International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Deep Learning-Based Hybrid Recommendation System with NLP-HAEC-Based Sentiment Analysis

*B. Madhurika[1], *D. Naga Malleswari[2]

Submitted: 26/01/2024 **Revised**: 04/03/2024 **Accepted**: 12/03/2024

Abstract: Customer feedback analysis is pivotal in improving products and services, enhancing customer satisfaction, and enabling personalized recommendations. The exponential growth of customer feedback data necessitates automated techniques for efficient analysis. However, the conventional methods failed to provide the perfect recommendations to the users due to cold-start problems of user interactions. So, this research aims to leverage the Hybrid Recommendation System Network (HRS-Net) using deep learning and natural language processing (NLP) techniques to analyse customer feedback. Initially, NLP-based Hybrid Agglomerative Elbow Clustering (HAEC) is introduced to cluster the user data and items separately based on multilevel user-item interactions. Then, the Convolutional Recurrent Neural Network (CRNN) is trained with the HAEC features, which learns the user feedback based on its textual features. Finally, the CRNN provides accurate suggestions to the users based on the pre-trained HAEC interactions. Finally, the simulation results show that the proposed method resulted in superior performance compared to traditional machine learning, clustering, and filtering-based recommendation systems. The proposed HRS-Net achieved 98.54% accuracy, 98.38% precision, 98.63% recall, and 98.45% F1-score, which are higher than traditional approaches.

***Keywords**: Hybrid Recommendation System, natural language processing, Hybrid Agglomerative Elbow Clustering, Deep Learning Convolutional Neural Network*

1. Introduction

Customer feedback is crucial in understanding user preferences, improving product/service quality, and enhancing customer satisfaction [1]. As the volume of customer feedback continues to grow exponentially with the advent of online platforms and social media, manual analysis becomes impractical and inefficient. Therefore, there is a pressing need for automated techniques that can effectively analyze and extract valuable insights from large-scale customer feedback data [2]. The emergence of deep learning and NLP techniques has revolutionized the field of text analysis. Deep learning models, such as CNNs, recurrent neural networks (RNNs) [3] and transformers, have remarkably performed in various NLP tasks, including sentiment analysis, topic modelling, and text classification. These models can effectively capture complex patterns and relationships within textual data, enabling more accurate and meaningful analysis [4].The HRS combine different recommendation approaches, such as collaborative filtering [5], content-based filtering [6], and knowledge-based systems [7], to provide personalized and diverse recommendations. Customer feedback data can be a valuable source of information for HRS, as it contains explicit and implicit user preferences that can be leveraged to improve recommendation accuracy and relevance. Deep learning and NLP techniques can contribute significantly to text analysis in the context of customer feedback for HRS. Deep learning models can learn distributed representations of words and phrases, capturing their semantic meanings [8].

It enables a more nuanced understanding of customer feedback, including detecting sentiment and opinion and identifying important topics. HRS can better comprehend user preferences by analyzing customer feedback at a deeper semantic level, leading to more accurate recommendations. The NLP techniques, such as contextual word embeddings (e.g., BERT, GPT), can capture the contextual information of words within sentences [9]. This contextual understanding allows for improved sentiment analysis by considering negation, sarcasm, and other linguistic nuances that impact the interpretation of feedback. Incorporating contextual analysis into HRS can enhance the system's ability to understand the subtleties of customer feedback, resulting in more precise and context-aware recommendations. Deep learning models can automatically learn relevant features from customer feedback data, reducing the reliance on manual feature engineering. It is particularly advantageous in the case of unstructured text data, where identifying informative features can be challenging [10]. HRS can uncover valuable insights and patterns contributing to better recommendation quality by extracting high-level features from customer feedback. Deep learning models can assist in building user profiles

[1] *Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, India 522302
email: madhurikab1@gmail.com
[2] *Department. of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, India 522302
email: nagamalleswary@kluniversity.in

based on customer feedback [11]. HRS can create more comprehensive and accurate user representations by analyzing the sentiment, preferences, and topics discussed in feedback. These profiles can then be used to personalize recommendations, considering users' tastes and preferences. HRS can enhance its recommendation accuracy by integrating deep learning and NLP techniques into text analysis for customer feedback [12]. The ability to extract fine-grained sentiment, identify important topics, and understand the context allows for more precise recommendation generation.

Consequently, customers receive recommendations that align closely with their preferences, increasing their satisfaction and engagement with the system. Finally, deep learning and NLP techniques for text analysis on customer feedback within a Hybrid HRS offer several significant advantages. These techniques enable semantic understanding, contextual analysis, automated feature extraction, user profiling, and improved recommendation accuracy. By leveraging the power of deep learning and NLP, HRS can unlock valuable insights from customer feedback data, leading to more effective and personalized recommendations that enhance the overall user experience. The novel contributions of this work are as follows:

- The proposed HRS-Net integrates the HAEC and CRNN approach, which offers several benefits for text analysis in customer feedback.
- The HAEC is introduced to cluster the user data and items separately based on multilevel user, user-user, and item-item interactions. The HAEC provides a robust clustering framework that captures explicit and implicit user preferences.
- The CRNN model learns distributed representations of words and phrases, capturing their semantic meanings and contextual information. It facilitates sentiment analysis, opinion mining, and topic identification in customer feedback, leading to a more comprehensive understanding of user preferences.
- The CRNN also enables extracting high-level features and creating user profiles based on sentiment, preferences, and discussed topics in the feedback. This personalized understanding of users enhances the accuracy and relevance of recommendations within the HRS.

The rest of the paper is organized as follows: Section 2 focuses on a detailed analysis of relevant research on recommendation systems. Section 3 entirely focused on the proposed HRS-Net with NLP-based HAEC analysis and CRNN framework. Then, the section 4 focused on simulation results with multilevel comparisons. Finally, section 5 concludes the article with possible future scope.

2. Literature Survey

Iftikhar et al. [13] comprehensively explore the E-commerce landscape, particularly within the Amazon product reviews classification. Their research not only navigates the intricacies of sentiment analysis but also harnesses the prowess of machine learning, deep learning methods, and BERT, a contextual language representation model. By amalgamating these cutting-edge techniques, the study strives to classify and categorize user-generated content, enabling a deeper understanding of customer sentiment and preferences. Amirifar et al. [14] illuminate the path toward a more accurate and informed E-commerce ecosystem. Their research introduces an innovative approach that intertwines NLP with the transformative potential of deep learning. By leveraging online reviews and product features, the study crafts a dynamic framework for predicting product ratings based on textual insights and intrinsic product attributes. This holistic approach transcends conventional methodologies, opening new vistas for predictive analysis within product recommendations and user feedback interpretation. The study by Karabila et al. [15] underscores the symbiotic relationship between sentiment analysis and collaborative filtering-based recommender systems. Recognizing the inherent value of user sentiments in refining product recommendations, the researchers infuse sentiment analysis into the recommendation framework. This innovative fusion augments the accuracy and personalization of recommendations, unveiling a pathway to more tailored and impactful user experiences.

Sangeetha and Kumaran [16] embark on a journey to unveil the multifaceted nature of user sentiments in the context of Amazon user reviews. By embracing a hybrid sentiment analysis approach, the researchers amalgamate distinct techniques to holistically capture and comprehend the intricacies of customer opinions. This comprehensive sentiment analysis aids in deciphering user preferences, which, in turn, informs decision-making processes across the E-commerce spectrum.

Deshai and Bhaskara Rao [17] champion the cause of transparency and authenticity in healthcare and E-commerce realms. Their research encompasses a multifaceted approach to detecting online fake reviews, harnessing the power of a dense neural network model with relevance mapping (DNN-RM). In an era marked by information proliferation, their study assumes a critical role in safeguarding consumer trust by facilitating the identification of genuine user-generated content. Ullah et al. [18] delve into the intricacies of user reviews within the context of low-rating software applications. Through an intricate exploration of rational information, the researchers unveil actionable insights that underscore the factors contributing to low ratings. This unique perspective not only aids developers in improving software applications but also offers a valuable vantage point for a nuanced understanding of user sentiments. Manikandan et al. [19] illuminate the path toward precision and personalization within E-commerce with their automatic product recommendation system. The researchers craft an innovative framework that refines and fine-tunes product recommendations by synergizing the Flamingo Search Optimizer and Fuzzy Temporal Multi Neural Classifier (FTMNC). This endeavour enhances customer satisfaction by delivering products that align more closely with individual preferences. Balaji and Haritha [20] unravel the complexity of sentiment analysis through their Ensemble Multi-Layered Sentiment Analysis (EMLSA) model. This sophisticated approach delves into the classification of complex datasets, acknowledging the nuanced interplay of emotions and opinions. The study captures the intricate tapestry of customer sentiments by elevating sentiment analysis to a multi-layered paradigm, providing businesses with a deeper understanding of user reactions. Duma et al. [21] embark on a pioneering quest to fortify the authenticity of online content through their DHMFRD–TER model. This deep hybrid model integrates review texts, emotions, and ratings, constituting a comprehensive framework for detecting fake reviews. In an age of digital misinformation, their innovative approach emerges as a beacon of reliability, bolstering consumer trust and fostering transparency.

Elangovan and Subedha [22] ally optimization and sentiment analysis, encapsulated in their Adaptive Particle Grey Wolf Optimizer framework (APGWO). By combining this optimization technique with deep learning-based sentiment analysis, the researchers unveil a novel avenue for comprehending user opinions and reactions. This adaptive approach encapsulates the dynamism of user sentiments, amplifying the precision of sentiment analysis. Qayyum et al. [23] contribute to the ongoing battle against fake reviews with their FRD-LSTM technique. By intertwining Deep Convolutional Weighted Representations (DCWR) with the Bi-LSTM method, the researchers forge a path toward more accurate and effective fake review detection. This endeavour aligns with the imperative to safeguard the veracity of online content, fostering a more informed and reliable E-commerce ecosystem. Gheewala et al. [24] elevate the realm of textual review-based recommender systems by infusing deep transformer models (DTM). By capitalizing on the power of these advanced architectures, the researchers enhance the depth and granularity of textual analysis. This integration can revolutionize recommendations, guiding users toward products that align more closely with their preferences and needs.

Choudhary et al. [25] epitomize the evolution of sentiment-based recommendation systems with their Deep Ensemble Learning (DEL) technique, known as SARWAS. By fusing a diverse array of sentiment-driven insights, this approach creates a comprehensive model for recommendation. The methodology's ensemble nature enhances accuracy and encapsulates the rich tapestry of user sentiments, enabling businesses to tailor their offerings more precisely.

3. Proposed methodology

Analyzing customer feedback is crucial for businesses to improve their products, services, and customer satisfaction. By understanding the opinions and preferences expressed by customers, companies can identify areas of improvement, address customer concerns, and tailor their offerings to meet customer needs. Effective customer feedback analysis enables businesses to generate personalized recommendations, enhancing customer experience and fostering long-term loyalty. Therefore, customer feedback analysis is pivotal for businesses seeking continuous improvement and customer-centric strategies. Figure 1 and Table 1 depict the proposed HRS-Net, which utilizes deep learning and NLP techniques to analyze customer feedback and provide accurate suggestions. The process begins with NLP-based HAEC, which clusters user data and items separately, considering multilevel user-item interactions, user-user interactions, and item-item interactions. This clustering approach captures explicit and implicit user preferences, providing a robust framework for understanding customer feedback. The HAEC clusters the user data and items based on their interactions, enabling a comprehensive analysis of user preferences. By considering the relationships between users and items at multiple levels, the HAEC can capture complex patterns and preferences within the data.

