

Indian journal of Engineering

To Cite:

Dinesh MG, Manisha G, Gunanandhini S, Anish TP, Thilagam T, Siva Subramanian R. Optimized Hierarchical Auto-Associative Polynomial Convolutional Transformer Network for Accurate Sentiment and Rating Prediction from Customer Reviews. *Indian Journal of Engineering*, 2026, 23, e16ije1724
doi:

Author Affiliation:

¹Dept of AI&DS, Easa College of Engineering and Technology, India, Email: dineshbabu.mg@gmail.com

²Dept of CSE, R.M.D. College of Engineering and Technology, India, Email: manisha.cse@rmd.ac.in

³Dept of CSE, S.A. Engineering College, India, Email: gunanandhini@saec.ac.in

⁴Dept of CSE, R.M.K. College of Engineering and Technology, India, Email: anishcse@rmkct.ac.in

⁵Dept of CSE, Vel Tech Rangarajan Dr. Sangunthala R&D Institute of Science and Technology, Chennai, India, Email: thilaka28@gmail.com

⁶Dept of CSE, R.M Institute of Science and Technology, India, Email: sivamr87@gmail.com

*Corresponding Author:

Thilagam T,
Dept of CSE, Vel Tech Rangarajan Dr.Sangunthala R&D Institute of Science and Technology, Chennai, India
Email: thilaka28@gmail.com

Peer-Review History

Received: 21 February 2026

Reviewed & Revised: 16/March/2026 to 18/September/2026

Accepted: 03 October 2026

Published: 12 October 2026

Peer-Review Model

External peer-review was done through double-blind method.

Indian Journal of Engineering
pISSN 2319-7757; eISSN 2319-7765



© The Author(s) 2026. Open Access. This article is licensed under a [Creative Commons Attribution License 4.0 \(CC BY 4.0\)](http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

Optimized Hierarchical Auto-Associative Polynomial Convolutional Transformer Network for Accurate Sentiment and Rating Prediction from Customer Reviews

Dinesh MG¹, Manisha G², Gunanandhini S³, Anish TP⁴, Thilagam T^{5*}, Siva Subramanian R⁶

ABSTRACT

The sentiment analysis and star rating prediction are essential to understand the consumer feedback from online reviews. Traditional approaches may not be sufficient to represent both nonlinear feature interactions and contextual relationships. To overcome these drawbacks, in this work, a Hierarchical Auto-Associative Polynomial Convolutional Network-based Geometric Algebra Transformer with Duck Swarm Optimization (HAPCNbGAT-DSO) is proposed. The framework is based on a hierarchical polynomial convolution for nonlinear feature learning, a Geometric Algebra Transformer (GATr) for relational contextual modelling, and Duck Swarm Optimization (DSO) for global hyperparameter search. After systematic pre-processing and feature representation, the model is tested on the Yelp Restaurant Review Dataset (YRRD) and Restaurant Review Dataset (RRD). The experimental results show the accuracy of 0.997 for RRD and 0.998 for YRRD, and the corresponding loss is 0.03 and 0.02 respectively. The study also assesses interpretability, cross domain generalization, computational efficiency, optimizer sensitivity, statistical significance and component-wise ablation.

Keywords: Duck swarm optimization, Geometric Algebra Transformer, Hierarchical Auto-Associative Polynomial Convolutional Network, natural language processing, Shapley Additive Explanations

1. INTRODUCTION

The internet is becoming an important resource for both consumers and service providers when it comes to restaurant reviews. These reviews are often used by customers when choosing a restaurant, assessing the quality of service, and setting their expectations for the dining experience. Therefore, the analysis of the

review text can be used to get evidence from the reviews that can be used to analyze customer satisfaction and areas for improvement (Iofrida et al., 2022; Chapman and Dilmperi, 2022). Perceptions of the population are presented in the review, which is very crucial in shaping organizational strategy and customer satisfaction (Ran et al., 2022; Gunawan et al., 2023). The analysis of textual reviews is therefore of utmost importance as a way of getting meaning out of the reviews. Sentiment analysis and opinion mining have been widely used to categorize customer comments and derive the associated ratings (Kumar et al., 2022). The majority of the current systems involve the use of NLP and machine-learning techniques based on lexical, syntactic or semantic representation (Nijhawan et al., 2022; Bondugula et al., 2022). While these methods work well, restaurant reviews often have nonlinear relationships, informal expressions, and context-dependent information, which can be challenging to capture using traditional feature-learning pipelines (Zhang et al., 2022; Briciu et al., 2024). In addition, recent findings in research show that both traditional deep learning approaches and those based solely on deep learning approaches do not contain characteristics such as interpretability and optimized algorithms required for practical implementation (Rejeb et al., 2023; Van Nguyen et al., 2022).

Transformer models offer good contextually relevant representations; however, they are not directly modeled explicitly in a conventional formulation. This inspires a unified architecture that integrates nonlinear feature transformation, relational representation, optimization, and post-hoc explanation (Manojkumar et al., 2023; Mendes, 2023). To overcome the above drawbacks, this work proposes a Hierarchical Auto-Associative Polynomial Convolutional Network-based Geometric Algebra Transformer with Duck Swarm Optimization (HAPCNbGAT-DSO). It is based on a polynomial network to expressively encode hierarchical relations, a Geometric Algebra Transformer (GATr) to capture relation dependencies, and Duck Swarm Optimization (DSO) for optimal parameter setting.

The main contributions of this study are as follows:

- We introduced a novel HAPCNbGAT framework that combines the hierarchical polynomial convolutional networks with a Geometric Algebra Transformer to model higher-order nonlinear feature interactions and the contextual relational dependencies in textual data. The integration is designed in a complementary way to feature learning, unlike the traditional hybrid approach.
- A good feature transformation approach is presented in which polynomial features are represented as multi vectors such that structural and semantic relations can be preserved for better sentiment analysis.
- Global hyperparameter optimization through Duck Swarm Optimization (DSO) is done since DSO is more stable than gradient optimization techniques, as gradient methods often fall into the trap of local optima.
- SHAP is used after model training to provide word-level and segment-level explanations of the predictions without changing the learned prediction function.
- Controlled baseline comparisons and cross domain evaluations are conducted and the optimizer comparison is performed to quantify the contribution of the proposed architecture, while statistical testing and ablation of individual components is conducted.

The organization of this Research Study is as follows: Section 2 gives an overview of some recent works on sentiment analysis and predicting star ratings from customer reviews. In Section 3, a description of the proposed approach is given. Section 4 gives results of the improved approach. Finally, the motivation behind the results of the research work is provided in Section 5.

2. LITERATURE REVIEW

Lin et al., (2023) explored the aspect-based sentiment analysis with elemental discourse units (EDUs). Their EDU-Capsule model is a model for aspect-level information within the context of sentences, which they tested on benchmark datasets and found to be useful for aspect and sentiment analysis. Based on the above finding, suggest an EDU-level Capsule network (EDU-CN) for ABSA. In EDU-Capsule, the learning of EDU representations is performed in their context in a sentence in order to infer attributes and emotion. When tested on two baseline datasets, EDU-Capsule beats robust baselines. The EDU-level annotations and EDU-Capsule source code are released to promote further study in this area.

