Received: 25 April 2023 | Revised: 18 October 2023 | Accepted: 29 November 2023
DOI: 10.4218/etrij.2023-0162

ORIGINAL ARTICLE

ETRI Journal WILEY

Amazon product recommendation system based on a modified convolutional neural network

Yarasu Madhavi Latha | **B. Srinivasa Rao**

Department of Computer Science and Engineering, Koneru Lakshmaiah Educational Foundation, Vaddeswaram, India

Correspondence
Yarasu Madhavi Latha, Department of Computer Science and Engineering, Koneru Lakshmaiah Educational Foundation, Vaddeswaram, India.
Email: maadhaviresearch2020@gmail.com

Funding information
This research received no external funding.

Abstract

In e-commerce platforms, sentiment analysis on an enormous number of user reviews efficiently enhances user satisfaction. In this article, an automated product recommendation system is developed based on machine and deep-learning models. In the initial step, the text data are acquired from the Amazon Product Reviews dataset, which includes 60 000 customer reviews with 14 806 neutral reviews, 19 567 negative reviews, and 25 627 positive reviews. Further, the text data denoising is carried out using techniques such as stop word removal, stemming, segregation, lemmatization, and tokenization. Removing stop-words (duplicate and inconsistent text) and other denoising techniques improves the classification performance and decreases the training time of the model. Next, vectorization is accomplished utilizing the term frequency–inverse document frequency technique, which converts denoised text to numerical vectors for faster code execution. The obtained feature vectors are given to the modified convolutional neural network model for sentiment analysis on e-commerce platforms. The empirical result shows that the proposed model obtained a mean accuracy of 97.40% on the APR dataset.

KEYWORDS
convolutional neural network, e-commerce platform, machine learning, online products, sentiment analysis, singular value decomposition, vectorization

1 | INTRODUCTION

Recommendation systems play a crucial role in modern human life because of the rapid growth of the big data environment, particularly in the financial domain [1, 2]. The development of effective personalized e-commerce recommendation systems has gained attention among researchers owing to the growth in Internet transactions and e-commerce platforms [3]. The main conventional recommendation systems include hybrid recommendation, content-based recommendation, and collaborative filtering techniques. Classical collaborative filtering techniques effectively leverage the information between items and users, but are limited by cold starts and data sparsity problems [4, 5]. In addition, classical collaborative filtering techniques are lower-level techniques, which cannot learn the deeper representations of the items and users [6].

On the other hand, content-based recommendation techniques need vectorization methods to achieve better results. Handcrafted and lower-level features fail in practical applications because enormous amounts of user data are obtained from the Internet [7, 8]. In content-based recommendation techniques, analyzing heterogeneous data (labels, text, and images) is rich but computationally expensive [9, 10]. Hybrid recommendation techniques combine auxiliary information to alleviate the concerns of cold starts and data sparsity in conventional recommendation systems. The complex nature (uneven distribution, heterogeneous data, and multimodality) of auxiliary information makes hybrid recommendation techniques ineffective [11, 12]. In recent times, deep learning models have shown impressive performance in the fields of computer vision, signal processing, and natural language processing. Therefore, a new modified convolutional neural network (MCNN) model is introduced in this article for effective product recommendation.

The major contributions are as follows:

- An MCNN model for effective product recommendation is proposed that incorporates pretrained models (skip-gram and GloVe) in a word-embedding layer to reduce the dimensions of the feature vectors. The integration of skip grams and GloVe in a convolutional neural network (CNN) model creates an ensemble effect, which leads to improved and reliable recommendations by highlighting relevant features and words in product descriptions.
- The use of GloVe and skip-gram embeddings captures different word semantics (global co-occurrence statistics and syntactic similarities) that enhance the CNN model's understanding of the product context.
- A singular value decomposition (SVD) layer integrated into the MCNN model has relevant decomposition information and recommends products based on the similarity between users.

Existing articles related to sentiment analysis are surveyed in Section 2. The methodological details, numerical analysis, and conclusion are presented in Sections 3, 4, and 5, respectively.

2 | LITERATURE SURVEY

A literature review of the existing articles on the topic of sentiment analysis is presented in this section. This section is divided into two subsections: (i) deep-learning models and (ii) machine-learning models.

2.1 | Deep-learning models

Yang and others [13] implemented a new e-commerce product recommendation system based on a bidirectional gated recurrent unit (Bi-GRU) and CNN. To improve the sentiment features in the collected text, a sentiment dictionary was created to weigh the word vectors of the sentiment words. Then, the CNN and Bi-GRU models were applied to extract important context features, which were finally fed to the fully connected layer to classify the sentiment features. The time complexity was high in this e-commerce product recommendation system because of the hybridization of the deep-learning models.

Onan [14] integrated GloVe word embedding and TF-IDF to extract contextual feature vectors from the collected dataset. Then, a hybrid architecture (CNN with long short-term memory [LSTM]) was implemented for effective sentiment analysis. However, the integration of two or more models increases the developed system's complexity. Suresh and Belinda [15] integrated the Broyden–Fletcher Goldfarb–Shanno algorithm with a GRU model for effective product recommendation. While performing experiments with larger datasets, the developed model consumed more processing time, which was considered a major issue in this study.

Kolhe and others [16] used stop-word removal and lemmatization techniques to eliminate artifacts from the acquired Amazon data. Then, the feature selection was carried out utilizing a swarm intelligence-based optimization algorithm. Finally, a novel clustering algorithm was used to recommend relevant products. The extensive experimental analysis by means of different evaluation measures demonstrated that the developed model obtained superior performance in product recommendation. However, the developed model was ineffective in learning the nature of the product and customer behavior.

Shobana and Murali [17] first extracted the contextual and semantic features from the acquired data using a skip-gram technique. The obtained contextual and semantic information was fed into the LSTM network for sentiment analysis. The efficacy of the LSTM network was improved by optimizing its weight parameters using the adaptive particle swarm optimization algorithm. A conventional LSTM network requires more training data for effective learning, but it is computationally costly.

Zhang and others [18] used a collaborative filtering technique to exploit product and user ratings for effective sentiment analysis. However, the developed collaborative filtering technique solves cold-start problems to some extent. Shoja and Tabrizi [19] developed a latent Dirichlet allocation (LDA) technique for extracting discriminative attributes from a product category, and then a matrix factorization was utilized to predict the product rating. The developed system's performance was tested on the Amazon review dataset using different performance measures, but the developed matrix factorization suffers from sparsity problems.

Mandhula and others [20] initially performed data preprocessing utilizing lemmatization, stemming, and stop-word removal techniques. Furthermore, topic modeling was accomplished using possibilistic fuzzy c-means with LDA. The selected keywords were classified into three classes (neutral, positive, and negative) by implementing a CNN model. The high dimensionality of the acquired data slightly affects the performance of the CNN model. Hajek and others [21] integrated consumer emotions, word context, and a bag of words for fake review detection. In experiments, the developed system outperformed the baseline systems using f1-score, accuracy, and area under curve (AUC), but the time complexity was higher in the developed system.

Umer and others [22] analyzed the impact of fast text embedding and a CNN model on text classification. The results indicate that stacking several layers increases the complexity of the CNN model. In addition, Sachin and others [23] developed a gated recurrent neural network (RNN) for effective sentiment analysis. When analyzing enormous amounts of data, the gated RNN has two major concerns: vanishing and exploding gradient problems.

Alsayat [24] developed a new ensemble language model for sentiment analysis based on the LSTM network using an advanced word-embedding method. As mentioned earlier, the LSTM network requires more training data for efficient learning, which is computationally expensive. Ghasemi and Momtazi [25] used a collaborative filtering technique for precise recommendations; however, it has issues such as scalability, cold-start problems, and synonyms.