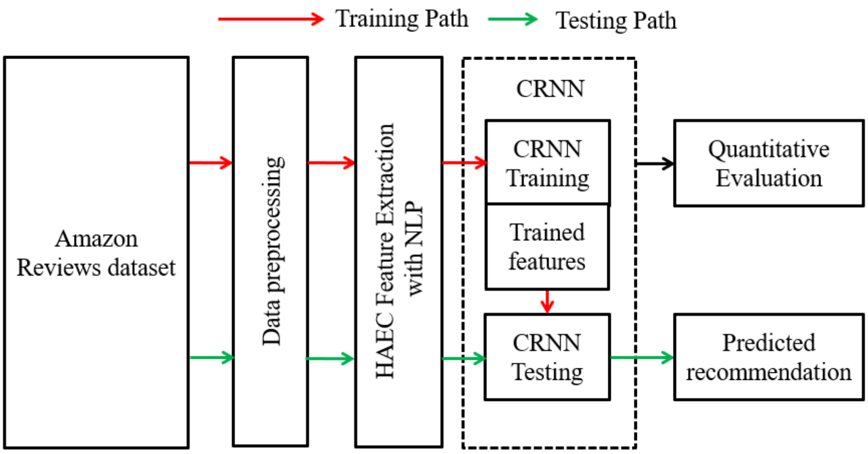

Fig 1. Proposed HRS-Net block diagram.

Table 1. Proposed HRS-Net algorithm.

Input: Original dataset **Output:** Recommendations to users.
Step 1: Perform the preprocessing operation to normalize the dataset based on reviews and ratings. **Step 2:** Perform NLP-based HAEC on the user data and items separately. • Consider multilevel user-item interactions, user-user interactions, and item-item interactions. • Cluster the user data and items based on their interactions to capture explicit and implicit user preferences. • Use a robust clustering framework to understand customer feedback comprehensively. **Step 3:** Train a CRNN using the features extracted by HAEC. • Extract textual features from the user feedback to understand semantic meanings and contextual information. • Enable sentiment analysis, opinion mining, and topic identification within the customer feedback. • Enhance the understanding of user preferences by creating user profiles based on sentiment, preferences, and discussed topics. **Step 4:** Leverage the high-level features extracted by the CRNN to generate tailored and precise suggestions for individual users. • Utilize the personalized understanding of users captured in the user profiles. • Enhance recommendation accuracy and relevance within the HRS.

Next, The CRNN is trained using the features extracted by the HAEC. It learns the textual features in the user feedback, allowing for a deeper understanding of the semantic meanings and contextual information. It facilitates sentiment analysis, opinion mining, and topic identification within the customer feedback, enabling a more comprehensive understanding of user preferences. The CRNN also plays a crucial role in creating user profiles based on sentiment, preferences, and discussed topics in the feedback. These user profiles capture the personalized understanding of users, leading to enhanced accuracy and relevance of recommendations within the HRS. By leveraging the high-level features extracted by the CRNN, the system can generate more tailored and precise suggestions for individual users. This superiority can be attributed to the combined power of HAEC and CRNN in capturing complex patterns, understanding semantic meanings, and considering contextual information in customer feedback. The approach improves recommendation accuracy, enhancing user satisfaction and engagement with the HRS.

3.1 Dataset preprocessing

We employ a comprehensive preprocessing strategy tailored to the Amazon Reviews dataset to ensure a seamless and optimised data processing workflow. Table 2 presents the proposed preprocessing algorithm, which orchestrates a series of systematic steps to transform the raw data into a structured and insightful format. Here, we outline the intricacies of this preprocessing approach, which serves as a vital precursor to robust data analysis and modelling. This process involves reading and extracting relevant information from the dataset, converting star ratings to a numerical scale, processing review dates, performing text preprocessing and sentiment analysis on the review text, and aggregating statistics at the product and user levels. It also includes creating interaction and feature matrices for users and products, normalizing data, and preparing it for model training and evaluation. This results in a structured dataset ready for in-depth analysis and modelling of user-product interactions and preferences.

Table 2. Proposed preprocessing algorithm.

Input: Raw Amazon Reviews dataset **Output:** Processed and structured dataset
Step 1: Read the Amazon reviews dataset. Step 2: Extract relevant attributes from the dataset, including review text, star ratings, product information, review date, and reviewer details. Step 3: Convert star ratings to a numerical scale, mapping them to values from 1 to 5. Step 4: Process review dates to extract meaningful time-related features, such as day of the week, month, and year. Step 5: Implement text preprocessing techniques on the review text, including tokenization, stop-word removal, and stemming or lemmatization. Step 6: Apply sentiment analysis to evaluate the sentiment polarity of each review, generating a numerical sentiment score. Step 7: Extract product identifiers (ASIN or ISBN) from the product information and create a mapping between products and their unique identifiers. Step 8: Aggregate review statistics for each product, including average rating, sentiment score, and review count. Step 9: Perform user-level analysis by grouping reviews based on reviewer information. Calculate user-specific statistics, such as average rating given by the user, review count, and sentiment trends. Step 10: Generate a user-product interaction matrix, capturing the interactions between users and products through reviews and ratings. Step 11: Create a user-feature matrix, incorporating user preferences and sentiment scores for different product attributes. Step 12: Construct a product-feature matrix reflecting each product's sentiment scores and attributes. Step 13: Normalize the matrices to ensure consistent scales and facilitate subsequent analysis. Step 14: Split the dataset into training and testing subsets for model evaluation and validation. Step 15: Store the preprocessed dataset in a structured format, ready for further analysis and modelling.

3.2 HAEC feature extraction

Elbow clustering, a prominent technique in the field of data clustering, serves as a pivotal tool for discerning the optimal number of clusters within a dataset. This method is characterized by its distinctive visual representation, resembling the shape of an elbow on a plot of the within-cluster sum of squares (WCSS) against the number of clusters. The fundamental objective of elbow clustering is to strike a balance between two competing factors: minimizing the variance within each cluster while avoiding excessive complexity through an excessive number of clusters. This equilibrium is achieved by identifying a "knee point" on the elbow curve, which signifies a point of diminishing returns. In simpler terms, it corresponds to the number of clusters at which adding more clusters does not significantly reduce the WCSS. The elbow curve is constructed by plotting the WCSS, the distortion, against a range of values for the number of clusters. The WCSS quantifies the sum of squared distances between data points and their assigned cluster centres. As the number of clusters increases, the WCSS tends to decrease, as smaller clusters lead to reduced internal variance. The mathematical analysis of WCSS is given as follows:

$$WCSS(K) = \sum_{i=1}^{K} \sum_{j=1}^{n_i} \left|\left|x_{ij} - \mu_i\right|\right|^2 \quad (1)$$

Here, K is the number of clusters used for testing, I represents the clustered index from 1 to K, n_i is the number of data points in cluster I, x_{ij} is the j^{th} data point in cluster I, μ_i is the centroid (mean) of cluster i. The WCSS measures the sum of the squared Euclidean distances between each data point x_{ij} in cluster I and its corresponding centroid μ_i. It measures how closely data points within each cluster are grouped around their centroid. A smaller WCSS indicates that data points are closer to their respective centroids, suggesting more compact clusters.