Punetha and Jain (2023a) proposed a novel unsupervised sentiment classification model based on MCDM that can estimate customer satisfaction by detecting the sentiments like positive or negative sentiment in the reviews. The technique consists of two stages, namely the generation of performance measures from the integration of context, ratings, and emotion ratings, followed by the application of non-cooperative game theory for Nash Equilibrium. Customer satisfaction feedback and emotional sentiment polarity of the reviews are taken into consideration in the framework's outcome. Experimental results were achieved by applying the framework

to two restaurant review datasets. Since the framework is free of any language and domain limitations, it ensures consistent and rational results.

Durga and Godavarthi (2023) proposed a decision-based recurrent neural network for restaurant sentiment analysis. Their framework combines pretrained language representations, optimization, bag-of-words processing, and word embeddings to classify sentiment across review domains. Sentiments have been classified in terms of aspect and priority models using deep sentiment analysis (DSA). The model uses Python and other software such as Keras and Pandas to extract sentiments from Twitter, restaurant, and laptop datasets. Emotions predicted by the model are validated using the confusion matrix analysis method of its performance.

Anbumani and Selvaraj (2024) proposed a sigTan-Beta activation function for CNN-based sentiment classification. The study uses Word2Vec representations and modifies neuron activation to improve classification performance. In order to achieve high efficiency, the weights of neurons are tuned through the novel sigTan-Beta activation function. The proposed model has shown better results than other models regarding accuracy, precision, and F1-score.

Khan et al., (2023) developed a bidirectional modified-LSTM architecture with convolutional attention for aspect-category detection. The method models sequential patterns and contextual dependencies to improve the extraction of entity-attribute relationships. The model captures the contextual representation of the words to improve the process of aspect detection. This model was tested using SentiHood, SemEval-2015, and SemEval-2016 datasets, which produced F1-scores of 78.96%, 79.10%, and 79.03%, respectively.

Punetha and Jain (2023b) also presented mathematical optimization approaches for sentiment orientation and aspect-based ranking. This study presents two approaches called Sentiment Orientation Tagger Model (SOTM) and the Aspect-based ranking model (ABRM). SOTM uses simple additive weighting and the idea of Nash equilibrium from game theory to predict the sentiment of the review. The preference selection index (PSI) technique has been used by ABRM to rank different features. The validity of the model has been tested statistically, and the model has similar performance compared to other best models.

According to Mutinda et al., (2023), LeBERT is a framework for sentiment classification that integrates features from the lexicon-based sentiment classification technique along with those from CNN, BERT, and n-grams. LeBERT uses vectors to represent the input text selected via a combination of both context and lexicon features before CNN classification takes place. The performance of the approach was validated using different benchmark datasets such as restaurant reviews from Yelp, product reviews from Amazon, and movie reviews from IMDb. The model performed better using measures such as accuracy, precision, and F-measure, with an F-measure of 88.73%.

Obiedat et al., (2022) addressed imbalanced sentiment classification by combining support vector machines with particle-swarm optimization and alternative oversampling strategies, including SVM-SMOTE, ADASYN, Borderline-SMOTE, and SMOTE. SVM method will be used to predict the sentiment of the reviews on the famous Arabic review social network known as Jeeran. Four different oversampling algorithms, including SVM-SMOTE, ADASYN, Borderline-SMOTE, and SMOTE, have been used, while the PSO algorithm helps improve the weight of features. Performance-wise, the proposed PSO-SVM technique proves to outperform other classification algorithms.

Table 1. Comparison of the existing methods

Reference	Method	Advantage	Disadvantage
(Lin et al., 2023)	EDU-CN	Outperforms strong baselines, showcasing the effectiveness of Capsule networks for ABSA tasks.	Performance might vary on datasets with different structures or languages, limiting cross-domain applicability.
(Punetha and Jain, 2023a)	MCDM	Eliminates the need for labeled data, saving time and effort in annotation.	The computational overhead may become prohibitive for very large datasets or real-time applications.
(Durga and Godavarthi, 2023)	D-RNN	Optimized using Stochastic Gradient Descent (SGD) to improve task-specific performance.	Fine-tuned models may struggle with domain-specific nuances in unseen datasets.
(Anbumani and Selvaraj,	CNN	Demonstrates superior results in classifying product reviews into	Performance might vary across domains without additional fine-

2024)		positive or negative classes.	tuning.
(Khan et al., 2023)	Bi-MLSTM	Treats entity and attribute pairs as sequences or patterns, improving detection accuracy.	Risk of overfitting due to reliance on dataset-specific patterns during training.
(Punetha and Jain, 2023b)	MCDM-SOTM	Eliminates the need for labeled data, reducing manual annotation efforts.	Combining MCDM and game theory increases computational overhead.
(Mutinda et al., 2023)	LeBERT	Validated on multiple datasets showcasing its adaptability to different domains.	Resource-heavy architecture may face challenges with large-scale or real-time applications.
(Obiedat et al., 2022)	PSO-SVM	Utilizes PSO to optimize feature weights, enhancing classification performance.	Oversampling techniques, if not carefully applied, can lead to overfitting on the minority class.

The Table 1 shows a comparison of different sentiment analysis techniques for customer reviews. Each approach also has its advantages, such as better performance than other approaches, no need for labelled data and optimising the weights of the features. Yet, issues such as computational costs, domain-specific constraints, and potential problems like overfitting arise. In the context of advances in the field of sentiment analysis, there is a great deal of influence from transformer models and hybrid deep learning models. The latest study conducted by Aziz et al., (2024) has presented a new model using BERT and a multi-layer GCN. Moreover, there have been more developments in this area, such as UrduAspectNet as well as its modified version with biaffine attention. Graph attention models have exhibited an excellent ability to model dependencies and sentiments.

The latest research on low-resource languages has focused more on the creation of datasets and transfer learning methods. The Uzbek News Corpus (UZH) developed by Yusufu et al., (2025) is used as a baseline for NER, and Urdu related techniques by Aziz et al., (2025) focus on structure-based sentiment extraction, whereas domain adaptation techniques by Ahmed et al., (2025) focus on UACOS quadruple extraction. In addition, the generative models inspired by Zhou et al., (2025) have brought new methods into the field of implicit opinion mining.

The recent developments in transformers for sentiment analysis include advances made in explainable transformers (Perikos and Diamantopoulos, 2024), transformer-RNN hybrid models (Jahin et al., 2024), and language-specific transformers such as HaBERT (Musa et al., 2025). The surveys conducted by Tripathy and Sharaff (2026) and Wang (2026) are an indication of the increasing influence of transformers on aspect-based sentiment analysis. Furthermore, most previous studies focus only on single tasks such as sentiment classification or aspect analysis without any consideration for an approach that could achieve both rating prediction and sentiment classification. The optimization techniques adopted in previous models are mainly based on gradient descent methods that might result in poor convergence and weak generalization.

To overcome the above mentioned drawbacks a HAPCNbGAT-DSO framework has been developed, where HAPCN is used to extract features in a nonlinear way using hierarchical polynomials, GAT is used to perform relational analysis on multivectors and DSO is used to globally tune the model parameters.

Problem statement

Sentiment analysis and star rating prediction from restaurant reviews aim to automatically determine customer opinions and predict ratings based on textual data. As a result of issues like the need for large volumes of labeled data, domain biases, and challenges associated with generalization across diverse reviews, previous methodologies have struggled to capture the nuances of reviews. Furthermore, techniques like classical machine learning algorithms cannot suitably integrate contextual data, which has the effect of improper sentiment classification.

Deep models, though bearing a computational cost making them less ideal for real-time applications, are subject to being affected by overfitting due to oversampling methods used for class distribution correction. Such issues enhance the difficulty for current techniques to scale and adapt well for optimal sentiment and rating forecasting on various domains. The proposed model has remarkable accuracy (0.997/0.998), minimal loss (0.03/0.02), and optimal computation times of 1.5s (RR) and 2.7s (YRR).

3. METHODOLOGY

In the proposed study, a novel solution for predicting star ratings and sentiment analysis, Hierarchical Auto-Associative Polynomial Convolutional Network-based Geometric Algebra Transformer with Duck Swarm Optimization (HAPCNbGAT-DSO) is proposed. YRRD and RRD datasets were gathered and preprocessed to eliminate noise with the latest in NLP methods. Feature extraction involves the use of the Absolute Deviation Factor-Class Document Frequency (ADF-CDF) method to identify discriminative terms for sentiment classification. Shapley Additive Explanations (SHAP) is employed for segmentation to examine the word or phrase contribution toward sentiment scores and star ratings. Sentiment classification and star rating prediction employ the HAPCNbGAT, which integrates polynomial convolutional layers to learn nonlinear hierarchical features with a Geometric Algebra Transformer (GATr) that provides spatial and relational expressivity through the utilization of geometric primitives. Duck Swarm Optimization algorithm is used to optimize the parameters of HAPCNbGAT, such as error, weight, and learning rate, to best run it for improved sentiment classification and correct star rating prediction. Figure 1 describes the structural design of the proposed strategy.

Data collection

Customer reviews from two datasets, the Yelp Restaurant Review dataset (YRRD) and the Restaurant Review dataset (RRD), are given as input. Since these reviews were collected from the Kaggle website, they are full of noise. Pre-processing techniques are applied to remove this noise, and they are given in the following section:

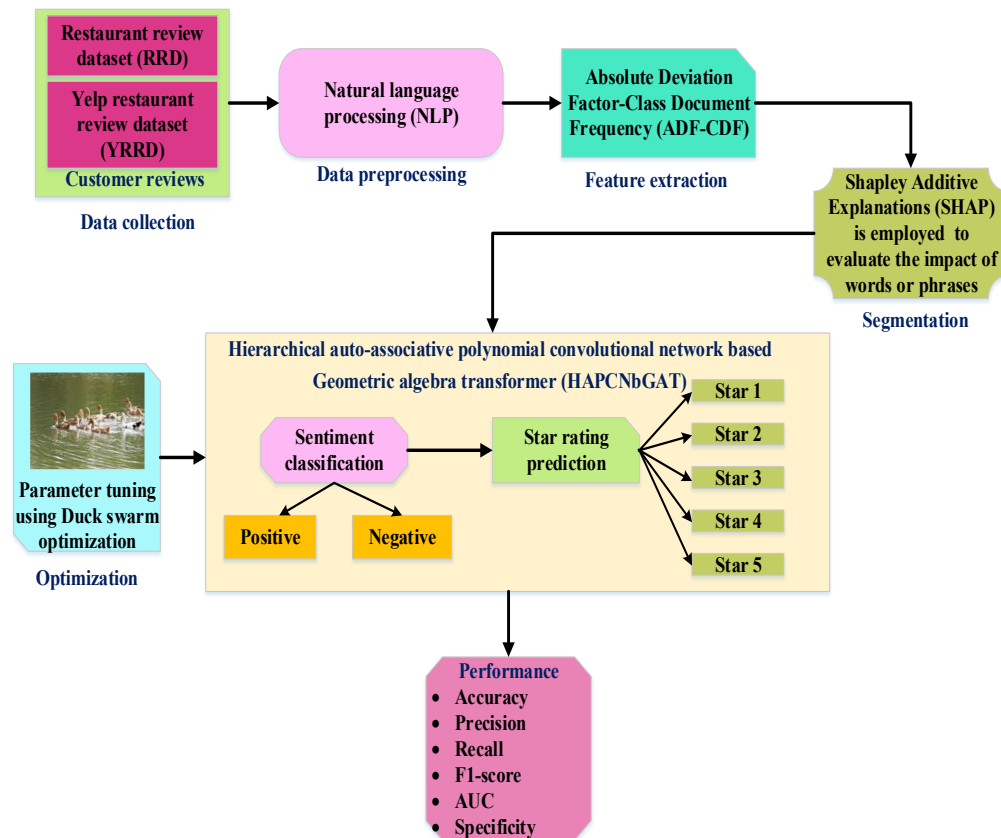


Figure 1. The architecture of the proposed approach

Data pre-processing

Data pre-processing is an important phase in NLP where raw reviews are converted into structured forms that can be used for sentiment analysis and rating predictions. In our research, data pre-processing is conducted on the YRRD and RRD corpora with specific tasks that maintain sentiments and reduce noise.

Text Cleaning

Irrelevant elements from raw text data include web addresses, markup languages, numbers, and other such artifacts. These are filtered out to have cleaner text as input. The whole text is then made lowercase, and contractions in the text are expanded out (e.g., “don’t” to “do not”).

Normalization of Slang and Abbreviations

User-generated reviews are likely to be casual and include slang, misspellings and abbreviations like as “gr8,” “luv,” “avg.” These are standardized into formal language such as “great,” “love,” “average” to avoid any confusion in the meaning and maintain consistency in feature learning.

Tokenization and Stop word Handling

Tokenization is done for the pre-processed document at the word level. Some stopwords are eliminated, but negations and other words that are necessary for determining sentiment, such as “not” and “never,” are preserved in order to avoid wrong sentiment determination.

Spell Correction and Semantic Preservation

Spelling variations and elongated words (e.g., “gooooood” → “good”) are normalized using rule-based correction. Emojis and emoticons are mapped to sentiment indicators (e.g., “😊” → positive, “😞” → negative) to retain emotional context. Negation handling is explicitly applied to capture polarity shifts in phrases.

Sequence Preparation

In addition, the tokenized sequences are then transformed into numeric forms and padded to a specific length, making sure that all inputs have a uniform size when fed into the neural network.

The above data preparation process is uniquely tailored to optimize the extraction of sentiment-dependent features.

Feature extraction using Absolute Deviation Factor- Class Document Frequency (ADF-CDF)

The extraction process using ADF-CDF method (Jin et al., 2023) is related to the evaluation of the discriminatory power of terms (features) based on the review texts in the Restaurant and Yelp Restaurant datasets. The idea is to use document frequency (DF) of the terms in different classes to extract relevant features for sentiment classification. The set of documents $D = \{d_1, d_2, \dots, d_M\}$ that make up the input training corpus,

where each document d_i is a review. The corpus contains the terms $T = \{t_1, t_2, \dots, t_n\}$, and the sentiment classes (such as positive and negative) are represented by $C = \{c_1, c_2, \dots, c_h\}$. For every term t_j , the document frequency (DF) $df(t_j, d_i)$ is calculated as a binary value that indicates whether t_j appears in document d_i . To determine the frequency of a phrase t_j in a particular class c_k , the CDF $cdf(t_j, c_k)$ is computed. It is provided by equation (1).

$$cdf(t_j, c_k) = \frac{\sum_{i=1}^M df(t_j, d_i) I(d_i, c_k)}{m_k} \quad (1).$$

Where m_k is the total number of documents in class c_k , and $I(d_i, c_k)$ is an indicator function that equals 1 if document d_i is a member of class c_k . The mean of the class document frequencies for all classes is the central tendency (CT) for each term t_j is given in equation (2).

$$CT_d(t_j) = \frac{1}{h} \sum_{k=1}^h cdf(t_j, c_k) \quad (2).$$

The document frequency's departure from this central tendency is measured by the ADF. It is provided by equation (3).

$$ADF(t_j) = |cdf(t_j, c_k) - CT_d(t_j)| \quad (3).$$

This metric measures the degree to which the document frequency of term t_j deviates from the class frequencies' central tendency. For every term t_j across all classes c_k , the ADF-CDF score is determined as follows in equation (4).

$$ADF_d(t_j, c_k) = |cdf(t_j, c_k) - CT_d(t_j)| \quad (4).$$

The final ADF-CDF score for each term $ADF_d(t_j)$ is computed as a weighted aggregation of class-wise deviations, defined as equation (5).

$$ADF_d(t_j) = \sum_{k=1}^h p(c_k) \times ADF_d(t_j, c_k) \quad (5).$$

Where, $p(c_k)$ indicates the probability of class c_k , and $ADF_d(t_j, c_k)$ indicates the ADF score for the term t_j in class c_k . It is believed that words with higher $ADF_d(t_j)$ values have more discriminative potential. These terms have been chosen as the most crucial characteristics for sentiment classification. ADF-CDF allows for improved feature selection and improves the performance of the sentiment classification model by efficiently measuring the relevance of terms by examining their deviations from the central tendency across classes.

Therefore, the feature extraction process with the ADF-CDF can effectively extract the discriminative terms that can enhance the performance of sentiment analysis for restaurant reviews in the dataset RRD and YRRD. The greater the ADF-CDF score, the more important the term is for the classification of sentiment classes.

Segmentation using Shapley additive explanations (SHAP)

Discovering the effect of some words or expressions on sentiment scores and how they affect star ratings is the first part of the segmentation process for the review text segmentation task with SHAP (Joung and Kim, 2023) in sentiment analysis of the RRD and YRRD.

Split the review text into segments that help calculate the sentiment score or star rating, for example, sentiment or product characteristics. Assign a score to each word or expression based on how it affects the sentiment score or star rating with SHAP. The application of a pretrained sentiment classifier or a set of optimal machine classifiers on the review dataset is the beginning of the process. The prediction of sentiment or star rating is made by each classifier.

The Shapley value is calculated using the following formula for each word (or segment) pf_i (product feature or sentiment-related word) in the review is given in equation (6).

$$\varphi_{pf_i}(v) = \sum_{S \subseteq F: i \notin S} \frac{|S|! \cdot (|F| - |S| - 1)!}{|F|!} \cdot (v(S \cup pf_i) - v(S)) \quad (6).$$

Here, $\varphi_{pf_i}(v)$ indicates the Shapley values of the i^{th} product word, S indicates the subset of words excluding pf_i , F indicates the complete set of words in the review, $v(S \cup pf_i)$ indicates the contribution of the set S with pf_i included, and $v(S)$ indicates the contribution of the set S without pf_i .

The difference in predictions when a word is included versus excluded is used to calculate the contribution of each word or segment. A term with a higher absolute SHAP value has a greater impact on the sentiment score or star rating.

Absolute SHAP values of all product words in reviews im are calculated using Equation (7).

$$Im\ p_{imk'} = |SHAP_{imk'}| \quad (7).$$

Where, $Im\ p_{imk'}$ indicates the importance of the word for the $k' - th$ classifier. To get a weighted total, add up the significance ratings for each of the K' classifiers are given in equation (8).

$$Im\ p_{im} = \sum_{k'=1}^{K'} w_{k'} Im\ p_{imk'} \quad (8).$$

Where, $w_{k'}$ indicates the weight assigned to the $k' - th$ classifier based on its performance, as given in equation (9).

$$w_{k'} = \frac{w_{k'}}{\sum_{k'=1}^{K'} w_{k'}} \quad (9).$$

Each review is segmented based on SHAP values for meaningful words or phrases. Meaningful words or phrases in a review are those having high SHAP values.

The use of SHAP-based segmentation makes it possible to find meaningful words or phrases in a review text that can help boost sentiment analysis. The application of SHAP to words in a review and averaging their SHAP values for multiple classifiers ensures meaningful segmentation consistent with the sentiment and star rating predictions.

Sentiment classification and star rating prediction using hierarchical auto-associative polynomial convolutional network based Geometric algebra transformer (HAPCNbGAT)

In case of predicting the sentiments and star rating score predictions, a hierarchical auto-associative polynomial convolutional network (Martell and Asari, 2018) combined with a geometric algebra transformer (HAPCNbGAT), (Brehmer et al., 2023). The inclusion of HAPCNbGAT: Polynomial Convolutional Layers for better representation, which make use of polynomial expansions in capturing nonlinear hierarchical data; The Geometric Algebra Transformer (GATr) provides the advantage of ensuring equivariance and expressivity through extracting spatial relational data by using geometric primitives like geometric product, join, and dual. The pipeline consists of:

- Polynomial convolutional layers for input processing.
- Geometric attention for feature aggregation.
- Using fully connected layers, sentiment categorization and star rating prediction are achieved.

The architecture of HAPCN is given in Figure 2.

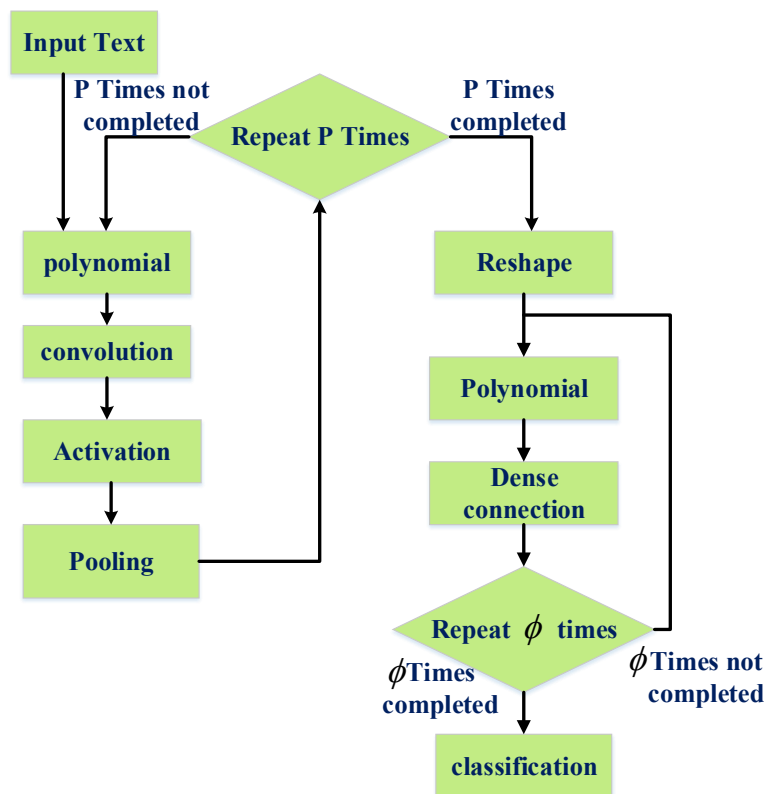


Figure 2. The architecture of HAPCN

Sentiment classification process

Following word embedding and initial encoding, the polynomial convolutional layers manage the hierarchical representation of the review text, which is represented as feature maps. Every polynomial layer is applicable is given in equation (10).

$$f_{p_1}(x, y, z, n) = f_{p_0}(x, y, z)^n, \text{ for } n = 1, 2, \dots, N, \quad (10).$$

Here, $f_{p_0}(x, y, z)$ represents the feature map input from the previous layer or pooling layer, and N represents the order of the polynomial expansion. The convolution operation will increase the size of these feature maps by stacking them along the z axis.

The GATr that works with multivectors-valued tensors takes in the output of the polynomial layers and applies: Scalars, vectors, and bivectors can be formed using the geometric product of vectors as shown in equation (11).

$$Geometric(x, y) = xy \quad (11).$$

Where, x, y indicates the input multivectors. Dual information is encoded via the join operation to improve expressivity, as given in equation (12).

$$EquiJoin(x, y; z) = z_{0123}(x^* \wedge y^*) \quad (12).$$

Where, z_{0123} indicates the pseudoscalar component of a reference multivector z . Values (v), keys (k), and multivector-valued queries (q) are handled using equation (13).

$$Attention(q, k, v)_{i, c'} = \sum_i Soft\ max_i \left(\frac{\langle \sum_c q_{ic} k_{ic} \rangle}{\sqrt{8n_c}} \right) v_{ic'} \quad (13).$$

Where, i, i' indicates token indices, c, c' indicates the channel indices, and $\langle ., . \rangle$ indicates the invariant inner product. It ensures that the network retains its geometrical connections while focusing on key features of the review text. The polarity of sentiments is predicted using the fully connected layers through the following equation (14).

$$g_{\varphi_1}(k, n) = g_{\varphi_0}(k)^n, \text{ for } n = 1, 2, \dots, N \quad (14).$$

The Softmax function is applied to determine the sentiment score.

Star Rating Prediction Process

A five-class star rating scale (1 to 5 stars) is mapped to the output feature space in the star rating prediction, which expands the sentiment categorization. The contribution of the polynomial layer to the spread of errors is given in equation (15).

$$e_{\varphi}^{pre-poly}(k) = \sum_{n=1}^N e_{\varphi}^{post-poly}(k, n) \quad (15).$$

Where, $e_{\varphi}^{post-poly}(k, n)$ indicates the post-polynomial error, and $e_{\varphi}^{pre-poly}(k)$ indicates the error propagated back to the preceding layer. The error is spread even further by equation (16).

$$e_{\varphi}(k, n) = \sum_{l=1}^L e_{\varphi}(l) w_{\varphi}(l, k, n) \quad (16).$$

Where, $w_{\varphi}(l, k, n)$ indicates the weights connecting layers. The gradient of the polynomial features is taken into account in the weight updates is given in equation (17).

$$\Delta w_{\varphi}(l, k, n) = \eta e_{\varphi}(l) g_{\varphi_2}(l) g_{\varphi_1}(k, n) \quad (17).$$

Where, η indicates the learning rate. An argmax function over the five different categories will be used along with the aggregation of the output from the fully connected layers to predict the star rating.

All the variables and operations of the proposed HAPCNbGAT framework are clearly defined for clarity and reproducibility. Let $x \in R^{n \times d}$ denote the input embedding matrix, where n represents the number of tokens and d denotes the embedding dimension. The

hierarchical polynomial convolutional layer produces feature representations $H^{(l)}$, where l denotes the layer index and k represents the polynomial degree.

The polynomial convolution is designed to model the nonlinear interactions of order higher than two and is expressed as a weighted sum of transformed features at different polynomial degrees, which can be represented by equation (18).

$$M^{(l)} = g(H^{(l)}) \quad (18)$$

where $g(\cdot)$ denotes the geometric algebra mapping function that transforms vector-valued features into multivector representations. Such an approach provides a means for the inclusion of relational dependencies and structure dependencies.

The connection between the polynomial convolution operator and GATr can be established through the application of $M^{(l)}$ to the geometric attention model, whereby the computation of multivector queries, keys, and values is employed to learn context-dependent relations.

By following these procedures, this architecture achieves best performance on sentiment classification and star rating prediction tasks for restaurant review datasets.

Theoretical Justification of GATr for Linguistic Representation

The usage of GATr in the proposed architecture is conceptually justified by the fact that the GATr enables modeling higher-order interactions based on multi vector representations that go beyond the conventional vectorial embedding space. In natural language processing tasks, each word is first modeled as a vector in a continuous vector embedding space. These vectors are basically first-order geometric entities.

By utilizing the GATr in NLP applications, the input vectors are converted into multivectors and hence enable the capturing of interactions among different linguistic items. For example, in the case of word pair interactions, the GATr converts the vectors into bivectors based on the geometric product of two vectors. Thus, word interaction and relational properties like word co-occurrences are captured.

The mapping enables the model to take into account semantic relations generated by combinations of words rather than standalone instances. For instance, sentiments expressed in phrases are modeled using multivectors in such a way that they enable more accurate representations of polarities. The operations of joining and duality can be employed to incorporate both local and global contexts in order to preserve consistency in relation learning.

Thus, the implementation of the GATr method makes it possible to achieve an effective approach to linguistic entity modeling based on mathematical reasoning.

Tuning the parameters of HAPCNbGAT using Duck Swarm Algorithm (DSO)

The Duck Swarm Optimization (DSO) algorithm is used within the suggested framework for optimizing the hyperparameters of the HAPCNbGAT model. The learning rate, filter size of the convolutional operation, degree of polynomial expansion, number of attention heads of transformers, and weights are the parameters optimized by DSO. Each solution corresponds to an instantiation of the above-mentioned parameters, while the fitness objective is formulated using the combined fitness function. The step-by-step process of DSO is shown in Figure 3.

The objective of the optimization process here is to enhance the overall performance of the HAPCNbGAT network on classification and star rating prediction through the minimization of the loss function $f(X)$, with $X = [E, W, \eta]$ being the set of parameters comprising error (E), weights (W), and learning rate (η). Parameters such as the weights W help learn good feature representation in the convolutional and transformer layers, the learning rate η dictates the pace at which weight updates are done, and error E represents the gap or loss $f(X)$ in terms of classification and prediction.

Step 1: Initialization (Duck Swarm Position Initialization)

One of the potential solutions (X_i one of the sets of parameters) is indicated by every duck in a D-dimensional search space. The first swarm can be initialized randomly using Equation (19).

$$X_i = L_b + (U_b - L_b) \cdot \omega \quad (19).$$

Where, X_i indicates the initial position of the i^{th} duck in the parameter space, ω indicates the random matrix with values in (0, 1), N indicates the number of ducks in the swarm, and L_b, U_b indicates the lower and upper bounds for the parameters.

Step 2: Fitness evaluation

Depending on the fitness of the HAPCNbGAT, the fitness function $f(X_i)$ will be determined for each solution (X_i). The fitness is evaluated based on:

- Performing sentiment classification through cross-entropy loss.
- The use of MSE, or mean squared error, to predict star ratings.

Step 3: Exploration phase (Searching for food sources)

Use of MSE or mean squared error to predict star ratings. The following is the most recent position update for the i^{th} duck is, given in equation (20).

$$X_i^{t+1} = \begin{cases} X_i^t + \mu X_i^t \cdot \text{sign}(r - 0.5) \text{ if } P > \text{rand} \\ X_i^t + CF_1 \cdot (X_{leader}^t - X_i^t) + CF_2 \cdot (X_j^t - X_i^t), \text{ if } P \leq \text{rand} \end{cases} \quad (20).$$

Where, X_{leader}^t indicates the best solution (leader duck) in the t^{th} iteration, X_j^t indicates the position of a neighboring duck, μ indicates the control parameter for global search, calculated as in equation (21) with $K = \sin(2 \cdot \text{rand}) + 1$.

$$\mu = K \cdot (1 - t/t_{max}) \quad (21).$$

And, CF_1, CF_2 indicates the cooperation and competition coefficients (0, 2). This stage guarantees a broader search range ($\mu > 1$) to find possible areas with superior solutions.

Step 4: Exploitation phase (Foraging in groups)

Ducks converge on the best solution when promising areas have been found. The i^{th} duck's position has been modified as follows in equation (22).

$$X_i^{t+1} = \begin{cases} X_i^t + \mu \cdot (X_{leader}^t - X_i^t) \text{ if } f(X_i^t) > f(X_i^{t+1}) \\ X_i^t + KF_1 \cdot (X_{leader}^t - X_i^t) + KF_2 \cdot (X_k^t - X_i^t), \text{ Otherwise} \end{cases} \quad (22).$$

Where, X_k^t, X_j^t indicates the positions of the neighboring ducks, KF_1, KF_2 indicates the cooperation and competition coefficients (0, 2), and $f(X)$ indicates the fitness function (e.g., accuracy and loss). This stage guarantees that ducks use nearby areas to hone their solutions.

Step 5: Termination criteria

The algorithm repeatedly goes through the phases of exploration and exploitation until one of the subsequent criteria is satisfied:

- The maximum iteration count (t_{max}) has been achieved.
- There has been no discernible improvement in $f(X)$ fitness value when the fitness value converges.

By applying DSO, the HAPCNbGAT achieves improved sentiment classification accuracy and more precise star rating predictions.

In order to maintain coherence and logical sequence, the suggested model HAPCNbGAT-DSO is developed based on the sequential approach. In the first step, the raw reviews are preprocessed to eliminate any noise for improved feature quality. After that, discriminative features are obtained using the ADF-CDF technique by measuring the deviation in terms of each class. Next, the SHAP method is utilized to assign relevance scores to the words and phrases.

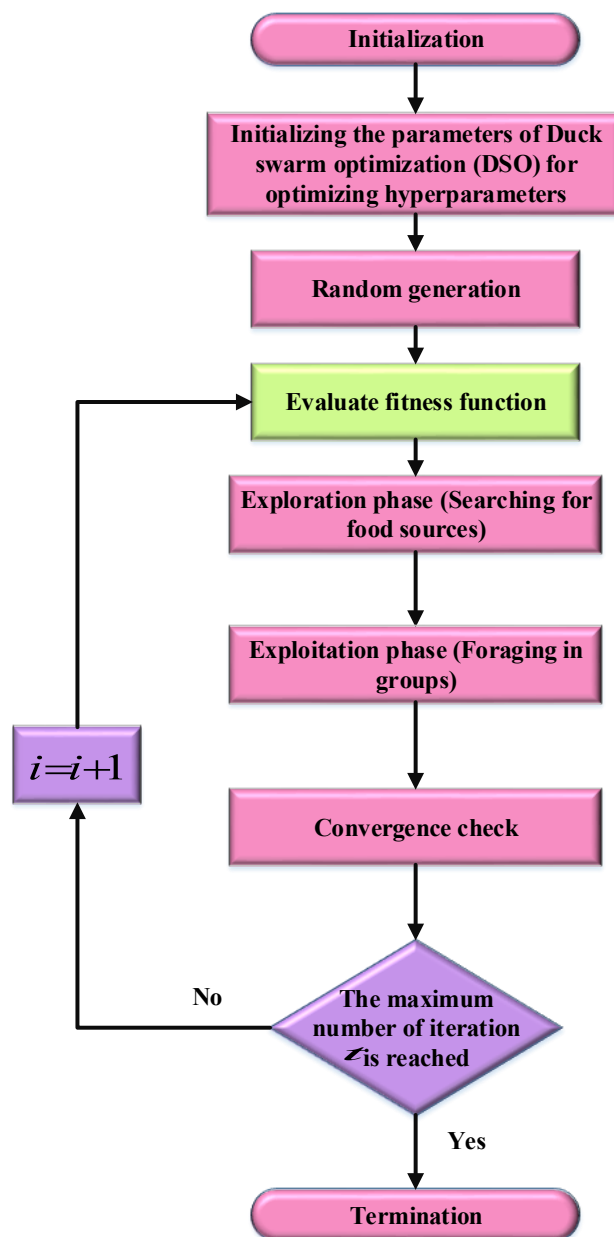


Figure 3. The flowchart of DSO

These preprocessed features are subsequently used for training in the HAPCNbGAT architecture, whereby hierarchical polynomial convolution is leveraged for capturing nonlinear higher-order interactions between different features, which cannot be appropriately captured using traditional techniques. In addition to this, the Geometric Algebra Transformer is used for modeling the context and relation-based dependencies between features using multi vector operations.

Finally, the optimization of important parameters such as learning rate, filters for convolution, polynomial degrees, and number of attention heads is done using Duck Swarm Optimization. Utilizing global search helps avoid the problem of convergence and reaching local minima that can arise from other gradient-based techniques. Hence, each step of the process serves an important function.

4. RESULTS & DISCUSSION

The results and analysis of the proposed HAPCNbGAT-DSO technique are shown in this section. The proposed approach is tested on the Python 3.7.14 programming language environment and tested on the Windows 10 platform. Table 2 summarizes the

implementation environment and configuration settings used for conducting the experiments, ensuring transparency and reproducibility of the proposed approach.

Table 2. Experimental Environment and Configuration Settings

Parameter	Description
Programming language	Python
Version	3.7.14
Operating system	Windows 10
Dataset	RRD, YRRD
Data type	Text
Optimization	Duck swarm optimization

Dataset description

The experiments are conducted on two benchmark datasets: Restaurant Review Dataset (RRD) (<https://www.kaggle.com/datasets/joebeachcapital/restaurant-reviews>) and Yelp Restaurant Review Dataset (YRRD) (<https://www.kaggle.com/datasets/farukalam/yelp-restaurant-reviews>). Both datasets are applied in the implementation of two distinct supervised learning tasks: (i) Sentiment classification, where texts of the review are assigned positive or negative labels, and (ii) star rating prediction (from 1 to 5), which is treated as a regression/classification output.

The initial RRD dataset consists of 10,000 reviews, out of which 9,842 remain after removing duplicates, null values, and noisy text. The initial YRRD dataset consists of 20,491 reviews, out of which 20,103 remain after removing duplicates, null values, and noisy text.

In relation to the sentiment classification, the RRD dataset comprises 5,912 positive and 3,930 negative samples, while the YRRD comprises 12,064 positive and 8,039 negative samples. Concerning the star ratings prediction task, both datasets contain 5-star class distributions (1 to 5) where higher ratings (4 to 5) correlate more often with positive sentiment and vice versa.

Input features include only review texts, whereas sentiment labels and rating values will be considered as separate target variables for their respective tasks.

In this case, there are two principal tasks that will be solved: sentiment prediction and star rating prediction – both formulated as supervised learning problems.

Sentiment prediction can be considered as a binary classification problem with the goal of estimating whether the given text expresses negative or positive sentiments. As for input data, the model will receive only preprocessed textual reviews, which are represented as features using embeddings and hierarchical feature learning. Discriminative patterns correlated with sentiment-bearing expressions, context, and polarity will be captured to classify review texts as positive or negative.

Rating prediction can be treated as a multi-class classification problem to predict the actual numeric rating (from 1 to 5) based on the textual description provided in the review. Since rating predictions need to take into account the fine granularity of differences between ratings, such features have to be learned from the textual review input data, in order to map different sentiment intensities and context emphasis.

Both systems perform operations on the same dataset, but the models learn independently so that there is no implicit data leakage between them. This allows for proper learning and hence allows the model to perform unbiased sentiment analysis and rating prediction.

Data Leakage Prevention

In order to guarantee the validity of the obtained results and to prevent any potential data leakage, specific precautions are taken at every step of experiment construction. Specifically, input data consist solely of review texts, whereas labels correspond to their respective ratings. At no point does metadata become a part of the training and testing procedure, such as timestamps, user IDs, or restaurant IDs.

Besides, sentiment classification and rating prediction represent two separate supervised learning problems, with labels being assigned on a totally independent basis from text content and never from rating value. This excludes any data leakage caused by correlation between label and text content.

Data splitting takes place before any pre-processing steps are implemented to make sure that test data do not get into the training procedure and do not influence its outcomes. Finally, cross-validation is run exclusively on training data. These steps make sure that the high performance claimed is independent of any information leak or unintended data leakage and is really dependent on the ability of the proposed model to generalize.

Performance Metrics

In order to have a thorough evaluation of the performance measures, all the performance metrics will be calculated within the same experimental setup. These metrics include accuracy, precision, recall, F1, and AUC scores, which can be calculated from confusion matrix scores in a typical way. For errors in rating predictions, metrics including MSE, MAE, and RMSE are calculated.

For better comprehension of the evaluation metrics, the definition and importance of each one is outlined separately. Accuracy shows the ratio of all true predictions to the total number of predictions made, and it gives an estimate of the overall model performance. Precision shows the share of true positive predictions in all positive predictions, which gives an indication of avoiding false positive cases. Recall shows how many positive examples are captured by the model out of the entire set of positive examples.

F1 score represents the harmonic average of both precision and recall, which makes it more relevant for the evaluation of models where there is a lack of balance between classes. The area under the curve gives an estimate of the model's ability to differentiate between classes irrespective of the thresholds used.

As for the rating prediction, Mean Squared Error (MSE) indicates the mean value of squared differences between the predicted and real ratings, thus paying more attention to higher discrepancies. In turn, Mean Absolute Error (MAE) is used for measuring the mean of absolute differences and allows obtaining a more understandable indicator of error. Finally, Root Mean Squared Error (RMSE) is computed as the square root of MSE and shows the error in units of the target variable.

In turn, computational time indicates the model's performance in terms of the time needed to learn the data, measured under the same experimental conditions. That is, computational time is equal to the time spent on the training process in order to reach 50 epochs when using a batch size of 32. The training includes both forward and backward propagations but not the time consumed by the pre-processing stage. The experiment is performed using an Intel Core i7, 16 GB RAM, and support of the GPU. All obtained results are averaged over random seeds to increase stability.

Experimental Setup

In order to maintain transparency and reproducibility, the experimental environment is well specified. In particular, the data are separated into training, validation, and test sets following a 70:10:20 ratio through stratified sampling. Training is carried out using the training set, whereas the validation set is used for hyperparameter tuning, and the final performance is evaluated on the independent test set.

To enhance robustness and reduce variance, 5-fold cross-validation is performed on the training data, and the experiments are repeated using five different random seeds (i.e., 42, 52, 62, 72, and 82).

Text pre-processing operations include lowercasing, removing punctuation and stop words, tokenization, and padding sequences to a fixed length. The embedding and hierarchical feature-learning process of the proposed framework is then applied to the input review texts, to generate dense vector representations.

The training procedure is done using Adam optimizer. Adam optimizer is used for the training procedure. We have a batch size of 32 and 50 epochs for training. To avoid overfitting, patience = 5 epochs are used in early stopping. The detailed architecture and optimization configuration define the hyperparameter search space and the optimized configuration that is optimized by Duck Swarm Optimization (DSO).

To make a fair comparison, the same data partitioning, pre-processing methods, and metrics for evaluation are used for all baseline algorithms (CNN, LSTM, BERT, RoBERTa). The grid search over known ranges is used to tune hyperparameters of the baseline models.

The implementation code is implemented using Python 3.7.14 and the experiments are performed on a computer with an Intel Core i7 CPU and 16 GB of RAM. This configuration helps ensure that the experimental results are reliable and reproducible.

Detailed Architecture and DSO Configuration

The detailed model architecture and optimization configuration are described as follows to further improve the reproducibility of the proposed HAPCNbGAT-DSO framework. A trainable word embedding layer (300 dimensional) is used to convert the input review texts into dense vector representations. The embedding representations are randomly initialized and then fine-tuned throughout the training.

The HAPCN component consists of three convolutional layers of 64 filters each and kernel sizes of 3, 4 and 5 respectively. The polynomial degree is optimized in the range {2,3,4} with DSO and the final configuration is degree 3 with the validation performance. The hidden layers use the ReLU activation function and a dropout rate of 0.5 is applied to prevent overfitting. Xavier uniform initialization is used to initialize model parameters.

The GATr component is made of two layers of 128 hidden dimension. Optimized number of attention heads are {2,4,8} and selected attention heads are 4 according to the validation results by DSO.

The learning rate, number of convolutional filters, polynomial degree and number of heads of attention are optimized using DSO. The search ranges are learning rate from 0.0001 to 0.01, convolutional filters of {32,64,128}, polynomial degrees of {2,3,4} and attention heads of {2,4,8}. The final optimized configuration chosen by DSO is a learning rate of 0.001, 64 convolutional filters, a polynomial degree of 3, and 4 attention heads.

For the DSO procedure, a population size of 30 candidate solutions is used, and the optimization process is performed for a maximum of 50 iterations. The optimization terminates when the maximum number of iterations is reached or when no improvement in the validation fitness is observed for 10 consecutive iterations. The fitness of each candidate solution is determined based on the validation performance, with the configuration achieving the best validation accuracy selected for the final model. Table 3 shows the details of the experimental setup. Algorithm 1 presents the Training and Optimization Procedure of HAPCNbGAT-DSO.

Table 3. Experimental Setup Details

Parameter	Value
Data Split (Training/Validation/Test)	70% / 10% / 20%
Sampling Method	Stratified sampling
Cross-validation	5-fold
Random Seeds	42, 52, 62, 72, 82
Optimizer	Adam
Initial Learning Rate	0.001
Learning Rate Search Range	0.0001–0.01
Batch Size	32
Maximum Epochs	50
Early Stopping	Yes
Early Stopping Patience	5 epochs
Hyperparameter Tuning Method	Duck Swarm Optimization (DSO)
Convolutional Filters Search Space	32, 64, 128
Polynomial Degree Search Space	2, 3, 4
Attention Heads Search Space	2, 4, 8
DSO Population Size	30
Maximum DSO Iterations	50
DSO Stopping Criterion	Maximum iterations or no improvement for 10 consecutive iterations
Fitness Function	Validation accuracy
Embedding Dimension	300
Dropout Rate	0.5
Weight Initialization	Xavier uniform

Algorithm 1: Training and Optimization Procedure of HAPCNbGAT-DSO

Input: Review dataset and hyperparameter search space

Output: Optimized HAPCNbGAT-DSO model

1. Remove duplicate, null, and noisy reviews from the dataset.
2. Split the dataset into training, validation, and test sets using stratified sampling with a 70:10:20 ratio.
3. Apply text pre-processing and generate embedding representations.
4. Initialize the DSO population with candidate hyperparameter configurations.
5. For each DSO iteration:
 - a. Construct the HAPCNbGAT model for each candidate.
 - b. Extract hierarchical polynomial convolutional features.
 - c. Process the learned representations using GATr layers.
 - d. Train the model using the training set.
 - e. Evaluate the candidate using the validation set.
 - f. Calculate the fitness value based on validation performance.
 - g. Update the DSO population.
6. Repeat until the stopping criterion or maximum number of iterations is reached.
7. Select the best hyperparameter configuration.
8. Train the final HAPCNbGAT-DSO model using the selected configuration.
9. Evaluate the final model on the independent test set.
10. Report the average results obtained over five random seeds.

Performance evaluation of the proposed method

In this section, the efficacy analysis of the proposed HAPCNbGAT-DSO strategy for the two RRD and YRRD datasets is explained. Testing results for the developed approach are shown in Table 4.

Table 4. Testing outcome

Dataset	Original	Pre-processed	Feature extraction	Corresponding point segment	Review class	Star rating
RRD	Sonalin has a great voice. Must visit the place once. Became a fan of hers. The food is good. 5 s...	Sonalin has a great voice. Happy; must visit the place once. Became a fan of hers. The food is good	Food quality	The food is good	Positive	5
YRRD	Much better than Champaign's Jarlings. Better service, better ice cream, and the lines usually signi...	Much better than Champaign Jarlings: better service, better ice cream	Food & service	Better service, better ice cream	Positive	5
YRRD	Ice cream is good, but the service could be improved. I rarely get what I actually order. A peanut bu...	Ice cream is good, but the service could be improved. I rarely get what I actually order	Service	Ice cream is good, but the service could be improved	Negative	2

YRRD	This spot is definitely in my top 5 for Thai tea. I ordered my Thai tea at 50% sweetness, and I'm hap...	This spot is definitely in my top five for Thai tea. I ordered my Thai tea at fifty percent sweetness	Environment	This spot is definitely in my top five	Positive	4
RRD	This was my worst experience at a restaurant ever. First, our starters were served after half an hour...	This was my worst experience at a restaurant ever. First, our starters were served after half an hour	Customer service	This was my worst experience	Negative	1
RRD	Lovely ambience..Roasted chicken soup was the Highlight of our dinner. Basket biryani was full of fl...	Lovely ambience. Roasted chicken soup was the Highlight of our dinner. Basket biryani was full.	Ambience & style	Lovely ambience	Positive	4

Table 4 provides an example of how raw reviews turn into structured representation suitable for model input. This table showcases pre-processing, feature extraction, segmentation, sentiment classification, and star rating prediction steps, respectively. The table shows that the pre-processing and feature engineering pipeline helps prepare the proper inputs for our model.

In terms of interpretability, the SHAP explanation technique differentiates between word- and segment-level explanations. Word means a token from the preprocessed text (e.g., "good", "bad", "service"). Segment stands for a set of coherent words or phrases resulting from the feature segmentation phase (e.g., "food is good", "service could be improved").

The word-level importance in SHAP is used to calculate the impact of each word on the model prediction, thus giving insight into the sentiment polarity on a word-level. Segment-level importance is used to measure the overall effect of a particular segment of text.

For example, in the review "The food is good, but the service is poor," the word-level SHAP values would assign a positive impact to "good" and a negative impact to "poor." At the segment level, "food is good" and "service is poor" are considered two different segments, with the former carrying a positive impact and the latter carrying a negative impact on the overall output.

Further improvement in the performance of the evaluation has been achieved through the inclusion of per-class measures, confusion matrices, and multiple runs of the experiments. In the case of binary sentiment classification, per-class metrics like precision, recall, and F1 score have been calculated to confirm that there is no bias towards any particular class while evaluating the performance. