \tag{35}$$

$$loss_2 = \sum\nolimits_{d \epsilon T} \sum\nolimits_{c=1}^{C} p_c^g \left(d^{up}\right) . \log(p_c^u(d^{up})) \tag{36}$$

$$p^p = \mathrm{SoftMax}(W^p d^p + b^p) \tag{37}$$

$$loss_2 = \sum\nolimits_{d \epsilon T} \sum\nolimits_{c=1}^{C} p_c^g \left(d^{up}\right) . \log(p_c^p(d^{up})) \tag{38}$$

Predicted distribution of the user's view and product's view is represented as p^u and p^p respectively. The weighted sum of $loss_1$, $loss_2$, and $loss_3$ represent the final loss as follows.

$$L = \lambda_1 loss_1 + \lambda_2 loss_2 + \lambda_3 loss_3 \tag{39}$$

review representation is improved by $loss_1$ and $loss_2$ introducing supervised information. Review sentiment label and user-product interaction are predicted according to the distribution p due to it containing both user and product information.

D. NEURAL COLLABORATIVE FILTERING

we implement the intermediate hidden layers to design user-product interactions module by feeding the d^{up} to obtain non-linear complex relationship of user-product:

$$z_1^{\mathrm{up}} = f_1^{\mathrm{up}} \left(\mathrm{W}_1^{\mathrm{up}} . d_1^{\mathrm{up}} + b_1^{\mathrm{up}}\right) = [d^{up}] \tag{40}$$

$$z_2^{\mathrm{up}} = f_2^{\mathrm{up}} \left(\mathrm{W}_2^{\mathrm{up}} . z_1^{\mathrm{up}} + b_2^{\mathrm{up}}\right) \tag{41}$$

$$z_3^{\mathrm{up}} = f_3^{\mathrm{up}} \left(\mathrm{W}_3^{\mathrm{up}} . z_2^{\mathrm{up}} + b_3^{\mathrm{up}}\right) \tag{42}$$

$$\ldots$$

$$z_L^{\mathrm{up}} = f_L^{\mathrm{up}} (\mathrm{W}_L^{\mathrm{up}} . z_{L-1}^{\mathrm{up}} + b_L^{\mathrm{up}}) \tag{43}$$

$$\hat{y}_{up} = \sigma(\mathrm{h}^T (z_L^{\mathrm{up}})) \tag{44}$$

$$\hat{y}_{up} = tanh(\mathrm{h}^T (z_L^{\mathrm{up}})) \tag{45}$$

Here f_L^{up} the "activation function", $\mathrm{W}_L^{\mathrm{up}}$ is "weight matrix" and b_L^{up} is "bias vector" are represented for the L^{th} layer while L represents the number of layers in the deep interaction module. activation function as tanh () is applied. The interaction's output is the projected interaction value between the user u and the product p. Filtering is performed according to the user's profile or product profile using similarity measures. These features of the profile will be matched using cosine similarity techniques. "*Cosine similarity*" measurement is based on two vectors used for measuring the angle value of the user profile and product profile genre for similarity manipulation and correlations are computed. Here zero angle represents the most similar to users' profile or product's profile and vice versa. Here y_{up} and $\hat{y}_{up}$ may be user or product.

$$\overrightarrow{y_{up}} . \overrightarrow{\hat{y}_{up}} = \left\| \overrightarrow{y_{up}} \right\| . \left\| \overrightarrow{\hat{y}_{up}} \right\| . \cos\theta \tag{46}$$

$$sim\left(\overrightarrow{y_{up}} . \overrightarrow{\hat{y}}_{up}\right) = \cos\theta = \frac{\overrightarrow{y_{up}} . \overrightarrow{\hat{y}}_{up}}{\left\| \overrightarrow{y_{up}} \right\| . \left\| \overrightarrow{\hat{y}_{up}} \right\|} \tag{47}$$

E. NEURAL SENTIMENT CLASSIFICATION

The process of identification and classification of opinions is stated in the microblog or short-text to determine the topic, polarity, attitude, and emotions is known as sentiment analysis. Preprocessing and Text structuring are necessary for machine learning because the machine directly cannot understand the semantics of the text. Sentiment labelling of review is predicted by document level classification. For sentiment analysis of reviews given by users for products, all words or sentences are not participating equally. Because some words or sentences solidly indicate the user's preferences while others show the product's characteristics. Therefore, the sentiment label of reviews is inferred by two kinds of information with latent semantic representation in the user's review and product's reviews by incorporating the information of the user and product for NSC using a hierarchical user attention network and a hierarchical product attention network. Finally, the sentiment label of the review is predicted by the SoftMax classifier in which d^{up} is taken as a feature of users and products. Specifically, C classes of sentiment distribution for review, the SoftMax layer and the Linear layer are implemented to project review representation d^{up}.

$$C^{up} = \mathrm{SoftMax}(W^{up}d^{up} + b^{up}) \tag{48}$$

After evaluating the reviews, the neural sentiment analyzer will compute the semantic score and classify the semantic score that will be tagged with emotions of the favorability label. Semantic emotions will be classified into five major classes on the bases of their relative semantic scores. These classes will indicate the user's emotions as mentioned in the following **Table 2.**

F. DEEP MULTIVARIANT RATING (DMR)

Stars, votes, likes and sentiment polarity are computed to find out the multivariate rating for j$^{\text{th}}$ product for true popularity. Here S^{p_i} the normalized total stars rating about i$^{\text{th}}$ product, V^{p_i} is the normalized total votes rating about i$^{\text{th}}$ product and L^{p_i} is the normalized total likes rating about i$^{\text{th}}$ product given by different users. Here $C_{u_j}^{p_i}$ is a class favorability of user j for the product i.

$$r_{u_j}^{p_i} = \left[f\left(S^{p_i}\right); f\left(V^{p_i}\right); f\left(L^{p_i}\right); f\left(C_{u_j}^{p_i}\right)\right] \tag{49}$$

Here $r_{u_i}^{p_i}$ is the concatenated multivariate overall rating of i$^{\text{th}}$ product for j$^{\text{th}}$ user.

$$\hat{r}_{u_i}^{p_i} = \mathrm{SoftMax}\left(\mathbf{W}^{up} r_{u_i}^{p_i} + \mathbf{b}^{up}\right) \tag{50}$$

$$\boldsymbol{p}_c = \frac{\exp\left(\hat{r}_c^{up}\right)}{\sum_{k=1}^{C} \exp\left(\hat{r}_k^{up}\right)} \tag{51}$$

Here p_c is used for a multivariant rating class. Every product will be labeled with a medal according to a popularity score that is calculated by a Multivariate recommender module that determines the class of popularity with their medal.

$$\mathrm{top\_n\_}\widehat{y}_{up} = \widehat{y}_{up} * p_c = \left[(1) + \frac{1}{(1) + \sqrt{r_{u_j}^{pi} *}}\right] \times \propto \tag{52}$$

The popularity status can be measured by using the multivariate overall rating at ten scales when $\propto =$ 10. This popularity status with their medal will be determined by the multivariate value that will be measured by integration of rating (likes, votes, stars and semantic score of reviews) about a product. We will stretch the multivariate semantic value on the 10 scales. Every product will be labeled with a medal according to a popularity score that is calculated by a Multivariate recommender module that determines the class of popularity with their medal. The popularity status and their medal are divided into five classes will indicate the importance of products. After that multivariate rating scores are classified according to flowing fuzzy. represented in following **Table 3** , the ranges of the popularity scores, their relative medals and status of popularity are classified according to the criteria.

V. EXPERIMENTAL SETUP

A. METHODOLOGY

As the experiment is based on the reviews and other ratings from IMDb, Fandango, Metacritic and Rotten Tomatoes. The reviews are consisting of text-based information. The text-based data is taken from the three datasets such as IMDB, challenge and yelp 2013-2014. The data from these datasets is preprocessed using Stanford core NLP [47] and considered for training, testing, and validation. The training data contains 80% of the overall data, for testing purposes 10% data is used and the rest of the 10% is used for the validation purpose as the validation increased the accuracy of the sentiment process and collaborative filtering process. For the pre-training purpose, two hundred words are filtered as embedded words, these words are considered in each dataset by using the Skip Gram [37]. For better analysis, these words are applied to the user's attention and the product attention because only the training is not enough for giving the efficient proposed results. The user's and product attention are applied by the two hundred words in a uniform distribution method represented by U (0:01; 0:01). While two hidden layers are given the dimensions that the simple hidden layer of LTSM is given the dimension of one hundred words while the bi-directional LTSM contains the embedding of two hundred words. Furthermore, the threshold value is set for giving the limit to each sentence in the document, the limit of sentences in a document is forty while the quantity of each sentence contains the 50 words. The initial rate of learning is set of 0.005 and for updating the attributes of data Adam [48] is used. Sorting of attributes is the most important task to select the most appropriate features but the regularization and dropout methods are not used [49]. Hyperparameters of interaction module are randomly initialized to train the model from scratch. Predictive factors are measured by the last hidden layer for NSC and Deep User-Product Interaction module.

B. EVALUATION METHODS

The performance of the classification is achieved by the standard measure of "*Accuracy*" and the "*Root Means Square Error*" (RMSE) method is used to find the ratio of grounded document labels and predicted labels. Reviews sentiment classification and user-product interaction modeling results in terms of accuracy and which mean (High accuracy, high is better) and RMSE (less is better) are the evaluation methods. Bold results show our performance. The propose novel model overshadows the previous best state-of-the-art approach. These equations are defined below:

$$Accuracy = \frac{T}{N} \tag{53}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(\mathrm{gd}_K - \mathrm{pr}_K\right)^2}{N}} \tag{54}$$

In the above equation T represents the number of predicted labels of the documents. In the second equation, gd_K and pr_K

TABLE 2. Emotional status and emotions.

Semantic Score	Favorability Status	Emotion
0.8 <= and <=1.00	"Highly Favorable"	😀
0.6 <= and < 0.8	"Favorable"	🙂
0.4 <= and < 0.6	"Average Favorable"	😮
0.2<= and <0.40	"Unfavorable"	😢
0.00< 0.20	"Highly unfavorable"	😭

TABLE 3. Popularity status and medals.

Popularity Score	Medal Tittle	Popularity Status	Medals
8-10	"Platinum"	"Highly Popular"	1
6-7.9	"Gold"	"High Popular"	2
4-5.9	"Silver"	"Average Popular"	3
2-3.9	"Bronze"	"Unpopular"	4
0-1.9	"Copper"	"Highly Unpopular"	5

are used to represent the ground predicted document labels and predicted labels. While *N* shows the total number of documents that are reviewed.

C. BASELINES

The prosed models are compared with the several methods of baselines for classification and filtering.

- ***Majority Assigns:*** This defines the majority of label training set to every reviewed document in the given set.
- ***Trigram:*** Trigram makes use of bigrams and unigrams for training "*Support Vector Machine*" (SVM) classifier with the collaboration of LIblinear [50].
- ***Text Feature:***Filters the most appropriated text attributes for the training purpose -sophisticated text features to train SVM [51].
- ***UPF:*** Gets the corresponding and leniency attributes of the text sentiments for product popularity attributes. Reference [52] from training data, and further concatenates them with the features in Trigram and Text Feature.
- ***AvgWordvec:*** Average embedding of words creates the documental representation, then enters these words into the SVM classifier as attributes of the datasets. SSWE understands the sentiments based on specific embedded words. (SSWE) ….(..) [53], and uses the max/min/average method of pooling to get the representation of the document.
- ***RNTN + RNN:***Implements the "*Recursive Neural Tensor Network*" (RNTN) [54] to get the representation of sentences and then gives these words to RNN. Afterward, the vector's space of hidden RNN is calculated to obtain the average of the sentiments to create the documents for classification.
- ***Vectors space:*** For paragraphs implements the model for distribution memory for classifying the documents. [55]. The size of the window is activated for validation purposes.
- ***JMARS:*** is an algorithm for classification which sorts the data by taking the information of the user and different filtering aspects and finally gives the rating of the classification on document level …(..) [56]
- ***UPNN:*** implements the classification at the world level based on information about user and product and gives the outcomes in the preference matrix, with the help of the CNN classifier [57]. This also gives the feasibility to alter the meaning of the words in the preference matrices. Finally, it combines the vectors of input user or product and reviewed documental results into the matrix as the attributes and the gives to the SoftMax layer [24].
- ***LUPDR:*** classifies the user and product reviews by implementing the RNN in the form categories [25].
- ***NSC:*** implements the LSTM model which is based on procedural strategies for classification only and Local attention is added to LTSM for further classification.
- ***NSC+UPA:*** puts information of user and product take accounts together and uses a hierarchical LSTM model with an attention mechanism to produce a review representation for classification of sentiment. Unlike [11] and [27] do not implement the model.
- ***NSC+UPA and NSC+LA:*** with bidirectional LSTM. To make the experimental results more convincing, we implement and train them in our experimental settings. In addition to LUPDR and the models related to NSC, we report the results in [24] since we use the same datasets for other baseline methods above.
- ***HUAPA:*** incorporating the users and product information for emending and classification of reviews documents using hierarchical user attention and hierarchical product attention, using attention mechanism more improved way as compared to conventional learning methods [34].