The HAEC technique emerges as a potent enabler within HRS-Net, poised to deliver heightened personalized recommendations to Amazon Reviews users. Through the adept utilization of clustering algorithms, HAEC embarks on a mission to partition Amazon Reviews into distinct clusters, each underscored by shared attributes. These clusters crystallize a refined spectrum of recommendations, harmonizing diverse methodologies to amplify user satisfaction. The HAEC methodology capitalizes on the multiple clustering techniques, deftly synthesizing their outcomes to cultivate a rich tapestry of suggestions. Applying HAEC to the Amazon Reviews can be stratified according to various dimensions. For instance, one cluster unites users, while another could amalgamate them based on their cumulative ratings, thus opening novel avenues for tailored recommendations. The resulting clusters, a testament to HAEC's intricate orchestration, unlock possibilities in catering to audiences with kindred preferences. By weaving together techniques like clustering and collaborative filtering, HAEC crafts a landscape where user preferences drive the segmentation of individuals into subsets of akin tastes. This agglomeration of preferences, carefully nurtured by clustering algorithms, forms the proposal foundation.

Tables 3 and 4 show the algorithmic steps of agglomerative and elbow clustering, while Figure 2 visualizes the cascading flowchart of HAEC's feature extraction. Table 5 captures the systematic progression of the HAEC algorithm. At its core, HAEC stands as a testament to NLP's potential, orchestrating users into cohesive subgroups bound by shared tastes. The algorithm, underpinned by user attributes such as age, gender, and viewing history, carves clusters that mirror the intricate weave of users' cinematic preferences. The clustering process manifests as an artful dance, evaluating the congruence between user attributes. Proximity to a cluster's centroid age reverberates as a decisive factor, a pivotal piece in crafting age-based recommendations. HAEC bequeaths a realm of meticulously tailored suggestions by evaluating degrees of similarity, selecting neighbours, and orchestrating predictions.

Prediction Computation: The resulting predictions are generated by comparing the closest neighbours identified inside the system database. The forecast was calculated using the following formula:

$$predicition_{u,i} = \frac{\sum_{n \in Neighbors}(r_{n,i} - \overline{r_n})\, sim_{u,n}}{\sum_{n \in Neighbors}|sim_{u,n}|} + \overline{r_u} \tag{2}$$

The elbow clustering is the nearest neighbour to the centroid to make a forecast. To show prediction ($sim_{u,n}$), HAEC utilizes both the elbow clustering and the correlation of the other variable on the right-hand side of the equation. In equation (2), $\overline{r_u}$ stands for the elbow cluster outcome, and $\overline{r_n}$ shows the correlation between the two other variables.

Table 3. Proposed Agglomerative Clustering Algorithm.

Input: Preprocessed data **Output:** Agglomerative Clustering Features
Step 1: Initialize each user and item as a separate cluster. **Step 2:** Calculate the similarity or dissimilarity between each pair of users and items based on a similarity metric, such as Jaccard similarity or cosine similarity. **Step 3:** Merge the most similar user-item pair iteratively until reaching the desired number of clusters or a stopping criterion. **Step 4:** Update the similarity or dissimilarity matrix after each merge by recalculating the similarities or dissimilarities between the newly formed clusters and the remaining users or items.

Table 4. Proposed Elbow Clustering Algorithm.

Input: Agglomerative Clustering Features **Output:** Elbow Cluster Features
Step 1: Choose the Range of Clusters: Define a range of potential cluster numbers to explore, typically starting from a small number and gradually increasing. **Step 2: Calculate WCSS:** For each value of k (number of clusters) within the defined range, perform clustering and compute the WCSS. **Step 3: Plot Elbow Curve:** Plot the values of k against the corresponding WCSS to create the elbow curve. **Step 4: Identify Elbow Point:** Examine the elbow curve to identify where the curve starts to bend or exhibit an "elbow-like" shape. This point represents a balance between minimizing WCSS and avoiding excessive complexity. **Step 5: Choose Optimal Cluster Number:** The optimal number of clusters is typically chosen at or near the identified elbow point. This value strikes a balance between meaningful cluster separation and practical interpretability.

Table 5. Proposed HAEC Algorithm.

Input: Preprocessed data **Output:** HAEC Features
Step 1: Initialize the algorithm parameters, such as the number of clusters (K), the maximum number of iterations, and convergence criteria. **Step 2:** Preprocess the user data and item data. Obtain user-item interaction data, capturing multilevel, user-user, and item-item interactions. **Step 3:** Represent the user and item data in appropriate feature representations, such as numerical vectors or text embeddings. **Step 4:** Perform agglomerative clustering on the user and item data separately, as mentioned in Table 3. **Step 5:** Assign users and items to the agglomerative clusters. For each user and item, assign them to the cluster with the highest similarity or lowest dissimilarity based on the agglomerative clustering results. **Step 6:** Perform Elbow clustering on the agglomerative clusters as mentioned in Table 4. **Step 7:** Output the resulting clusters. Provide the clusters of users and items to represent user preferences and item characteristics within the HRS.