Onan [26] developed a bidirectional convolutional RNN with a group-wise enhancement for text classification. The results indicate that the inclusion of the group-wise enhancement process with the deep-learning model significantly outperformed the existing models in sentiment analysis; however, the main issue of the RNN is the vanishing gradient. Furthermore, Onan [27] developed a novel clustering technique to address the problem of class imbalance. In that study, a consensus clustering-based undersampling technique was implemented to balance the acquired datasets. Onan and others [28] ensembled different classifiers and keyword-extraction techniques to achieve efficient text classification. The empirical experimental analysis states that the integration of various keyword extraction techniques improves the scalability and predictive performance of the classifier, but it increases the classifier's complexity.

Onan [29] integrated numerous word-embedding techniques such as LDA2vec, word-position2vec, part-of-speech (POS)2vec, and word2vec for topic extraction. Furthermore, text clustering was carried out by an ensemble model that combines many clustering techniques such as self-organizing maps, k-means++, k-modes, and k-means. Additionally, Onan and Korukoğlu [30] developed a new model for classifying text sentiment by integrating many feature selection methods based on genetic rank aggregation. The integration of several word-embedding, text clustering, and feature selection methods increased the complexity of the framework. In addition, Onan and others [31] implemented an ensemble pruning method based on a multi-objective evolutionary algorithm and consensus clustering. In that study, the presented method's efficacy was validated on 12 unbalanced and balanced datasets, and an empirical examination showed the superiority and validity of the proposed method over existing methods.

Onan [32] presented an RNN model for opinion mining, and its performance was evaluated on a corpus of instructor evaluation reviews that includes 154 000 reviews. The RNN model yields better results than traditional deep-learning, ensemble-learning, and machine-learning models. Onan [33] analyzed the performance of deep-learning and text-mining models in sentiment analysis on a massive open-source dataset. Onan [34] analyzed the performance of different base learners (random forests, logistic regression, naïve Bayes, support vector machines (SVMs), and K-nearest neighbors (KNN)) and feature engineering methods (POS n-grams, character n-grams, linguistic features, and authorship attribution) in text genre classification. Onan and Toçoğlu [35] used a stacked bi-directional LSTM model for identifying sarcasm. In that study, the presented model achieved promising results in terms of accuracy.

Onan [36] implemented a topic-enriched word-embedding model for effective sarcasm identification. However, the presented model faced problems such as overfitting and vanishing gradients. Onan [37] performed biomedical text classification based on an optimized topic model and ensemble pruning. Additionally, Onan and others [38] integrated LDA with an improved ant-optimization algorithm for clustering text documents. The overlapping of unrelated topics or documents in conventional LDA degrades the performance of document clustering.

Onan [39] developed a novel text classification system that includes the following methods: contextual node embedding, hierarchical graphs, and dynamic fusion using bidirectional encoder representations from transformers (BERT). The developed system achieved high classification accuracy on benchmark datasets relative to existing baseline models. However, resource intensity and complexity were major problems noted in this study.

Onan [40] developed a new model for effective text augmentation that integrates genetic algorithms and graph-based neural networks. In that study, the developed model generates high-quality and diverse augmented data by exploring the high-dimensional feature space of the data. Additionally, Onan [41] integrated ant colony optimization (ACO) with semantic role labeling (SRL) for effective text augmentation. The developed framework utilizes SRL to identify a word's semantic roles in sentences and ACO generates new sentences to preserve the word's roles. The experimental outcomes indicate the efficiency of the SRL-ACO model in sarcasm identification, toxic text detection, and sentiment analysis, but it can only handle limited non-textual data.

2.2 | Machine-learning models

Zhao and others [42] initially extracted customer product reviews from e-commerce websites using a web-scraping tool. Then, the raw text was pre-processed using techniques such as snowball stemming, Gensim lemmatization, and tokenization. Furthermore, the contextual feature vectors were extracted from the denoised text utilizing the term weighting method. Then, the feature selection and classification were accomplished using an improved bat algorithm and Elman Neural Network (ENN). The ENN model classifies the sentiments of the customer reviews as neutral, positive, or negative. However, the developed ENN model is expensive because it requires a large number of customer reviews to obtain better classification performance.

Fauzi [43] developed a new model for sentiment analysis based on Word2Vec and SVM; however, the developed classification model supports only binary classification. Bansal and Srivastava [44] initially extracted the frequent trigrams and bigrams from a text corpus, and then feature extraction was accomplished by applying the TF-IDF and POS tagging techniques. Finally, the sentiment orientation of every review was determined based on the assumption of the labeled attributes. However, the developed model's performance was slightly degraded owing to factors such as data sparsity.

Gokalp and others [45] performed sentiment classification by implementing a wrapper algorithm called the iterated greedy algorithm. The numerical analysis shows that the developed iterated greedy algorithm produces better results than existing algorithms on four Amazon datasets. However, the developed iterated greedy algorithm must concentrate on other common problems, such as synonymy and scalability.

Hamdi [46] integrated an SVM with the ACO algorithm to classify the sentiments of customer reviews in online shopping. The developed system's performance was evaluated on two online datasets in terms of the f1-measure and accuracy. In contrast, the developed system consumes more computational time when classifying the sentiments of customer reviews.

2.3 | Summary

Overall, the existing literature highlights the significance of accurate and personalized recommendations for enhancing and driving sales and user experience on e-commerce platforms. Researchers continue to explore new techniques and algorithms (transfer learning, graph-based models, sequential recommendation models, matrix factorization with deep learning, and collaborative filtering) to improve the recommendation performance, ethical aspects, and interpretability of Amazon product recommendation systems. In order to address the aforementioned concerns, the MCNN model is introduced for effective product recommendations.

3 | METHODOLOGY

In this study, the development of the product recommendation system includes four steps: dataset description: the APR dataset; data denoising: stop-word removal, segregation, stemming, lemmatization, and tokenization; vectorization: the TF-IDF technique; and recommendation: MCNN. The workflow of the developed product recommendation system is presented in Figure 1.

3.1 | Dataset description

In this decade, the e-commerce websites Flipkart, Amazon, BookMyShow, India MART, Myntra, and Firstcry utilize dissimilar recommendation models to provide suggestions to users. Currently, Amazon utilizes deep learning models that rescale massive datasets and provide better recommendations [47, 48]. In this research article, a new automated product recommendation system is proposed for Amazon products. Here, the MCNN model's efficacy is tested on the APR dataset, which includes five attributes: text (the review), time-stamp (time of the rating), rating (product rating given by the user), product-ID (each product has a unique ID), and user-ID (each user has a unique ID). The APR dataset has a total of 60 000 reviews, of which 14 806, 25 627, and 19 567 are neutral, positive, and negative reviews, respectively.