TABLE 4. Models comparison without information of user and product.

Models	IMDB		Yelp 2013		Yelp 2014	
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
Majority	0.196	2.495	0.411	1.060	0.392	1.097
Trigram	0.399	1.783	0.569	0.814	0.577	0.804
TextFeature	0.402	1.793	0.556	0.845	0.572	0.800
AvgWordvec+SVM	0.304	1.985	0.526	0.898	0.530	0.893
SSWE+SVM	0.312	1.973	0.549	0.849	0.557	0.851
Paragraph Vector	0.341	1.814	0.554	0.832	0.564	0.802
RNTN+Recurrent	0.400	1.764	0.574	0.804	0.582	0.821
UPNN(CNN and no UP)	0.405	1.629	0.577	0.812	0.585	0.808
NSC	0.443	1.465	0.627	0.701	0.637	0.686
NSC+LA	0.487	1.381	0.631	0.706	0.630	0.715
NSC+LA(BiLSTM)	0.490	1.325	0.638	0.691	0.646	0.678

TABLE 5. Models comparison with information of user and product.

Models	IMDB		Yelp 2013		Yelp 2014	
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
Trigram+UPF	0.404	1.764	0.570	0.803	0.576	0.789
TextFeature+UPF	0.402	1.774	0.561	1.822	0.579	0.791
JMARS	N/A	1.773	N/A	0.985	N/A	0.999
UPNN(CNN)	0.435	1.602	0.596	0.784	0.608	0.764
UPNN(NSC)	0.471	1.443	0.631	0.702	N/A	N/A
LUPDR	0.488	1.451	0.639	0.694	0.639	0.688
NSC+UPA	0.533	1.281	0.650	0.692	0.667	0.654
NSC+UPA(BiLSTM)	0.529	1.247	0.655	0.672	0.669	0.654
HUAPA	0.550	1.185	0.683	0.628	0.686	0.626

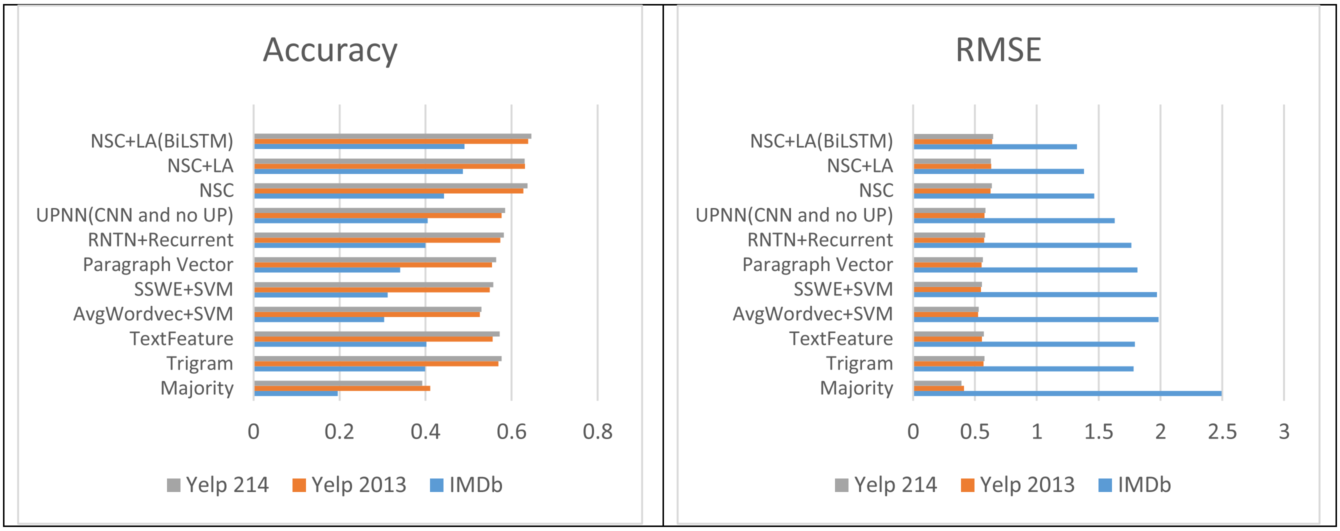

FIGURE 3. Models comparison without information of user and product.

- ***Multivariant Rating:***This model [10] is designed to compute the multivariant rating in which sentiment scores are find out using the TFIDF approach while model [11] is also designed to compute the multi-variant rating in which sentiment scores computed by using the UPA to find out the significant popularity of products.
- ***NCF:***is based on the MLP and GMF for CF based recommender systems. The nonlinear relationship between users and products is captured by only using one-hot vectors for the user-product implicit interaction model [20] unlike our proposed model.
- ***J-NCF:***This model used the DNNs for the features of user and product vector separately concatenated and