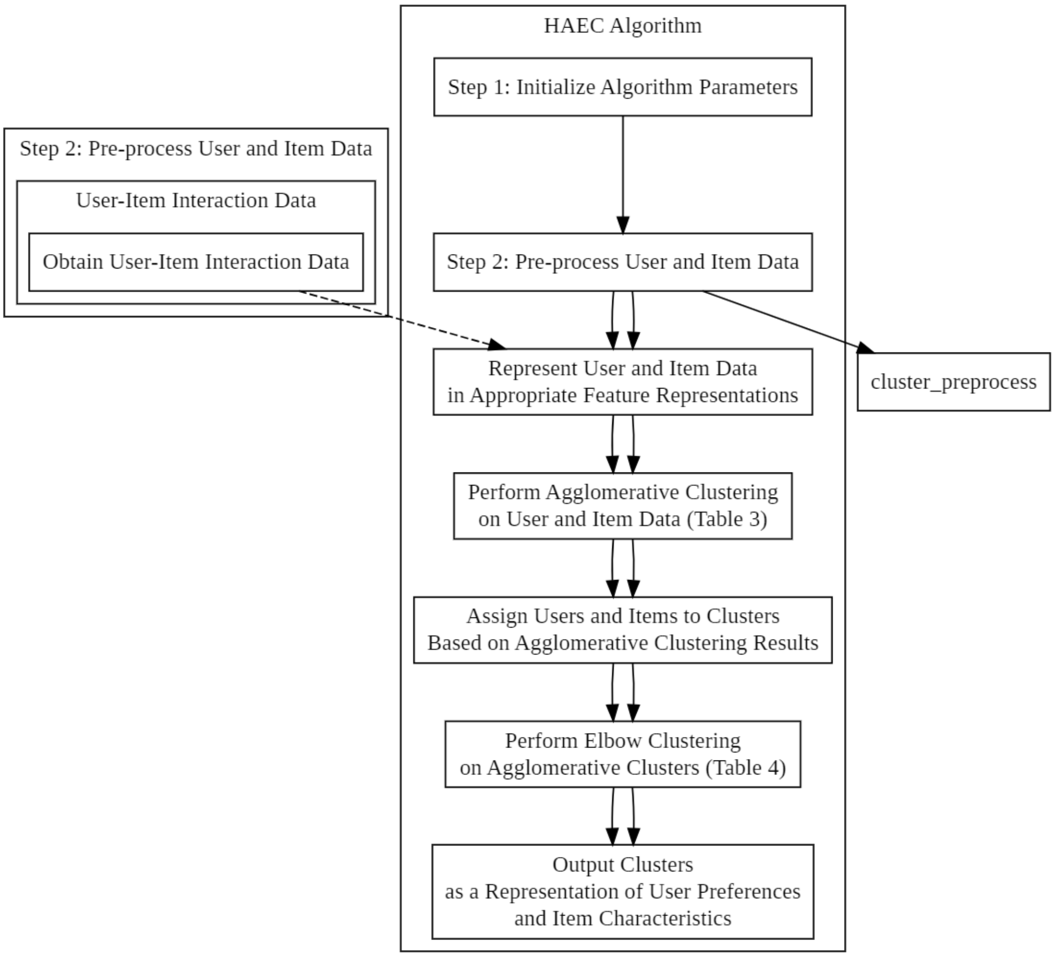

Fig 2. Proposed HAEC flowchart.

3.3 CRNN

A CRNN-based recommendation system provides consumers with suggestions tailored to their interests by assessing the user's profile, the product's attributes, and the users' ratings. Figure 3 shows the proposed CRNN architecture. The CRNN models frequently ignore user-specific temporal context information or automatically include these variables when providing suggestions. On the other hand, the success of a recommender system often hinges on how well it grasps and uses the context of the suggestion requests. CRNN calculates prediction ratings by considering the user's current time context and the item's dynamic attributes and qualities to provide relevant recommendations for a specific user. Similar people going through the same situations at the same time will naturally share their preferences. The effectiveness of a CNN-based time-aware recommendation system depends on selecting users who are most like the intended recipient and who also exist in the same temporal environment. So, the CRNN log the time-sensitive information, or temporal context, tied to the user's activities. Then, the user traits, object characteristics, and time information were sent into the CNN's input layer to recreate the original matrix. The convolution layer is then used to extract features from the matrix, and a method is supplied to compute the final output. The features selected in the convolution layer are defined as follows:

$$N_2 = \frac{(N_1+2P-F)}{stride+1}, \tag{3}$$

Here, N_2 is the output size, N_1 is the input data size, F is the convolution kernel size, the stride is the convolution kernel's sliding step, and P is the value used to fill in the input data so that it is divisible when the stride is larger than 1. The object is then compiled for storage after completion. Because the activation function in the activation layer is used to perform nonlinear operations, the neural network can simulate more complex models than if it were limited to simulating calculations between adjacent layers of the network, which it does by using linearity but only linear operation. Layer-to-layer interaction inside the neural network is strictly sequential. In CRNN, the activation functions are the most frequent. The conventional Tanh, sigmoid, and other activation functions suffer from a lack of gradient and a small usable interval range. Further, a nonlinear operation that uses resources well is used, revealing that the rectified linear unit (ReLU) functions are the primary tool for solving these two issues. The ReLU activation function is defined as follows

$$: f(x) = max(0, x), \tag{4}$$

The real number takes 0 when the gradient is less than 0, and the actual number takes when the gradient is greater than 0, solving the gradient disappearance issue. For computational efficiency, the pooling layer down samples and sparsely processes feature data. The average and maximum pooling techniques are two common examples of pooling strategies, where MaxPooling provides better feature selection performances. The features selected by MaxPooling are given as follows:

$$N_2 = \frac{(N_1 - F)}{stride+1}, \tag{5}$$

Then, the CRNN employ the fully connected layer with two dense methods to retrain the CRNN tail with lesser feature information loss.

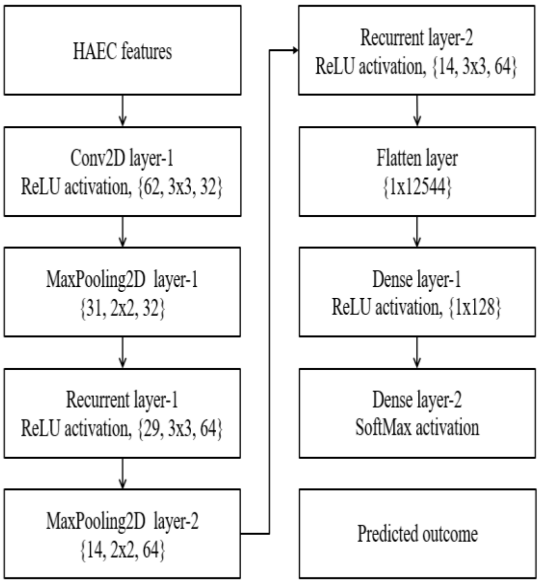

Fig 3. Proposed CRNN flowchart.