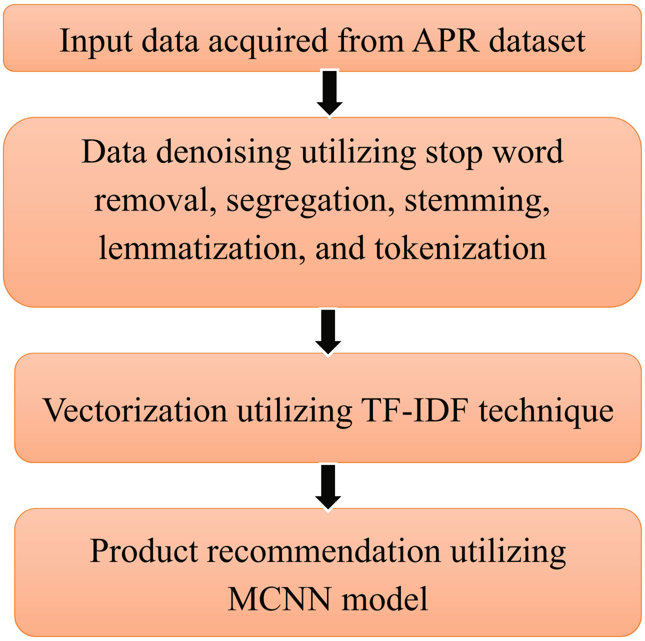

FIGURE 1 Workflow of the developed product recommendation system.

3.2 | Data denoising

After acquiring the product reviews from the APR dataset, data denoising is performed utilizing stop-word removal, stemming, lemmatization, and tokenization [49]. Initially, the collected customer reviews are categorized into words, symbols, tokens, and phrases. Then, stop words such as "should," "for," "the," "I," "and," and "is" are eliminated from the tokenized reviews by utilizing the stop-word list of the Natural Language Toolkit (NLTK). Morphological stemming is utilized to reduce the words to their base structures/forms. For example, the words "simplified" and "simplifying" are changed to their base form, "'simple." Furthermore, the lemmatization process breaks a word into its root meaning to identify its similarities. For example, the words "happiness" and "happy" are changed to "happ," which is meaningless but simple. At last, the segregation process eliminates these special characters: "?", "!", "#", "%", "&", "*", "+", "-", "[", "]", "] n", "^", "|", and "~" from the customer reviews.

3.3 | Vectorization

This section describes how vectorization is performed using the TF-IDF technique after the acquired customer reviews have been denoised. The TF is determined as the ratio of the number of repetitive words in a denoised customer review to the total number of words in the customer review. Correspondingly, the IDF is the ratio of the number of customer reviews to the total number of customer reviews with repetitive words [50, 51].

The mathematical expressions for the TF and IDF are given in (1)–(4), where the term frequency t in a customer review d is represented as $f_d(t)$ and the corpus of customer reviews is denoted as D. The obtained vectors are passed to the MCNN model to classify the opinions/sentiments of Amazon products as positive, negative, or neutral.

$$\mathrm{TF}(t, d) = \frac{f_d(t)}{\max\limits_{a \epsilon d} f_d(a)}, \tag{1}$$

$$\mathrm{IDF}(t, D) = \ln\left(\frac{|D|}{|\{d \epsilon D : t \epsilon d\}|}\right), \tag{2}$$

$$\text{TF-IDF}(t, d, D) = \mathrm{TF}(t, d) \times IDF(t, D), \tag{3}$$

$$\text{TF-IDF}'(t, d, D) = \frac{\mathrm{IDF}(t, D)}{|D|} + \textit{TF-IDF}(t,d,D). \tag{4}$$

3.4 | Product recommendation system

The vectors extracted from the TF-IDF technique are fed to the MCNN model for Amazon product recommendations. The proposed MCNN model has 12 layers: a one-word embedding layer, five convolutional layers, four pooling layers, one SVD layer, and one flattened layer. In the initial phase, the vectors extracted using the TF-IDF technique are transformed into fixed lengths in the word-embedding layer. This process reduces the dimensions of the extracted feature vectors and overcomes the curse of dimensionality [52, 53]. The word-embedding layer is a crucial component in deep-learning models, especially in natural language processing. It is designed to convert words or tokens into dense vector representations. Dense vector representations are known as word embeddings, and they capture contextual and semantic information about words that makes it easier for neural networks to understand the nature of text data. The word-embedding layer starts with a vocabulary containing all the unique words in the APR dataset; here, every word in the vocabulary is assigned an index. The word-embedding layer takes every word in a sentence or text sequence and looks up its corresponding index in the vocabulary while processing the text data. The word-embedding layer retrieves the pre-trained or trainable embedding vector associated with the word index once the index of a word has been determined.

In the word-embedding layer, the pretrained skip-grams and GloVe models are utilized to determine the relationship between customer reviews and products. The skip-gram and GloVe models predict the appropriate contextual words based on the targeted words, and this process helps find relevant ratings. In contrast, the information layer in skip-gram uses a one-hot encoded vector, which is similar in size to the size of the vocabulary dictionary. The vectors fed into the word-embedding layer are M-dimensional vectors. The word-embedding layer outputs a matrix, where every row corresponds to the word embeddings of the input words after embedding the vectors for all words in a sequence. The matrix is further passed to the subsequent layers of the neural network.

Furthermore, according to hierarchical softmax functions, the output layer of the MCNN model predicts the neighborhood words. This process reduces the training time and is computationally effective. To compute the probability in the hierarchical softmax functions, a binary tree structure is employed to predict the words. The probability of a word $\mathrm{pr}(w/w_{\mathrm{input}})$ is identified in the skip-gram model as

$$\mathrm{pr}\left(\frac{w}{w_{\mathrm{input}}}\right)=\prod_{j=1}^{L-1}\sigma\left(m(w,j+1)=child(m(w,j))v'_{m(w,j)}v_{w_{input}}\right), \tag{5}$$

where the input vector is represented as v, the input word is denoted as w_{input}, word representation of the output vector is denoted as v', the path length is denoted as L, the sigmoid activation function is specified as $\sigma()$, j^{th} the node of the binary tree is denoted as $m(w,j)$, and the child node is represented as $\mathrm{child}(m)$. The error that occurs during word prediction is mathematically expressed as

$$\mathrm{Error}=\frac{1}{N}\sum_{n=1}^{N}\sum_{-c\le j\le c}log\ pr\ (w_{n+j}|w_n), \tag{6}$$

where the context size is represented as c, the weight matrix function with the minimal error value is denoted by w, and the number of word sequences is denoted by N. Then, the max-pooling operation is performed in the pooling layer and is expressed as

$$o_{x,w}=\max_{(i,j)\in q_{x,w}} y_{i,j}, \tag{7}$$

where the pooling regions are represented as $y_{i,j}$, and the pooling operator is denoted by $o_{x,w}$. Around position (i,j), the pooling regions state the local neighborhoods $q_{x,w}(x,w)$. After the feature maps x have been extracted, a flatten layer is used to enhance the ability of non-linear mapping. The global features for classification are generated in the flatten layer by incorporating local features learned from the convolutional layers. The mathematical expression for the flatten layer is

$$y_j^{(q)}=r\left(\sum_{i=1}^{n}x_i^{(q-1)}\times w_{ij}^{(q)}+b^{(l)}\right), \tag{8}$$

where $w_{ij}^{(q)}$ represents the weight connections of neurons i and j, the activation function is denoted by r, the bias in the flattened layer is denoted by $b^{(l)}$, and the number of neurons is denoted by n. In this paper, the softmax layer with three neurons $\left[h^1, h^2, and\, h^3\right]$ represents three classes: positive, neutral, and negative, and it maps the output of many neurons to the range of zero to one. The mathematical expression of the softmax layer is

$$h^n=arg\ max\ \ \frac{e^{h^n}}{\sum_{n=1}^{3}e^{h^n}}\Bigg). \tag{9}$$

During data training and testing, the ReLU activation function is applied in the MCNN model to overcome the issues of vanishing and exploding gradients. The ReLU activation function $f(x)$ is mathematically expressed in (10). To further enhance the performance of the recommendation system, the MCNN model performs batch normalization $\widehat{x}^{(k)}$, which is mathematically expressed in (11).