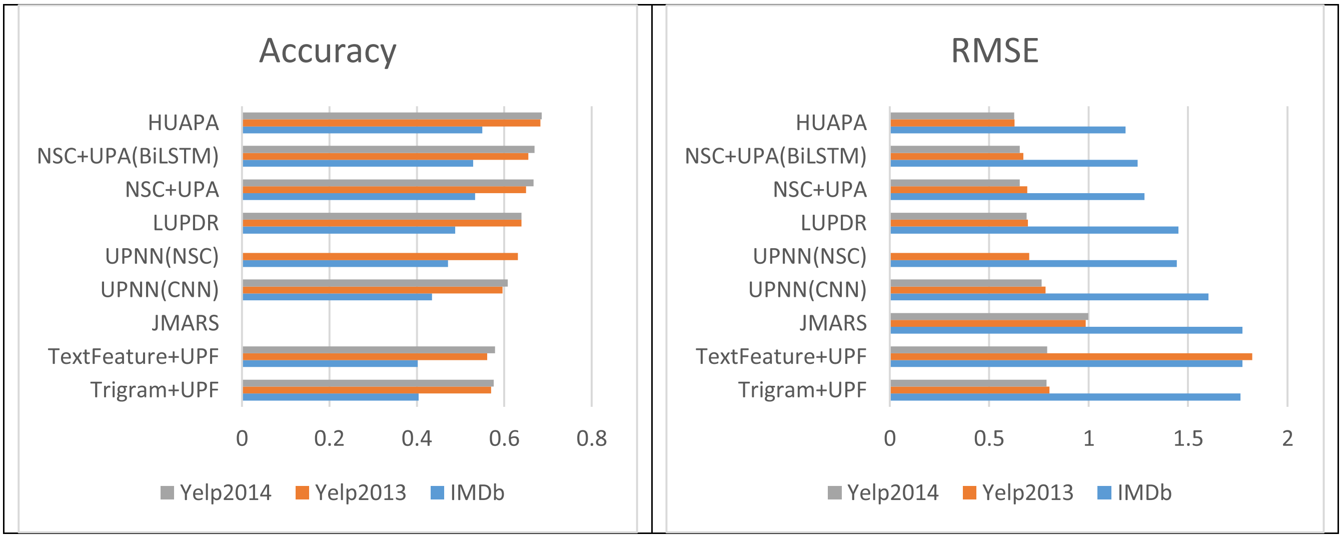

FIGURE 4. Models comparison with information of user and product.

TABLE 6. Effect of attention mechanism.

Models	IMDB		Yelp 2013		Yelp	2014
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
NSC+LA(BiLSTM)	0.490	1.325	0.638	0.691	0.646	0.678
HUA	0.521	1.300	0.649	0.691	0.663	0.661
HPA	0.493	1.326	0.641	0.681	0.646	0.678
HUAPA	0.550	1.185	0.683	0.628	0.686	0.626

TABLE 7. Effect of the different weighted loss.

λ1	λ2	λ3	IMDB		Yelp 2013		Yelp	2014
			Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
1.0	0.0	0.0	0.538	1.229	0.669	0.658	0.675	0.647
0.7	0.3	0.0	0.541	1.239	0.672	0.644	0.680	0.641
0.7	0.0	0.3	0.540	1.287	0.675	0.646	0.679	0.633
0.4	0.3	0.3	0.550	1.185	0.683	0.628	0.686	0.626

used as input to the integration model in nonlinear representation [28].

- ***DMF:*** in this model, rating matrix factorization is based on multilayer perceptrons. unlike our model, after performing the users and product projection into the low dimension, the interaction between users and product is computed by the inner product, which is the linear kernel [21].
- ***HDFC:*** in this model, user and product attention combined mechanism are used to generate the users prefrences as well as multivarinting [29].

VI. RESULTS AND EVALUATION

A. MODEL COMPARISONS FOR SENTIMENT CLASSIFICATION

The outcomes of the experiments without using user and product attention while with user product attention are given in **Table 4** and **Table 5**, **Figure 3** and **Figure 4** respectively given below. The outcome contains two parts. The first part contains the results of the model related to the local preferences only, while the second part contains the results of local and global preferences. From the results of local preferences, the majority haven't played an accurate role rather on the comparison of SVM, NN worked well on text, user and product results. Also, the outcomes illustrate NSC+LA more sophisticated and improved results as compared to the only implementation of NSC only. Because this gives the option to select more appropriate and meaningful words. The improvements in the classification also show the reason for implementing attention mechanisms. From the second part, it can be observed that by implementing the global preferences the results are much improved as compared to the first results. For example, Text Feature with the addition of UPF got improvements based on 0.5%. The NSC with UPA

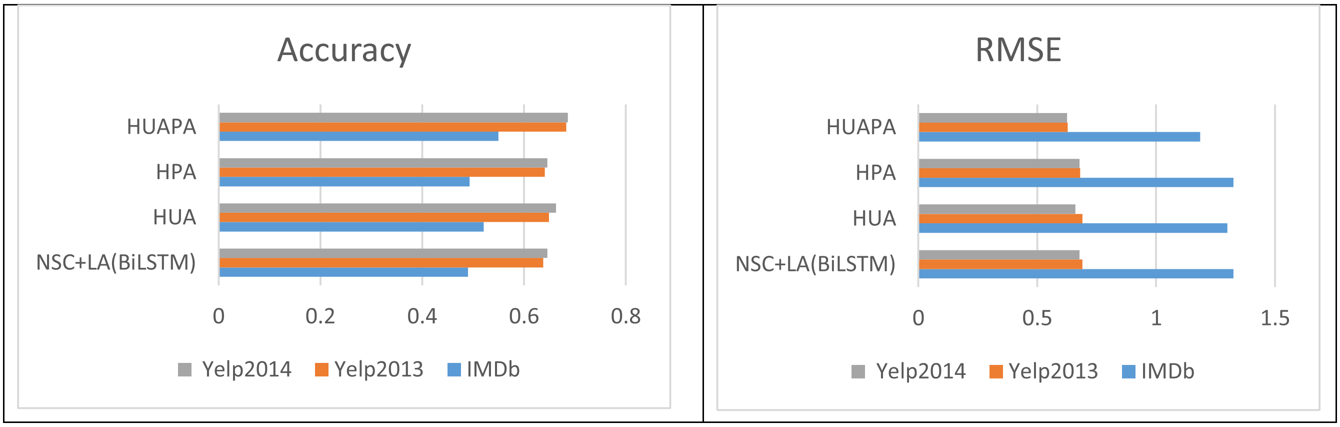

FIGURE 5. Effect of attention mechanism.