A recurrent layer is a type of layer commonly used in neural networks for processing sequential data. Unlike traditional feedforward layers, which process each input independently, recurrent layers have connections that allow them to maintain an internal memory of previous inputs. Ir makes recurrent layers particularly well-suited for tasks involving sequences, time-series data, or any data where the order of inputs matters. The fundamental building block of a recurrent layer is the recurrent neuron or unit, often referred to as an RNN cell. RNN cells process one input at a time while maintaining an internal state that captures information from previous inputs. This internal state is updated at each time step and is used to influence the processing of subsequent inputs. The user is presented with the obtained outcomes on the output layer using the SoftMax classifier. To utilize them as the index of the embedded matrix, the fields whose attributes are categories must first be changed to integers. The properties of each data table are next examined. In this case, the embedded layer is a part of the network's initial configuration. The embedded layer's output features are then utilized to transmit the features to the full-connection layer, and the full-connection layer's output features are delivered back into the embedded layer as input. After it is complete, the embedded layer's capabilities are used. The user and item features have finally advanced. A user feature vector and an item feature vector were trained to acquire prediction ratings for the items, and then the Top-k items with the highest ratings the target user has not rated were selected.

4. Results and discussion

This section gives a detailed analysis of simulation results, which are implemented using Python programming language. Further, the Amazon Reviews dataset is used to perform simulations, where the proposed HRS-Net and existing methods use the same dataset. In addition, the performance of HRS-Net is compared with other methods using multiple metrics.

4.1 Amazon Reviews dataset

An Amazon Reviews dataset typically refers to a collection of user-generated reviews and associated metadata extracted from the Amazon e-commerce platform. These datasets are often used for various purposes, such as sentiment analysis, recommendation system development, NLP research, and customer behaviour analysis. Amazon reviews datasets contain a wealth of information, including text reviews, star ratings, product identifiers, review dates, and sometimes additional attributes like helpfulness votes and reviewer demographics. Here is a general overview of what an Amazon Reviews dataset includes:

- Review Text: The actual text content of the reviews left by customers. Here, customers express their opinions, experiences, and thoughts about their purchased products.
- Star Ratings: A numerical rating assigned by customers to indicate their satisfaction level with the product. Ratings typically range from 1 to 5 stars, with 1 being the lowest and 5 being the highest.
- Product Information: Details about the product being reviewed, including product names, categories, and identifiers (such as ASIN or ISBN).
- Review Date: The date when the customer posted the review.
- Helpfulness Votes: Some datasets include information about how many other customers found the review helpful and how many total votes it received.

- Reviewer Information: Occasionally, datasets include anonymous identifiers for reviewers, which can be used for demographic analysis.
- Additional Metadata: There are other attributes, such as review titles and product features.

4.2 Performance comparison

Table 6 compares the performance of the proposed HRS-Net with existing AI-based recommendation system approaches. Here, the proposed HRS-Net resulted in superior performance as compared to BERT [13], EMLSA [20], and DHMFRD–TER [21] for all metrics. The graphical representation of these comparisons is presented in Figure 4. The proposed HRS-Net demonstrates improvements compared to BERT [13] with approximately 3.281% higher accuracy, 3.410% higher precision, 2.587% higher recall, and 2.874% higher F1-Score. The proposed HRS-Net shows improvements compared to EMLSA [20], with approximately 1.865% higher accuracy, 1.151% higher precision, 2.158% higher recall, and 1.631% higher F1-Score. The proposed HRS-Net exhibits improvements compared to DHMFRD–TER [21] with approximately 1.012% higher accuracy, 1.319% higher precision, 2.543% higher recall, and 1.888% higher F1-Score.

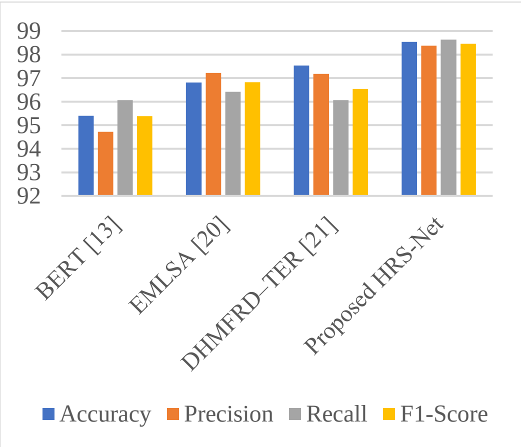

Figure 4. Graphical representation of Table 6.

Table 6. Performance comparison of proposed HRS-Net with AI approaches.

Method	Accuracy	Precision	Recall	F1-Score
BERT [13]	95.398	94.72	96.06	95.39
EMLSA [20]	96.812	97.22	96.42	96.82
DHMFRD –TER [21]	97.537	97.18	96.07	96.54
Proposed HRS-Net	98.54	98.38	98.63	98.45

Table 7 compares the performance of the proposed HRS-Net with existing clustering-based recommendation system approaches. Here, the proposed HRS-Net resulted in superior performance as compared to APGWO [22], DCWR [23], and DTM [24] for all metrics. The graphical representation of these comparisons is presented in Figure 5. Here, the proposed HRS-Net demonstrates a significant improvement compared to APGWO [22] with approximately 5.788% higher accuracy, 4.715% higher precision, 5.693% higher recall, and 5.783% higher F1-Score. The proposed HRS-Net shows a substantial improvement compared to DCWR [23], with approximately 9.915% higher accuracy, 9.111% higher precision, 9.431% higher recall, and 9.838% higher F1-Score. The proposed HRS-Net exhibits slight improvements compared to DTM [24], with approximately 0.663% higher accuracy, 2.489% higher precision, 1.977% higher recall, and 1.583% higher F1-Score.

Table 7. Performance comparison of proposed HRS-Net with clustering approaches.