$$f(x)=\begin{Bmatrix}0, & \text{if} \quad x>0\\ x, & \text{or} \quad \text{else}\end{Bmatrix}, \tag{10}$$

$$\widehat{x}^{(k)}=\frac{x^{(k)}-\mathrm{E}\left[x^{(k)}\right]}{\sqrt{\mathrm{Var}[x^{(k)}]}}. \tag{11}$$

Here, the standard deviation is represented as $\sqrt{\mathrm{Var}[x^{(k)}]}$, the mean of the k^{th} the neuron is specified as $\mathrm{E}\left[x^{(k)}\right]$, and the input data of the k^{th} the neuron is denoted as $x^{(k)}$. In batch normalization, the generalization capability of the MCNN model is enhanced by incorporating numerous constraints on the distributed data. The obtained mean and standard deviation values are rescaled to the range of zero to one. The learning parameters γ and β are utilized for data re-distribution $o^{(k)}$, as mathematically specified as follows:

$$\mathrm{o}^{(k)}=\gamma^{(k)}\widehat{x}^{(k)}+\beta^{(k)}, \tag{12}$$

where the variance and standard deviation values of the distributed data are represented by $\gamma^{(k)}$ and $\beta^{(k)}$, respectively. The mathematical expressions for the batch-normalized layer $\tilde{X}$ are

$$\mu = \frac{1}{N}\sum_{i=1}^{N} X'_i, \tag{13}$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (X'_i - \mu)^2, \tag{14}$$

$$X_i^{\text{norm}} = \frac{X'_i - \mu}{\sqrt{\sigma^2 + \varepsilon}}, \tag{15}$$

$$\tilde{X}_i = \gamma X_i^{\text{norm}} + \beta, \tag{16}$$

where the mean is represented as μ, the variance value is denoted by σ^2, the number of words in a sequence is denoted by N, the input data are represented as X'_i, the normalized data are specified as X_i^{norm}, and the epsilon value is denoted by ε. In the MCNN model, the SVD layer factorizes the weight matrix and replaces it with US, where U is the left unitary matrix and S is the singular value matrix. The experimental outcomes demonstrate that the replacing operations can further reduce the negative values in the sample space. The Euclidean distance between the samples is utilized to measure the feature expression changes in a sample space. The feature maps e_m and e_n of two dissimilar samples obtained in the flattening operation using the weight matrix are mathematically described as

$$p = e \times weight\ matrix, \tag{17}$$

$$q = e \times \text{US} \tag{18}$$

where p and q are specified as orthogonalized. A schematic of the MCNN model is shown in Figure 2. The pseudocode for the proposed MCNN model is given in the next section.

3.4.1 | Pseudocode of the MCNN model

ALGORITHM 1 Pseudocode of the MCNN model

Input: Word embeddings $w_1, w_2, \ldots w_n$
Output: Recommended Amazon products
Initialize input feature maps $map \in R$ to all zero
Initialize $\alpha 1$ and $\alpha 2$ as 0.4 and 0.6
$\mu 1 = exp(\alpha 1)/(exp(\alpha 1) + exp(\alpha 2))$
$\mu 2 = exp(\alpha 2)/(exp(\alpha 1) + exp(\alpha 2))$
For $i = 1\ to\ n$ **do**
  For $j = 1\ to\ n$ **do**
    For $dimension = 1\ to\ d$ **do**
      $Map[i][j][dimension] = \mu 1.wi[dimension] + \mu 2.wj[dimension]$
    End for
  End for
End for
Add the word-embedding layer in the CNN based on (5);
Perform max-pooling operation and flattening using (7) and (8);
Fit the CNN using the training data;
Pass predicted ratings to the SVD layer;
Based on the cosine similarity, recommend predicted ratings to similar users;
Recommend products to the test users.

In this scenario, the assumed parameters of the MCNN model are as follows: the optimizer is Adam; the loss function is categorical cross-entropy; the number of epochs is 500; the dropout rate is 0.5; the batch size is 128; the kernel sizes are 3, 5, and 7; the learning rate is 0.10; the window size is 15; the minimum word count is 2; and the embedding dimension is 200. The numerical analysis of the MCNN model in the product recommendation system is given in Section 4.

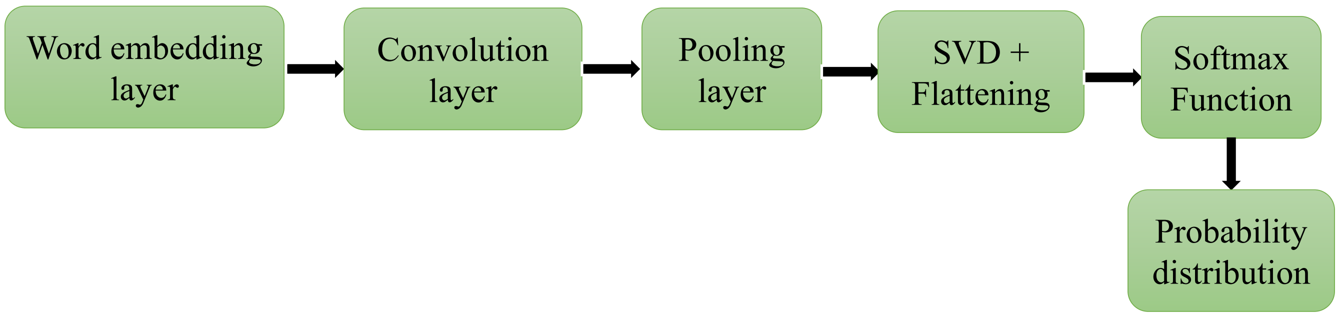

FIGURE 2 Architecture of the MCNN model.

4 | EXPERIMENTAL INVESTIGATION

In the product recommendation system, the proposed MCNN model was analyzed utilizing Python 3.7 with a Jupyter Notebook. The proposed MCNN model was validated on a system with a Windows operating system and 16 GB of RAM. Python libraries such as Pandas, NumPy, Matplotlib, SciKit Learn, TensorFlow, Keras, and NLTK were utilized for the experimental evaluation. The efficacy of the proposed MCNN model was tested on a benchmark dataset called the APR dataset using various evaluation measures, such as the mean squared error (MSE), root MSE (RMSE), mean absolute error (MAE), f-measure, accuracy, recall, and precision. The mathematical expressions for the error values MAE, RMSE, and MSE are given by

$$\mathrm{MAE} = \frac{1}{n}\sum\nolimits_{i=1}^{n} |(z_i - \widehat{z}_i)|, \tag{19}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n} (z_i - \widehat{z}_i)^2}, \tag{20}$$

$$\mathrm{MSE} = \frac{1}{n}\sum\nolimits_{i=1}^{n} (z_i - \widehat{z}_i)^2, \tag{21}$$

where the actual value is specified as z_i, the predicted value is denoted by $\widehat{z}_i$, and the total number of instances is specified as n. In addition, the F-measure has a single score that balances the concerns of recall and precision values in one number. The accuracy performance measure determines the degree of closeness between the true and measured values. On the other hand, the recall calibrates the number of positive predictions in the APR dataset, and precision evaluates the number of positive predictions belonging to the positive class. The performance measures, F-measure, accuracy, recall, and precision are mathematically expressed as

$$F\text{-}measure = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{22}$$

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{23}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{24}$$

$$Precision = \frac{TP}{TP + FP}, \tag{25}$$

Here, TN denotes true negatives, TP denotes true positives, FP denotes false positives, and FN denotes false negatives.