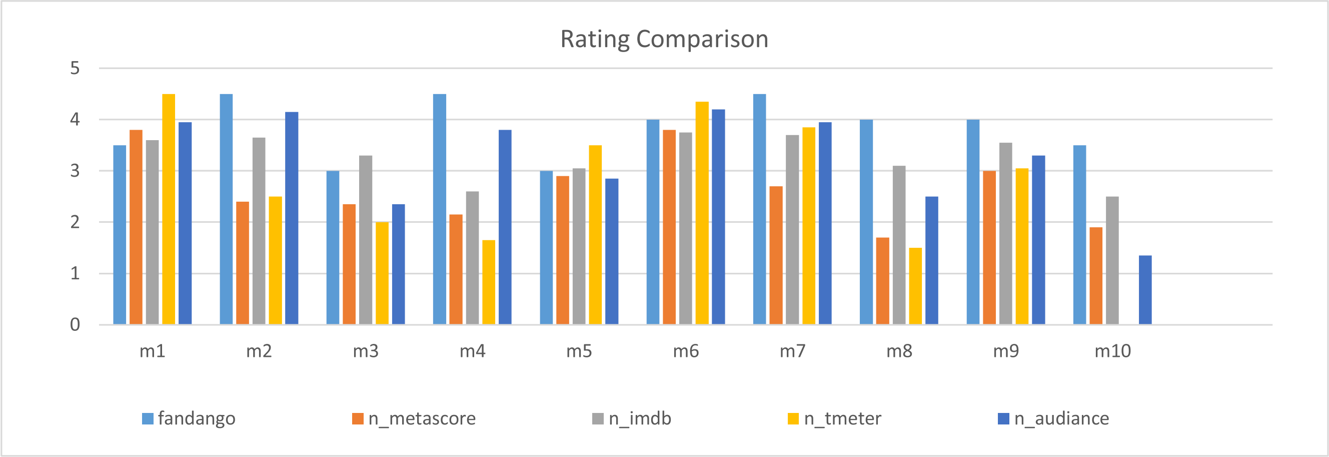

FIGURE 6. Movie_ratings_2016_17 differences.

improves based on 2.3% on the yelp 2013 accuracy level. The comparison results show that the accuracy of the sentiment classification has improved by implementing the user and product preferences. The outcome shows that the proposed model with global preferences and attention performed well on all the available datasets. The model still improved itself regardless of larger datasets. The proposed model worked in the best manner on the small datasets with an accuracy of 2.8% and 1.7%. The values of accuracy are still the same in giving the larger datasets. The analysis of the outcomes shows that the implemented model has great performance on the information of user and product. Which gives a new way for sentiment classification.

B. EFFECT OF ATTENTION MECHANISM

The analyze the results on a single attention-based model of user and product sentiments, we have also used HUA and HPA. The performance of the single effect attention-based model is given in **Table 6** and **Figure 5**. From the table, it can be observed that the model NSC+LA (BiLSTM) only used the local attention and preferences, HUA and HPA both got some advancements, which are related to the rationality of incorporating sentiment classification on sentiment attention. The outcomes also depict that the product and product attention reflex the information in more accurate ways. The information of the user is highly effective to improve the representation of the reviews. Although some words depict more relevancy to the sentiment and some are not. Hence it is authentic that the attention mechanism is more related to the sentiment analysis. In the comparison of single attention and attention of the product reviews our implemented model outperforms. The results show that the user's attention and preferences on the products can capture the specific attributes of the product and user preferences.

The effectivity of attention of user and product can be validated by the instances of reviews in Yelp 2013 datasets. For example, we observed some reviews, review 1: "we love the ambiance and how cool this place is". review 2: "the bar area is good 'people watching' and I love the modern contemporary décor." review 3: "Much of it was quite good but I was disappointed with the spider roll." in the first review, the words "love" and "cool" have highest attention

TABLE 8. Comparison models for collaborative filtering.

Models	IMDB		Yelp 2013		Yelp 2014	
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
[21] DMF	0.120	24.033	0.130	23.401	0.130	23.401
[20] NCF	0.170	20.871	0.184	19.986	0.204	18.721
[28] J-NCF	0.246	16.064	0.204	18.721	0.217	17.898
[10] CF+SC+MR	0.318	11.511	0.313	11.827	0.333	10.562
[11]CF+NSC+MR	0.400	6.325	0.444	3.542	0.457	2.720
[29] HDCF	0.465	2.214	0.479	1.328	0.478	1.391
[Proposed-Model] HNCF	**0.514**	**0.885**	**0.516**	**1.012**	**0.493**	**0.443**

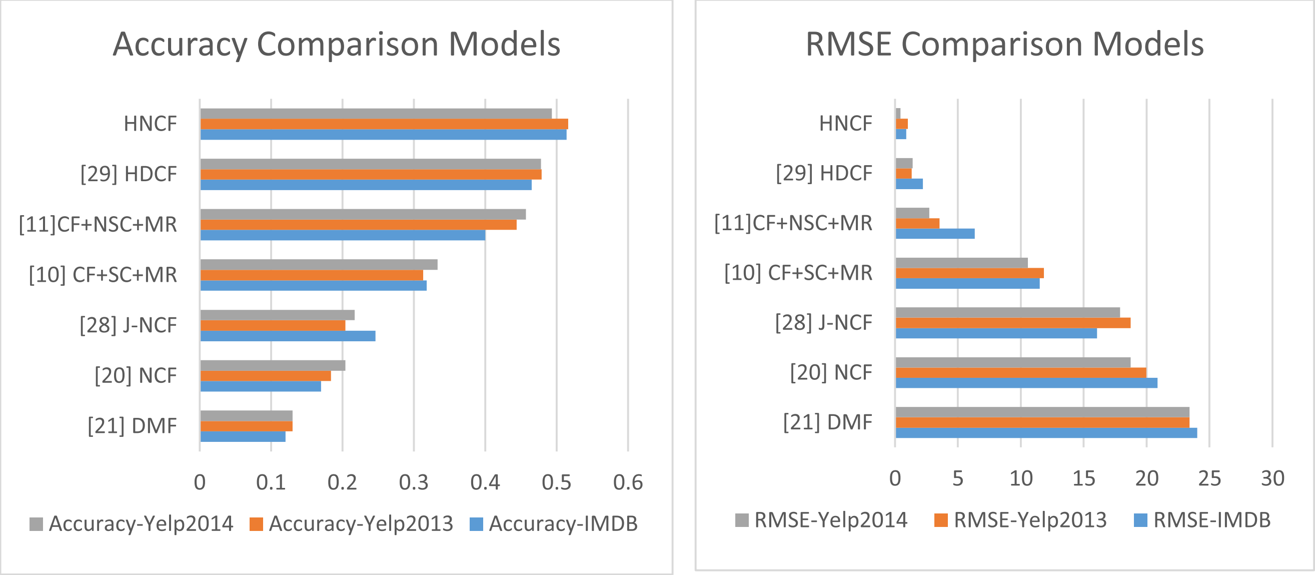

FIGURE 7. Comparison models for collaborative filtering.