Method	Accuracy	Precision	Recall	F1-Score
APGWO [22]	93.712	93.92	93.51	93.71
DCWR [23]	89.625	90.12	89.23	89.62
DTM [24]	97.879	96.07	96.69	96.88
Proposed HRS-Net	98.54	98.38	98.63	98.45

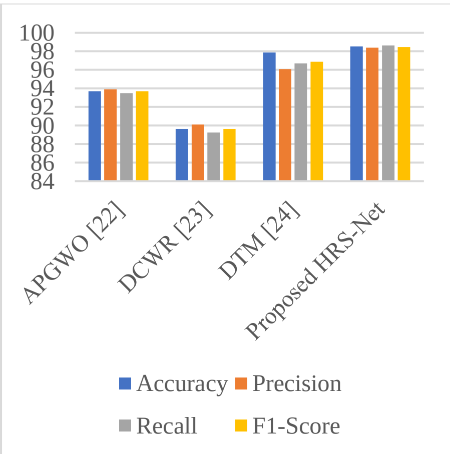

Figure 5. Graphical representation of Table 7.

Table 8 compares the performance of the proposed HRS-Net with existing filtering-based recommendation system approaches. Here, the proposed HRS-Net resulted in superior performance as compared to DEL [25], FTMNC

[19], and DNN-RM [17] for all metrics. The graphical representation of these comparisons is presented in Figure 5. The proposed HRS-Net demonstrates improvements compared to DEL [25] with approximately 4.019% higher accuracy, 4.043% higher precision, 4.104% higher recall, and 4.014% higher F1-Score. The proposed HRS-Net shows improvements compared to FTMNC [19], with approximately 2.518% higher accuracy, 1.992% higher precision, 2.869% higher recall, and 2.524% higher F1-Score. The proposed HRS-Net exhibits improvements compared to DNN-RM [17] with approximately 1.667% higher accuracy, 0.420% higher precision, 0.853% higher recall, and 0.573% higher F1-Score.

Table 8. Performance comparison of proposed HRS-Net with filtering approaches.

Method	Accuracy	Precision	Recall	F1-Score
DEL [25]	94.217	94.38	94.05	94.21
FTMNC [19]	96.129	96.42	95.81	96.12
DNN-RM [17]	96.875	97.97	97.78	97.88
Proposed HRS-Net	98.54	98.38	98.63	98.45

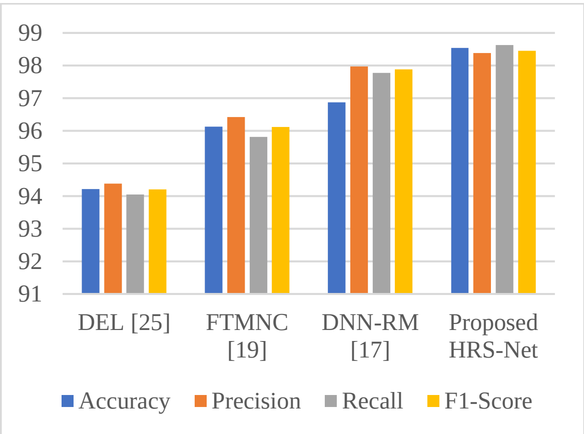

Figure 6. Graphical representation of Table 8.

4.3 Ablation study

Table 9 shows the ablation study of the proposed HRS-Net with various combinations. Here, the proposed HRS-Net resulted in superior performance compared to HRS-Net with agglomerative clustering, HRS-Net with Elbow clustering, and other combinations. Proposed HRS-Net without preprocessing shows slight decreases compared to the Proposed HRS-Net with approximately -0.634% lower accuracy, -0.562% lower precision, -0.620% lower recall, and -0.540% lower F1-Score. The proposed HRS-Net with agglomerative clustering shows slight improvements compared to the Proposed HRS-Net without preprocessing with approximately 0.331% higher accuracy, 0.490% higher precision, 0.150% higher recall, and 0.331% higher F1-Score. The proposed HRS-Net with elbow clustering shows slight improvements compared to the Proposed HRS-Net without preprocessing with approximately 0.226% higher accuracy, 0.370% higher precision, 0.060% higher recall, and 0.226% higher F1-Score. The graphical representation of these comparisons is presented in Figure 6.

Table 9. Ablation Study of proposed HRS-Net.

Method	Accuracy	Precision	Recall	F1-Score
HRS-Net without preprocessing	97.912	97.82	98.01	97.91
HRS-Net with agglomerative clustering	98.235	98.31	98.16	98.24
HRS-Net with elbow clustering	98.128	98.19	98.07	98.13
Proposed HRS-Net	98.54	98.38	98.63	98.45

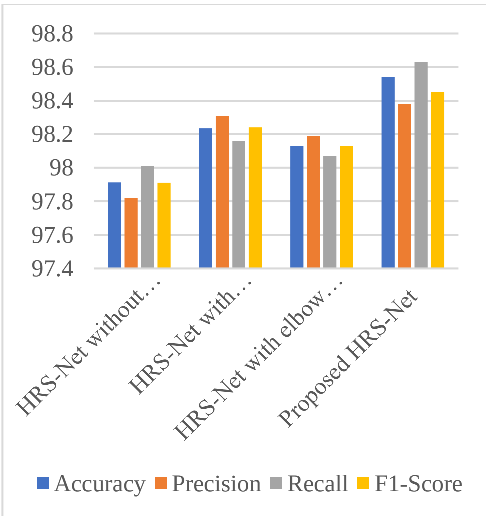

Figure 6. Graphical representation of Ablation Study.

5. Conclusion

The HRS-Net was leveraged, combining deep learning and NLP techniques to analyze customer feedback. The study introduced an NLP-based HAEC approach to effectively cluster user data and items separately, considering multilevel user-item interactions. This clustering technique captured intricate patterns and preferences among users and items. The CRNN focused on learning the textual HAEC features of user feedback, enabling a comprehensive understanding of customer preferences and sentiments. By incorporating the pre-trained HAEC interactions, the CRNN provided accurate suggestions to users, facilitating personalized recommendations. The simulation results validated the effectiveness of the proposed method, showcasing its superiority over conventional approaches. The proposed HRS-Net exhibits improvements compared to existing methods with approximately 1.012% higher accuracy, 1.319% higher precision, 2.543% higher recall, and 1.888% higher F1-Score. In terms of future scope, one potential direction to explore is the integration of RNN-BI-LSTM (RNN with Bidirectional Long Short-Term Memory) into the HRS-Net for customer feedback analysis. The RNN-BI-LSTM has shown promise in capturing sequential patterns and dependencies in text data, which can be valuable for analyzing customer feedback.