4.1 | Performance analysis

The results of the proposed MCNN model for various testing percentages according to recall, accuracy, and precision are listed in Table 1. By inspecting Table 1, it is clear that the efficacy of the proposed MCNN model and the existing models, CNN, and word-embedding CNN are validated for the Amazon datasets "Camera," "Kindle," "Media," "Electronics," "Home & Kitchen," "Book," "Cell Phones," and "Amazon Instant Video." Compared with the results obtained using 40% and 20% of the data for testing, the MCNN model obtained higher results when 30% of the data were tested.

The proposed MCNN model individually obtained accuracies of 97.20%, 96.65%, 96.36%, 96.68%, 95.99%, 96.16%, 96.41%, and 95.33% of accuracy on the Amazon datasets "Camera," "Kindle," "Media," "Electronics," "Home & Kitchen," "Books," "Cell Phones," and "Amazon Instant Video," respectively. The obtained results of the proposed MCNN model were 5% to 7% higher than those of the comparative models CNN and word-embedding CNN models. Correspondingly, the obtained results of the MCNN model for various testing percentages in terms of MSE, MAE, and RMSE are listed in Table 2. The proposed MCNN model has a minimum error value on the Amazon datasets camera," "Kindle," "Media," "Electronics," "Home & Kitchen," "Books," "Cell Phones," and "Amazon Instant Video," when 30% of the data are used for testing. The obtained results are better than those obtained by the comparative models (CNN and word-embedding CNN) and at different testing percentages (40% and 20% of the data used for testing).

4.2 | Ablation analysis

In this subsection, the ablation analysis is described for different feature extraction techniques (Word2Vec, BM25, and TF-IDF) and recommendation models (BRUCE, RNN, and random forest with gradient boosting machine [GBM]) in terms of accuracy, precision, and recall. By inspecting Table 3, it is clear that the combination TF-IDF and the MCNN model achieved the highest recommendation results on different datasets in terms of accuracy, precision, and recall, especially when 30% of the data were used for testing.

TABLE 1 Obtained results of the proposed MCNN model for various testing percentage according to recall, accuracy, and precision.

40% of the data used for testing									
	CNN (%)			Word-embedding CNN (%)			MCNN (%)		
Dataset	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Camera	91.88	93.32	92.95	93.45	95.13	94.90	94.85	96.13	96.30
Kindle	91.32	92.37	93.31	93.22	95.02	94.89	94.92	96.82	95.99
Media	91.94	93.21	94.16	94.16	95.04	94.81	95.76	96.44	95.81
Electronics	91.76	92.12	93.50	94.06	94.62	95.26	95.06	96.52	96.76
Home & Kitchen	92.32	92.75	93.53	93.89	94.20	94.45	95.59	95.80	95.85
Books	91.13	92.78	94.47	94.26	94.45	94.38	95.86	95.85	95.68
Cell Phones	92.88	92.76	93.81	93.61	94.39	95.54	95.01	95.79	96.54
Amazon Instant Video	90.83	92.43	93.43	92.68	93.60	93.50	94.38	95.10	95.20
30% of the data used for testing									
	CNN (%)			Word-embedding CNN (%)			MCNN (%)		
Dataset	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Camera	92.45	93.62	94.97	96.17	95.85	96.30	97.20	97.70	98.50
Kindle	92.46	93.33	95.66	95.02	96.23	95.85	96.65	97.35	98.15
Media	92.79	94.06	95.88	95.36	96.64	95.36	96.36	97.36	98.06
Electronics	92.81	93.40	95	94.41	96.45	95.68	96.68	97.48	97.98
Home & Kitchen	93.24	93.47	95.82	95.18	96.17	95.39	95.99	96.49	97.39
Books	92.04	93.25	95.21	95.23	95.85	95.26	96.16	96.86	97.76
Cell Phones	92.44	94.60	95.16	95.60	95.95	95.61	96.41	96.91	97.81
Amazon Instant Video	91.63	93.13	94.65	94.79	95.01	94.83	95.33	96.33	97.23
20% of testing									
	CNN (%)			Word-embedding CNN (%)			MCNN (%)		
Dataset	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Camera	94	92.74	92.32	95.51	94.87	95.60	96.40	97.40	98.20
Kindle	93.31	92.79	93.43	95.28	95.29	95.01	96.01	96.71	97.31
Media	93.46	93.56	92.39	95.78	94.83	95.82	96.52	97.42	98.12
Electronics	93.44	93.46	93.28	95.83	95.22	94.29	94.99	95.59	96.09
Home & Kitchen	93.13	93.94	93.16	95.14	94.59	94.43	95.23	95.83	96.33
Books	93.97	92.31	92.65	94.85	94.48	94.82	95.62	96.22	96.82
Cell Phones	93.30	92.94	93.70	95.66	95.21	95.27	96.17	96.87	97.57
Amazon Instant Video	92.65	92.40	92.20	94.35	94.40	94.10	95	96	96.90

Abbreviations: CNN: convolutional neural network; MCNN: modified convolutional neural network.

The TF-IDF technique is simpler and easier to use in product recommendation systems than other feature extraction techniques such as Word2Vec and BM25. The results of traditional machine-learning approaches (BRUCE, RNN, and random forest with GBM) are presented in Table 4. By analyzing Table 4, it is clear that the existing machine-learning approaches achieved much smaller results than the MCNN model. The proposed MCNN model learns both long- and short-term dependencies to solve the curse of dimensionality. Additionally, the inclusions of a word embedding layer (skip-grams and GloVe) and an SVD layer simplify customer data and recommend Amazon products to users based on the similarity between users. Additionally, the efficacy of the MCNN model is analyzed by varying the number of iterations. As presented in Table 5, the proposed MCNN model obtains the highest results at 500 iterations. The proposed MCNN

TABLE 2 Obtained results of the MCNN model for various testing percentages according to MSE, MAE, and RMSE.

40% of the data used for testing									
	CNN			Word-embedding CNN			MCNN		
Dataset	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
Camera	1.44	2.37	2.02	1.43	2.36	2.01	1.43	2.36	2.01
Kindle	1.78	2.46	1.67	1.78	2.46	1.67	1.77	2.45	1.66
Media	1.67	2.88	1.94	1.67	2.88	1.93	1.67	2.87	1.92
Electronics	1.38	3.10	1.79	1.38	3.10	1.78	1.37	3.10	1.78
Home & Kitchen	1.68	3.39	2.23	1.68	3.38	2.23	1.68	3.37	2.23
Books	1.40	2.70	1.75	1.39	2.70	1.74	1.39	2.69	1.74
Cell phones	2.03	3.85	2.80	2.03	3.85	2.80	2.02	3.84	2.79
Amazon Instant Video	2.26	3.43	2.90	2.25	3.42	2.90	2.25	3.41	2.90
30% of the data used for testing									
	CNN			Word-embedding CNN			MCNN		
Dataset	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
Camera	0.06	0.01	0.01	0.06	0.08	0.04	0.05	0.01	0.05
Kindle	1.50	2.43	2.06	1.49	2.42	2.05	1.49	2.42	1.05
Media	1.84	2.51	1.75	1.83	2.50	1.75	1.22	2.49	1.74
Electronics	1.68	3.48	2.27	1.67	3.47	2.26	1.17	3.07	1.25
Home & Kitchen	1.71	2.97	2.02	1.71	2.96	2.01	1.33	2.95	2.01
Books	1.46	3.19	1.82	1.46	3.19	1.82	1.25	2.19	1.31
Cell phones	1.45	2.77	1.79	1.44	2.76	1.79	1.44	2.76	1.79
Amazon Instant video	2.13	3.88	2.82	2.12	3.87	2.82	2.11	3.16	2.81
20% of the data used for testing									
	CNN			Word-embedding CNN			MCNN		
Dataset	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
Camera	0.11	0.05	0.04	0.11	0.05	0.04	0.10	0.04	0.04
Kindle	1.58	2.52	2.15	1.58	2.51	2.15	1.57	2.51	2.14
Media	1.86	2.54	1.82	1.86	2.53	1.81	1.85	2.54	1.81
Electronics	1.72	3.54	2.27	1.71	3.53	2.26	1.71	3.53	2.26
Home & Kitchen	1.71	2.99	2.11	1.71	2.99	2.10	1.73	2.98	2.10
Books	1.57	3.28	1.87	1.57	3.28	1.86	1.56	3.27	1.86
Cell phones	1.48	2.80	1.89	1.48	2.80	1.89	1.47	2.80	1.88
Amazon Instant Video	2.21	3.91	2.86	2.21	3.91	2.86	2.20	3.91	2.86