weight for user attention and product attention respectively. User's preference or affection and product characteristics are expressed by "love" and "cool" words respectively. There are also some reviews in which the user's preferences are inconsistent with the product's characteristics, such as in reviews 3. It shows that the word "good" is used for attributes of the product and the word "disappointed" is considered a negative word for the product opinion. The hierarchical attention mechanism demonstrates that our model can capture the global preference of users and the characteristics of the product.

C. EFFECT OF THE DIFFERENT WEIGHTED LOSS

The representation of the weighted loss as $loss_1$, $loss_2$ and $loss_3$. For better analysis proportion value is set for the weighted loss. When Lambda 2 is initialized by 0, we don't implement the $loss_2$ for better of review representation. Similarly, in the same way, lambda 3 to 0 for the avoidance of the effect of loss. The results of the experiments are given in the **Table 7** and **Figure 6** depicts the results of the comparison of implemented models using the datasets. It shows that the mechanism of attention is more suitable for product and user reviews. In terms of improvement concerning losses such as $loss_1$ and $loss_2$ both HUAPA and the model performed well in the process of sentiment analysis or classification.

D. EFFECT OF MULTIVARIANT RATING

These four famous "*IMDb*",[3] "*Metacritic*"[4] and "*Rotten Tomatoes*"[5] and "*Fandango*"[6] Movie_ratings_2016_17[7] datasets are discussed and shown in fig 6.4 to raise the issues that lead to adopting the Deep Multivariant Rating approach. Yet how are these four platforms different, decrease the confidence and are you supposed to trust them to get the rundown on movies? Here's what you need to say. So, we have now looked at what each of these platforms, along with their benefits and drawbacks, has to deliver. There's no one site, as you have already suspected, that is perfect for everything.

[3] https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV#ratings

[4] https://www.metacritic.com/about-metascores

[5] https://www.rottentomatoes.com/critics/

[6] https://www.fandango.com/movie-reviews

[7] https://github.com/mircealex/Movie_ratings_2016_17

However, for various reasons, we discuss each of these sites. IMDb is a great place to see what the general public thinks about a movie. If you don't know what the critics are saying and want to see what people like you think of a movie, then use IMDb. Only be aware that viewers sometimes distort the vote at 10-star ratings, which can somewhat reduce scores. Rotten Tomatoes delivers the best overall impression on whether a movie is worth watching in one instant. If you just trust high critics 'opinions and want to know if a movie is at least good, then you can use Rotten

Tomatoes. Although the Fresh / Rotten binary can oversimplify critics 'sometimes nuanced opinions, it can still assist you in weeding out junk movies. The balanced aggregate score is provided by Metacritic. If you don't know which views of the reviewers go into the final score and want to see a general average, then use Metacritic. Its expectations are still unknown but Metacritic makes it easy to compare side-by-side professional reviews with feedback. Fandango has a consumer rating of 1.1 stars from 1,039 reviews indicating that most consumers are generally dissatisfied with their purchases. The confusion is caused by a uniform distribution and it was found that the quantitative ratings on Fandango's site were always rounded to the next largest half-star, rather than the closest one (for example, a 3.1 average rating for a film should have been rounded to 3.5 stars, instead of 3.0).

We analyze the datasets and take the 10 samples of the movie to show, how different ratings from different sites for the same movie or products are varied. All problems in the discussion and graph show that different sites have different rating criteria and rating scores are different for the same movie which reduces the confidence of users in the recommendation system and reduces the popularity of actual popular products or movies. That is the reason we combine the discreet and continuous rating to generate the multivariant rating that incorporates the true rank product.

E. COMPARISON MODELS FOR COLLABORATIVE FILTERING

Let's now analyze the proposed model with the baseline's collaborative filtering and shows the results in **Table 8** and **Figure 7**. Taking the sample size 1000 from datasets IMDB, yelp2013, yelp2014 and experimented with shows the comparisons of collaborative filtering models. The propose models display improved performance showing the effectiveness of Deep Learning techniques in optimizing recommendation performance than the conventional CF models j-NCF, NCF, and DMF. These models J-NCF, NCF, and DMF still lose in terms of accuracy against the proposed model's accuracy and RMSE, which indicates that a Hybrid Neural Network structure that closely integrates deep learning features of users and products by incorporating users' preferences and products' characteristics using the hierarchical attention mechanism, and deep user-product interaction modeling using MLP improves recommendation performance.

The accuracy of our proposed models compared with baseline methods shows improvement from 14% to 23% while RMSE decreased from 15% to 1% at given datasets. That shows the considerable improvement in collaborative filtering for recommendation systems.

VII. CONCLUSION AND FUTURE WORK

Through this research work, we were investigating the collaborative filtering of neural network architectures. This work complements the standard shallow models for collaborative filtering, opening up a new field of recommendation-based research opportunities based on deep learning. For the recommender system, we propose a novel model a hybrid neural collaborative filtering framework HNCF that contains the four major modules that are jointly and tightly coupled. Initially, user and product features learning and encoded through a hierarchical attention network for user and product by incorporating the user's preferences and product characteristics, and user product interaction learning through multi-layer perceptron for user-product interaction modeling. using the feature vectors of users and products as inputs fed to user-product interactions module for prediction of users or product that has similar taste to other by measuring the similarity of users or products in an improved and reliable way. To make HNCF suitable for the top-N recommendation task, we add a revolutionary feature for losses. This requires data from both. First, concerning research study, we propose HNCF with more auxiliary information is used to improve the performance of [20] [28], and [29]. It is also essential to analyze heterogeneous information in an information base to enhance the efficiency of deep learning recommending systems. Secondly, in a session with Hierarchical Attention Network, we explore a user's contextual information to discuss complex aspects of systems. Additionally, a system of focus applied to HNCF, that could filter out uninformative material and choose the most relevant products while at the same time maintaining better understanding. The findings from the studies indicate HNCF's effectiveness. We also experimentally examined the performance of HNCF under circumstances, e.g., in addition, we also evaluated HNCF model with a large dataset, i.e. the results indicate that HNCF also gives outstanding performance against state-of-the-art baseline models on given datasets. Finally, because we found HNCF to be computationally more expensive than NCF, we expect to refine our model's structure and details of implementation to increase the performance.