6.Declarations

CONFLICTS OF INTEREST STATEMENT

Manuscript Title:

Deep Learning-Based Hybrid Recommendation System with NLP- HAEC-Based Sentiment Analysis

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensingarrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Authors Name:

B. Madhurika[1]

D. Naga Malleswari[2]

COMPETING INTERESTS

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject or materials discussed in this manuscript

FUNDING

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript. No funding was received for conducting this study

Informed Consent

No individual participants are involved in this study

Author Contribution

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed By **B. Madhurika[1]and D. Naga Malleswari[2].** The first draft of the manuscript was written by **B. Madhurika** all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability

The datasets generated during or analysed during the current study are not publicly available due to privacy policies of authors, but are available from the corresponding author on reasonable request.

Research Involving Human and /or Animals

This research does not involve human participants or animals.

References

[1] Krosuri, Lakshmi Revathi, and Rama Satish Aravapalli. "Novel heuristic bidirectional-recurrent neural network framework for multiclass sentiment analysis classification using coot optimization." *Multimedia Tools and Applications* (2023): 1-21.

[2] Sharma, S. P., Singh, L., & Tiwari, R. (2023). Integrated feature engineering-based deep learning model for predicting customer's review helpfulness. *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1-18.

[3] Farhat, S., Abdelkader, M., Meddeb-Makhlouf, A., & Zarai, F. (2023). CADS-ML/DL: efficient cloud-based multi-attack detection system. *International Journal of Information Security*, 1-25.

[4] Bountakas, Panagiotis, et al. "Defense strategies for Adversarial Machine Learning: A survey." *Computer Science Review* 49 (2023): 100573.

[5] Hiriyannaiah, Srinidhi, Siddesh GM, and K. G. Srinivasa. "DeepLSGR: Neural collaborative filtering for recommendation systems in smart community." *Multimedia Tools and Applications* 82.6 (2023): 8709-8728.

[6] Shanthini, B., and N. Subalakshmi. "Advanced Data Mining Enabled Robust Sentiment Analysis on E-Commerce Product Reviews and Recommendation Model." *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. IEEE, 2023.

[7] Jabeur, Sami Ben, et al. "Artificial intelligence applications in fake review detection: Bibliometric

analysis and future avenues for research." *Journal of Business Research* 158 (2023): 113631.

[8] Daniel, D. A. J., & Meena, M. J. (2023). Deep learning-based hybrid sentiment analysis with feature selection using an optimization algorithm. *Multimedia Tools and Applications*, 1-24.

[9] Parmar, J., Chouhan, S., Raychoudhury, V., & Rathore, S. (2023). Open-world machine learning: applications, challenges, and opportunities. *ACM Computing Surveys*, *55*(10), 1-37.

[10] Chauhan, G. S., Nahta, R., Meena, Y. K., & Gopalani, D. (2023). Aspect-based sentiment analysis using deep learning approaches: A survey. *Computer Science Review*, *49*, 100576.

[11] Khan, W., Daud, A., Khan, K., Muhammad, S., & Haq, R. (2023). Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends. *Natural Language Processing Journal*, 100026.

[12] Chan, Kit Yan, et al. "Deep Neural Networks in the Cloud: Review, Applications, Challenges and Research Directions." *Neurocomputing* (2023): 126327.

[13] Iftikhar, Saman, et al. "Amazon products reviews classification based on machine learning, deep learning methods and BERT." *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 21.5 (2023): 1084-1101.

[14] Amirifar, Tolou, Salim Lahmiri, and Masoumeh Kazemi Zanjani. "An NLP-Deep Learning Approach for Product Rating Prediction Based on Online Reviews and Product Features." *IEEE Transactions on Computational Social Systems* (2023).

[15] Karabila, Ikram, et al. "Enhancing Collaborative Filtering-Based Recommender System Using Sentiment Analysis." *Future Internet* 15.7 (2023): 235.

[16] Sangeetha, J., and U. Kumaran. "Sentiment analysis of amazon user reviews using a hybrid approach." *Measurement: Sensors* 27 (2023): 100790.

[17] Deshai, N., and B. Bhaskara Rao. "Transparency in healthcare and e-commerce: detecting online fake reviews using a dense neural network model with relevance mapping." *Soft Computing* 27.14 (2023): 9861-9875.

[18] Ullah, Tahir, et al. "Exploring and mining rationale information for low-rating software applications." *Soft Computing* (2023): 1-26.

[19] Manikandan, B., P. Rama, and S. Chakaravarthi. "Original Research Article An automatic product recommendation system in e-commerce using Flamingo Search Optimizer and Fuzzy Temporal Multi Neural Classifier." *Journal of Autonomous Intelligence* 6.2 (2023).

[20] Balaji, Penubaka, and D. Haritha. "An Ensemble Multi-Layered Sentiment Analysis Model (EMLSA) for Classifying the Complex Datasets." *International Journal of Advanced Computer Science and Applications* 14.3 (2023).

[21] Duma, Ramadhani Ally, et al. "DHMFRD–TER: a deep hybrid model for fake review detection incorporating review texts, emotions, and ratings." *Multimedia Tools and Applications* (2023): 1-17.

[22] Elangovan, Durai, and Varatharaj Subedha. "Adaptive Particle Grey Wolf Optimizer with Deep Learning-based Sentiment Analysis on Online Product Reviews." *Engineering, Technology & Applied Science Research* 13.3 (2023): 10989-10993.

[23] Qayyum, Huma, et al. "FRD-LSTM: a novel technique for fake reviews detection using DCWR with the Bi-LSTM method." *Multimedia Tools and Applications* (2023): 1-15.

[24] Gheewala, Shivangi, et al. "Exploiting Deep Transformer Models in Textual Review Based Recommender Systems." *Expert Systems with Applications* (2023): 121120.

[25] Choudhary, Chaitali, Inder Singh, and Manoj Kumar. "SARWAS: Deep ensemble learning techniques for sentiment based recommendation system." *Expert Systems with Applications* 216 (2023): 119420.