Abbreviations: CNN, convolutional neural network; MAE, mean average error; MCNN, modified CNN; MSE, mean squared error; RMSE, root MSE.

model has a mean F-measure of 81.20%, a recall of 76.70%, a precision of 86.90%, an accuracy of 97.40%, and a cumulative gain of 2.883 on the APR dataset. Increasing the number of iterations indicates that a large amount of data is utilized for the experimental examination. However, increasing the number of iterations does not improve the accuracy of the product recommendations.

4.3 | Comparative analysis and discussion

The effectiveness of the proposed MCNN model was compared with the comparative models developed by Shobana and Murali [17] and Zhang and others [18]. Shobana and Murali [17] implemented a skip-gram technique to extract semantic and contextual feature vectors

TABLE 3 Obtained results of different feature extraction techniques with the MCNN model in terms of accuracy, recall, and precision.

30% the data used for of testing									
	Word2Vec (%)			BM25 (%)			TF-IDF (%)		
Dataset	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Camera	93.44	92.66	92.05	95.34	93.88	95.20	97.20	97.70	98.50
Kindle	92.95	92.57	92.47	94.51	95.24	94.09	96.65	97.35	98.15
Media	93.26	93.37	91.45	95.34	94.70	95.47	96.36	97.36	98.06
Electronics	93.14	93.30	92.35	95.16	94.46	93.88	96.68	97.48	97.98
Home & Kitchen	93.02	93.79	93.14	95.03	94.52	94.38	95.99	96.49	97.39
Books	93.60	91.91	91.96	94.08	94.38	94.80	96.16	96.86	97.76
Cell Phones	92.98	92.91	93.02	95.17	94.87	95.25	96.41	96.91	97.81
Amazon Instant Video	92.34	92.33	91.57	94.17	94.06	93.24	95.33	96.33	97.23

Abbreviations: BM25, best match 25; TF-IDF, term frequency-inverse document frequency.

TABLE 4 Obtained results of different machine-learning approaches in terms of accuracy, recall, and precision.

30% the data used for of testing									
	BRUCE (%)			RNN (%)			Random forest with GBM (%)		
Dataset	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Camera	92.75	92.09	91.63	93.53	92.41	92.44	94.22	92.74	91.58
Kindle	92.82	92.41	92.39	92.99	92.75	92.44	92.66	91.38	92.14
Media	92.80	93.23	90.79	92.90	93.73	90.81	93.35	93.34	94.09
Electronics	92.73	92.71	92.23	93.22	93.34	92.75	91.64	92.47	93.94
Home & Kitchen	92.27	93.27	92.69	92.39	93.59	92.91	93.98	93.22	92.30
Books	92.67	91.37	91.71	93.46	91.87	92.44	92.89	92.83	93.32
Cell phones	92.68	92.54	92.40	93.24	92.56	92.89	93.42	94.90	93.80
Amazon Instant Video	92.25	92	90.96	93.01	92.52	91	92.90	93.11	92.20

Abbreviations: BRUCE, bundle recommendation using contextualized item embeddings; GBM, gradient boosting machine; RNN, recurrent neural network.

TABLE 5 Results of the proposed MCNN model at various iteration numbers.

Iteration	F-measure (%)	Recall (%)	Precision (%)	Accuracy (%)	Cumulative gain
50	76	72.30	81.40	91.70	2.774
100	76.50	73.40	82.10	92.50	2.776
150	77.20	73.60	82.30	92.80	2.776
200	77.80	73.80	82.50	93.40	2.785
250	78.10	74.10	83.30	93.70	2.791
300	78.30	75.20	84	94	2.838
350	78.70	76	85.70	94.90	2.852
400	80.40	76.10	86	95.60	2.883
450	80.40	76.30	86.80	97	2.883
500	81.20	76.70	86.80	97.40	2.883

from the APR dataset. The extracted semantic and contextual feature information was provided to the SVM, Artificial Neural Network (ANN), LSTM, and APSO-LSTM models for product-sentiment analysis. By inspecting Table 6, it is clear that the proposed MCNN model obtained better recommendation results than the

TABLE 6 F-measure, recall, precision, and accuracy of the MCNN model and comparative models.

Model	F-measure (%)	Recall (%)	Precision (%)	Accuracy (%)
SVM [17]	75.60	73	81	91.80
ANN [17]	76.80	72	71.90	81.90
LSTM [17]	78.57	74.50	83	94.10
APSO-LSTM [17]	80.04	76.08	85.28	96.80
MCNN	81.20	76.70	86.80	97.40

Abbreviations: ANN, artificial neural network; APSO, adaptive particle swarm optimization; LSTM, long short-term memory; MCNN, modified convolutional neural network; SVM, support vector machine.

TABLE 7 MAE and RMSE values of the MCNN and UTER models.

	MAE		RMSE	
Dataset	UTER [23]	MCNN	UTER [23]	MCNN
Baby	0.79	0.77	1.01	0.98
Video Games	0.79	0.77	1.01	0.99
Pet Supplies	0.85	0.84	1.06	0.99
Sports and Outdoors	0.61	0.58	0.85	0.83

Abbreviations: MAE, mean average error; MCNN, modified convolutional neural network; RMSE, root mean squared error.

comparative models in terms of f-measure, recall, precision, and accuracy. On the APR dataset, the proposed MCNN model achieved a mean f-measure of 81.20%, recall of 76.70%, precision of 86.80%, and accuracy of 97.40%, which are superior to those of the comparative models. On the other hand, Zhang and others [18] integrated the unifying paragraph embedding layer with a collaborative filtering technique (UTER) for effective Amazon product recommendations. As presented in Table 7, the proposed MCNN model obtained minimum MAE results of 0.77, 0.77, 0.84, and 0.58, and RMSE results of 0.98, 0.99, 0.99, and 0.83, respectively, on the Amazon datasets "Baby," "Video Games," "Pet Supplies," and "Sports and Outdoors." The obtained error value of the proposed MCNN model was limited compared with that of the UTER model.

As detailed earlier, the proposed MCNN model includes a word-embedding layer (skip-grams and GloVe) that reduces the dimensions of the feature vectors extracted using the TF-IDF technique. Correspondingly, the SVD layer effectively simplifies the data and removes the noise. The SVD layer has appropriate decomposition information, which recommends the appropriate products to users based on the similarity between users. Additionally, in the context of text classification, the percentage of training time is reduced when stop-word removal is performed in a neural network. However, providing an exact percentage of training time reduction is challenging because removing stop words results in smaller input data and requires less processing during every training iteration, especially for large datasets. Furthermore, the high proportion of stop words in the review text leverages parallelism for graphics processing unit-based processing. The proposed model experiences more significant time reductions on cleaner, more informative text, and larger batch sizes, which offset the effect of stop word removal, because this allows more efficient data processing. As a result, for every epoch, the training time of the neural network significantly reduced from 143 to 116 s, which is 10–20% on average.