ACKNOWLEDGMENT

This work was supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R236), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, 1994, pp. 175–186, doi: 10.1145/192844.192905.

[2] L. Jiang, Y. Cheng, L. Yang, J. Li, H. Yan, and X. Wang, "A trust-based collaborative filtering algorithm for e-commerce recommendation system," *J. Ambient Intell. Hum. Comput.*, vol. 10, pp. 3023–3034, Jun. 2019, doi: 10.1007/s12652-018-0928-7.

[3] A. K. Sahoo, C. Pradhan, R. K. Barik, and H. Dubey, "DeepReco: Deep learning based health recommender system using collaborative filtering," *Computation*, vol. 7, no. 2, p. 25, 2019, doi: 10.3390/computation7020025.

[4] P. Chen and J. Li, "A recurrent model with self-attention for product repurchase recommendation," in *Proc. 4th Int. Conf. Math. Artif. Intell.*, 2019, pp. 199–203, doi: 10.1145/3325730.3325763.

[5] J. Salter and N. Antonopoulos, "Cinemascreen recommander agent: Combining collaborative and content-based filtering," *IEEE Intell. Syst.*, vol. 21, no. 1, pp. 35–41, Jan. /Feb. 2006, doi: 10.1109/MIS.2006.4.

[6] Q. Yao, X. Liao, and H. Jin, "Hierarchical attention based recurrent neural network framework for mobile MOBA Game recommender systems," in *Proc. IEEE Intl Conf. Parallel Distrib. Process. Appl., Ubiquitous Comput. Commun., Big Data Cloud Comput., Social Comput. Netw., Sustain. Comput. Commun. (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*,Dec. 2019, pp. 338–345, doi: 10.1109/BDCloud.2018.00060.

[7] R. Huang, S. McIntyre, M. Song, E. Haihong, and Z. Ou, "An attention-based recommender system to predict contextual intent based on choice histories across and within sessions," *Appl. Sci.*, vol. 11, no. 4, p. 6633, 2018, doi: 10.3390/app8122426.

[8] V. P. Khadse, S. M. Basha, N. Iyengar, and R. D. Caytiles, "Recommendation engine for predicting best rated movies," *Int. J. Adv. Sci. Technol.*, vol. 110, pp. 65–76, Jan. 2018, doi: 10.14257/ijast.2018.110.07.

[9] A. Da'u and N. Salim, "Sentiment-aware deep recommender system with neural attention networks," *IEEE Access*, vol. 7, pp. 45472–45484, 2019, doi: 10.1109/ACCESS.2019.2907729.

[10] M. Ibrahim and I. S. Bajwa, "Design and application of a multi-variant expert system using apache Hadoop framework," *Sustainability*, vol. 10, no. 11, p. 4280, 2018, doi: 10.3390/su10114280.

[11] M. Ibrahim, I. S. Bajwa, R. Ul-Amin, and B. Kasi, "A neural network-inspired approach for improved and true movie recommendations," *Comput. Intell. Neurosci.*, vol. 2019, Aug. 2019, Art. no. 4589060, doi: 10.1155/2019/4589060.

[12] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–739, Jun. 2005, doi: 10.1109/TKDE.2005.99.

[13] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, 2019, doi: 10.1145/3285029.

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

[15] B. T. Betru, C. A. Onana, and B. Bernabe, "Deep learning methods on recommender system: A survey of state-of-the-art," *Int. J. Comput. Appl.*, vol. 162, no. 10, pp. 17–22, 2017, doi: 10.5120/ijca2017913361.

[16] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. ACM Conf. Recommender Syst.*, 2016, pp. 233–240, doi: 10.1145/2959100.2959165.

[17] S. Molaei, A. Havvaei, H. Zare, and M. Jalili, "Collaborative deep forest learning for recommender systems," *IEEE Access*, vol. 9, pp. 22053–22061, 2021, doi: 10.1109/ACCESS.2021.3054818.

[18] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 791–798, doi: 10.1145/1273496.1273596.

[19] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto–encoders for top-N recommender systems," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, 2016, pp. 153–162, doi: 10.1145/2835776.2835837.

[20] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[21] H. J. Xue, X. Y. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proc. IJCAI*, 2017, pp. 3203–3209, doi: 10.24963/ijcai.2017/447.

[22] 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*, 2017, pp. 425–434, doi: 10.1145/3018661.3018665.

[23] T. Bansal, D. Belanger, and A. McCallum, "Ask the GRU: Multi-task learning for deep text recommendations," in *Proc. 10th ACM Conf. Recommender Syst.* , 2016, pp. 107–114, doi: 10.1145/2959100.2959180.

[24] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1422–1432, doi: 10.18653/v1/d15-1167.

[25] T. Chen, R. Xu, Y. He, Y. Xia, and X. Wang, "Learning user and product distributed representations using a sequence model for sentiment analysis," *IEEE Comput. Intell. Mag.*, vol. 11, no. 3, pp. 34–44, Aug. 2016, doi: 10.1109/MCI.2016.2572539.

[26] Y. Li, W. Jiang, L. Yang, and T. Wu, "On neural networks and learning systems for business computing," *Neurocomputing*, vol. 275, pp. 1150–1159, Jan. 2018, doi: 10.1016/j.neucom.2017.09.054.

[27] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1650–1659, doi: 10.18653/v1/d16-1171.

[28] W. Chen, F. Cai, H. Chen, and M. D. E. Rijke, "Joint neural collaborative filtering for recommender systems," *ACM Trans. Inf. Syst.*, vol. 37, no. 4, pp. 1–30, 2019, doi: 10.1145/3343117.

[29] M. Ibrahim, I. S. Bajwa, N. Sarwar, H. A. Waheed, M. Z. Hasan, and M. Z. Hussain, "Improved hybrid deep collaborative filtering approach for true recommendations," *Comput. Mater. Continua*, vol. 74, no. 3, pp. 5301–5317, 2023, doi: 10.32604/cmc.2023.032856.

[30] A. J.-P. Tixier, "Notes on deep learning for NLP," 2018, *arXiv:1808.09772*.

[31] S. El Hihi and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies," in *Proc. NIPS*, 1995, pp. 1–7.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[33] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 551–561, doi: 10.18653/v1/d16-1053.

[34] Z. Wu, X. Y. Dai, C. Yin, S. Huang, and J. Chen, "Improving review representations with user attention and product attention for sentiment classification," in *Proc. AAAI Conf. Artif. Intell. 30th Innov. Appl. Artif. Intell. Conf. 8th AAAI Symp. Educ. Adv. Artif. Intell.*, 2018, pp. 5989–5996.

[35] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015, *arXiv:1409.0473*.

[36] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489, doi: 10.18653/v1/n16-1174.

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations ofwords and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1–9.

[38] H. Papadakis, A. Papagrigoriou, C. Panagiotakis, E. Kosmas, and P. Fragopoulou, "Collaborative filtering recommender systems taxonomy," *Knowl. Inf. Syst.*, vol. 64, no. 1, pp. 35–74, 2022, doi: 10.1007/s10115-021-01628-7.

[39] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, p. 2, Jan. 2009, doi: 10.1155/2009/421425.

[40] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295, doi: 10.1145/371920.372071.

[41] D. A. Adeniyi, Z. Wei, and Y. Yongquan, "Automated web usage data mining and recommendation system using K-nearest neighbor (KNN) classification method," *Appl. Comput. Inform.*, vol. 12, no. 1, pp. 90–108, 2016, doi: 10.1016/j.aci.2014.10.001.

[42] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering categories and subject descriptors," in *Proc. KDDCup*, Aug. 2007, pp. 1–4, doi: 10.1145/1557019.1557072.

[43] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manag.*, 2015, pp. 811–820, doi: 10.1145/2806416.2806527.

[44] S. Sedhain, A. K. Menony, S. Sannery, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 111–112, doi: 10.1145/2740908.2742726.

[45] J. Lian, F. Zhang, X. Xie, and G. Sun, "CCCFNet: A content-boosted collaborative filtering neural network for cross domain recommender systems," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 817–818, doi: 10.1145/3041021.3054207.

[46] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1725–1731, doi: 10.24963/ijcai.2017/239.

[47] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford coreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2015, pp. 55–60, doi: 10.3115/v1/p14-5010.

[48] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014, Accessed: Mar. 23, 2020. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[50] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol 9, pp. 1871–1874, Jun. 2014.

[51] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *J. Artif. Intell. Res.*, vol. 50, pp. 732–762, Aug. 2014, doi: 10.1613/jair.4272.

[52] W. Gao, N. Yoshinaga, N. Kaji, and M. Kitsuregawa, "Modeling user leniency and product popularity for sentiment classification," in *Proc. IJCNLP*, 2013, pp. 1107–1111.

[53] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 1555–1565, doi: 10.3115/v1/p14-1146.

[54] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[55] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 2931–2939.

[56] Q. Diao, M. Qiu, C. Y. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 193–202, doi: 10.1145/2623330.2623758.

[57] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751, doi: 10.3115/v1/d14-1181.

MUHAMMAD IBRAHIM received the Ph.D. degree in computer science from The Islamia University of Bahawalpur, Pakistan. He has more than ten years of teaching research experience in various institutes of Pakistan. He is currently a Lecturer with the Department of Computer Science, The Islamia University of Bahawalpur. His current research interests include artificial intelligence, the IoT, big data, and machine learning.

IMRAN SARWAR BAJWA received the Ph.D. degree in computer science from the University of Birmingham, U.K. He has more than 18 years of teaching research experience in various universities of Pakistan, Portugal, and U.K. He is currently an Associate Professor with the Department of Computer Science and Information Technology, The Islamia University of Bahawalpur. He has published more than 190 articles in well reputed peer-refereed journals and conferences. He has more than 80 publications and 1000 citations in Google Scholar. Additionally, he has more than 90 articles in Scopus and 370 citations of his work in Scopus. His personal impact factor is more than 40. His current research interests include artificial intelligence, the IoT, big data, and machine learning. He is an Associate Editor of IEEE Access journal. He has very good programming skills in Java and C#.

NADEEM SARWAR has been an Assistant Professor with the Department of Computer Science, Bahria University Lahore Campus, Lahore, Pakistan. He has ten years of teaching and research experience. He has published more than 55 international and national journals/conference publications in the last five years. He is also working as a reviewer of IEEE Access and *Proceedings of the Pakistan Academy of Sciences* (Pakistan Science Foundation), an Editorial Board Member of *Journal of Mathematics and Computer Science*, and an Academic Editor of *IET Software*, *PLOS One*, *Journal of Healthcare Engineering*, and *Security and Communication Networks*.

FAHIMA HAJJEJ received the Ph.D. degree in computer science from the Faculty of Sciences of Sfax, in 2017. She is currently an Assistant Professor with the Department of Information Systems, College of Computer and Information Sciences, PNU, Saudi Arabia. She is also a member of the Research Laboratory in Technologies of Information and Communication and Electrical Engineering (LaTICE). Her research interests include the modeling concepts of Education technology, artificial intelligence, big data, and deep learning.

HESHAM A. SAKR received the Ph.D. degree in communications and electronics engineering from the Faculty of Engineering, Mansoura University, Egypt, in 2020. He has been an Assistant Professor with the Electronics and Communication Department, Nile Higher Institute for Engineering and Technology, Mansoura, Egypt, since 2020. He has about 13 published articles in international journals. His current research interests include machine learning for emerging communication systems and applications, such as drone systems, the IoT, edge computing, caching, cybersecurity, smart cities, and vehicular networks.