5 | CONCLUSION

This research paper introduced an efficient personalized recommendation system for e-commerce platforms based on a deep-learning model. The proposed framework includes three major steps: denoising, vectorization, and sentiment analysis. From the acquired raw data, the noisy, inconsistent, and duplicate text is eliminated by implementing stop-word removal, stemming, segregation, lemmatization, and tokenization techniques. The training time of the classification model is reduced by converting text into feature vectors. Here, TF-IDF is employed as a feature extraction technique. Finally, the obtained vectors are passed to the MCNN for sentiment analysis. The efficacy of the proposed model was analyzed using evaluation measures MSE, MAE, RMSE, f-measure, accuracy, recall, and precision. The empirical results demonstrate that the proposed model outperforms comparative models in terms of accuracy. On the other hand, the proposed model achieved a mean accuracy of 97.40% on the APR dataset with a limited computational time of 44 s and linear complexity. In the future, this research can be extended to multimedia data by implementing an ensemble deep-learning model. In addition, the proposed model could be validated on real-time practical datasets with larger data sizes.

ACKNOWLEDGMENTS

This research received no external funding.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The datasets generated and/or analyzed during the current study are available from the Amazon Product Reviews Dataset repository, https://snap.stanford.edu/data/web-Amazon.html.

ORCID

Yarasu Madhavi Latha https://orcid.org/0000-0002-7494-0295

REFERENCES

1. R. Liang and J. Wang, *A linguistic intuitionistic cloud decision support model with sentiment analysis for product selection in e-commerce*, Int. J. Fuzzy Syst. **21** (2019), no. 3, 963–977. DOI 10.1007/s40815-019-00606-0.
2. Q. Sun, J. Niu, Z. Yao, and H. Yan, *Exploring eWOM in online customer reviews: sentiment analysis at a fine-grained level*, Eng. Appl. Artif. Intel. **81** (2019), 68–78. DOI 10.1016/j.engappai.2019.02.004.
3. B. Ozyurt and M. A. Akcayol, *A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: SS-LDA*, Expert Syst. Appl. **168** (2021), 114231. DOI 10.1016/j.eswa.2020.114231.
4. K. Wang, T. Zhang, T. Xue, Y. Lu, and S.-G. Na, *E-commerce personalized recommendation analysis by deeply-learned clustering*, J. Visual Commun. Image Represent. **71** (2020), 102735. DOI 10.1016/j.jvcir.2019.102735.
5. A. Iftikhar, M. A. Ghazanfar, M. Ayub, Z. Mehmood, and M. Maqsood, *An improved product recommendation method for collaborative filtering*, IEEE Access **8** (2020), 123841–123857. DOI 10.1109/ACCESS.2020.3005953.
6. S. G. K. Patro, B. K. Mishra, S. K. Panda, R. Kumar, H. V. Long, D. Taniar, and I. Priyadarshini, *A hybrid action-related K-nearest neighbour (HAR-KNN) approach for recommendation systems*, IEEE Access **8** (2020), 90978–90991. DOI 10.1109/ACCESS.2020.2994056.
7. M. Shaheen, S. M. Awan, N. Hussain, and Z. A. Gondal, *Sentiment analysis on mobile phone reviews using supervised learning techniques*, Int. J. Mod. Educ. Comput. Sci. **11** (2019), no. 7, 32–43. DOI 10.5815/ijmecs.2019.07.04.
8. T. U. Haque, N. N. Saber, and F. M. Shah, *Sentiment analysis on large scale Amazon product reviews*, In *Proc. IEEE International Conference on Innovative Research and Development [ICIRD], Bangkok, Thailand*, 2018, 1–6.
9. S. Wassan, X. Chen, T. Shen, M. Waqar, and N. Z. Jhanjhi, *Amazon product sentiment analysis using machine learning techniques*, Rev. Argentina Clín. Psicológica **30** (2021), no. 1, 695. DOI 10.24205/03276716.2020.2065.
10. F. Zhang, X. Hao, J. Chao, and S. Yuan, *Label propagation-based approach for detecting review spammer groups on e-commerce websites*, Knowl. Based Syst. **193** (2020), 105520. DOI 10.1016/j.knosys.2020.105520.
11. A. S. Ghabayen and B. H. Ahmed, *Polarity analysis of customer reviews based on part-of-speech subcategory*, J. Intell. Syst. **29** (2020), no. 1, 1535–1544. DOI 10.1515/jisys-2018-0356.
12. F. Xu, Z. Pan, and R. Xia, *E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework*, Inf. Process. Manage. **57** (2020), no. 5, 102221. DOI 10.1016/j.ipm.2020.102221.
13. 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 **8** (2020), 23522–23530. DOI 10.1109/ACCESS.2020.2969854.
14. A. Onan, *Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks*, Concurrency Comput. Pract. Exper. **33** (2021), no. 23, e5909. DOI 10.1002/cpe.5909.
15. A. Suresh and M. J. C. M. Belinda, *Online product recommendation system using gated recurrent unit with Broyden fletcher Goldfarb Shanno algorithm*, Evol. Intell. **15** (2022), no. 3, 1861–1874. DOI 10.1007/s12065-021-00594-x.
16. L. Kolhe, A. K. Jetawat, and V. Khairnar, *Robust product recommendation system using modified grey wolf optimizer and quantum inspired possibilistic fuzzy C-means*, Cluster Comput. **24** (2021), no. 2, 953–968. DOI 10.1007/s10586-020-03171-6.
17. J. Shobana and M. Murali, *An efficient sentiment analysis methodology based on long short-term memory networks*, Complex Intell. Syst. **7** (2021), no. 5, 2485–2501. DOI 10.1007/s40747-021-00436-4.
18. Y. Zhang, Z. Liu, and C. Sang, *Unifying paragraph embeddings and neural collaborative filtering for hybrid recommendation*, Appl. Soft Comput. **106** (2021), 107345. DOI 10.1016/j.asoc.2021.107345.
19. B. M. Shoja and N. Tabrizi, *Customer reviews analysis with deep neural networks for e-commerce recommender systems*, IEEE Access **7** (2019), 119121–119130. DOI 10.1109/ACCESS.2019.2937518.
20. T. Mandhula, S. Pabboju, and N. Gugulotu, *Predicting the customer's opinion on Amazon products using selective memory architecture-based convolutional neural network*, J. Supercomput. **76** (2020), no. 8, 5923–5947. DOI 10.1007/s11227-019-03081-4.
21. P. Hajek, A. Barushka, and M. Munk, *Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining*, Neural Comput. Appl. **32** (2020), no. 23, 17259–17274. DOI 10.1007/s00521-020-04757-2.
22. M. Umer, Z. Imtiaz, M. Ahmad, M. Nappi, C. Medaglia, G. S. Choi, and A. Mehmood, *Impact of convolutional neural*

network and FastText embedding on text classification, Multimedia Tools Appl. **82** (2023), no. 4, pp. 5569–5585. DOI 10.1007/s11042-022-13459-x.
23. S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, and P. Nagrath, *Sentiment analysis using gated recurrent neural networks*. SN Comput. Sci. **1** (2020), no. 2, 74. DOI 10.1007/s42979-020-0076-y.
24. A. Alsayat, *Improving sentiment analysis for social media applications using an ensemble deep learning language model*, Arab. J. Sci. Eng. **47** (2022), no. 2, 2499–2511. DOI 10.1007/s13369-021-06227-w.
25. N. Ghasemi and S. Momtazi, *Neural text similarity of user reviews for improving collaborative filtering recommender systems*, Electron. Commer. Res. Appl. **45** (2021), 101019. DOI 10.1016/j.elerap.2020.101019.
26. A. Onan, *Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification*, J. King Saud Univ. - Comput. Inform. Sci. **34** (2022), no. 5, 2098–2117.
27. A. Onan, *Consensus clustering-based undersampling approach to imbalanced learning*, Sci. Program. **2019** (2019), 1–14.
28. A. Onan, S. Korukoğlu, and H. Bulut, *Ensemble of keyword extraction methods and classifiers in text classification*, Expert Syst. Appl. **57** (2016), 232–247.
29. A. Onan, *Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering*, IEEE Access **7** (2019), 145614–145633.
30. A. Onan and S. Korukoğlu, *A feature selection model based on genetic rank aggregation for text sentiment classification*, J. Inf. Sci. **43** (2017), no. 1, 25–38.
31. A. Onan, S. Korukoğlu, and H. Bulut, *A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification*, Inf. Process. Manage. **53** (2017), no. 4, 814–833.
32. A. Onan, *Mining opinions from instructor evaluation reviews: a deep learning approach*, Comput. Appl. Eng. Educ. **28** (2020), no. 1, 117–138.
33. A. Onan, *Sentiment analysis on massive open online course evaluations: a text mining and deep learning approach*, Comput. Appl. Eng. Educ. **29** (2021), no. 3, 572–589.
34. A. Onan, *An ensemble scheme based on language function analysis and feature engineering for text genre classification*, J. Inf. Sci. **44** (2018), no. 1, 28–47.
35. A. Onan and M. A. Toçoğlu, *A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification*, IEEE Access **9** (2021), 7701–7722.
36. A. Onan, *Topic-enriched word embeddings for sarcasm identification, in Software Engineering Methods in Intelligent Algorithms*, In *Proceedings of 8th Computer Science On-line Conference 2019*, Vol. **18**, Springer International Publishing, 2019, 293–304.
37. A. Onan, *Biomedical text categorization based on ensemble pruning and optimized topic modelling*, Comput. Math. Methods Med. **2018** (2018), 2497471.
38. A. Onan, H. Bulut, and S. Korukoglu, *An improved ant algorithm with LDA-based representation for text document clustering*, J. Inf. Sci. **43** (2017), no. 2, 275–292.
39. A. Onan, *Hierarchical graph-based text classification framework with contextual node embedding and BERT-based dynamic fusion*, J. King Saud Univ. – Comput. Inform. Sci. **35** (2023), 101610.
40. A. Onan, *GTR-GA: harnessing the power of graph-based neural networks and genetic algorithms for text augmentation*, Expert Syst. Appl. (2023), 120908.
41. A. Onan, *SRL-ACO: a text augmentation framework based on semantic role labeling and ant colony optimization*, J. King Saud Univ. - Comput. Inform. Sci. **35** (2023), 101611.
42. H. Zhao, Z. Liu, X. Yao, and Q. Yang, *A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach*, Inf. Process. Manage. **58** (2021), no. 5, 102656. DOI 10.1016/j.ipm.2021.102656.
43. M. A. Fauzi, *Word2Vec model for sentiment analysis of product reviews in Indonesian language*, Int. J. Electr. Comput. Eng. **9** (2019), no. 1, 525–530. DOI 10.11591/ijece.v9i1.pp525-530.
44. B. Bansal and S. Srivastava, *Hybrid attribute based sentiment classification of online reviews for consumer intelligence*. Appl. Intell. **49** (2019), no. 1, 137–149. DOI 10.1007/s10489-018-1299-7.
45. O. Gokalp, E. Tasci, and A. Ugur, *A novel wrapper feature selection algorithm based on iterated greedy metaheuristic for sentiment classification*, Expert Syst. Appl. **146** (2020), 113176. DOI 10.1016/j.eswa.2020.113176.
46. M. Hamdi, *Affirmative ant colony optimization based support vector machine for sentiment classification*, Electronics **11** (2022), no. 7, 1051. DOI 10.3390/electronics11071051.
47. J. Woo and M. Mishra, *Predicting the ratings of Amazon products using big data*, Wiley Interdiscip. Rev.: Data min. Knowl. Discov. **11** (2021), no. 3, e1400. DOI 10.1002/widm.1400.
48. S. Dey, S. Wasif, D. S. Tonmoy, S. Sultana, J. Sarkar, and M. Dey, *A comparative study of support vector machine and naive Bayes classifier for sentiment analysis on Amazon product reviews*, In *2020 International Conference on Contemporary Computing and Applications [IC3A], Lucknow, India*, 2020, 217–220.
49. T. Bezdan, C. Stoean, A. A. Naamany, N. Bacanin, T. A. Rashid, M. Zivkovic, and K. Venkatachalam, *Hybrid fruit-fly optimization algorithm with k-means for text document clustering*, Mathematics **9** (2021), no. 16, 1929. DOI 10.3390/math9161929.
50. S. W. Kim and J. M. Gil, *Research paper classification systems based on TF-IDF and LDA schemes*. Hum.-Centric Comput. Inform. Sci. **9** (2019), 30. DOI 10.1186/s13673-019-0192-7.
51. Z. Zhu, J. Liang, D. Li, H. Yu, and G. Liu, *Hot topic detection based on a refined TF-IDF algorithm*, IEEE Access **7** (2019), 26996–27007. DOI 10.1109/ACCESS.2019.2893980.
52. S. Tao, C. Shen, L. Zhu, and T. Dai, *SVD-CNN: a convolutional neural network model with orthogonal constraints based on SVD for context-aware citation recommendation*, Comput. Intell. Neurosci. 2020 (**2020**), 5343214. DOI 10.1155/2020/5343214.
53. D. Liu, X. Lai, Z. Xiao, D. Liu, X. Hu, and P. Zhang, *Fault diagnosis of rotating machinery based on convolutional neural network and singular value decomposition*, Shock Vib. 2020 (**2020**), 6542913. DOI 10.1155/2020/6542913.

AUTHOR BIOGRAPHIES

Yarasu Madhavi Latha is pursuing her PHD in computer science and engineering from Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, India. She did MTech (CSE) from Jawaharlal Nehru Technological University, Kakinada, and MCA from Acharya Nagarjuna University, Guntur, India. She has 15 years' of experience in teaching. She is currently working as an assistant professor in the Department of Computer Science and Engineering at Rise Krishna Sai Gandhi Group of Institutions, Ongole, AP, India. She has presented two research papers in International Conferences and published two technical papers in International Journals and two papers in National Journals. She has one patent in Big Data Processing in Cloud Environment.

B. Srinivasa Rao has obtained his PHD in computer science and engineering from Acharya Nagarjuna University, Guntur, AP, India. He has done MTech (CSE) from Jawaharlal Nehru Technological University, Anantapur, and Btech from VR Siddhartha College of Engineering, Vijayawada, AP, India. He has 30 years' of experience in teaching. He is currently working as a professor in the Department of Computer Science and Engineering at Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, India. He has presented five research papers in International Conferences and published 26 technical papers in International Journals. He has two patents in Digital Image Processing.

How to cite this article: Y. M. Latha and B. S. Rao, *Amazon product recommendation system based on a modified convolutional neural network*, ETRI Journal **46** (2024), 633–647, DOI 10.4218/etrij.2023-0162.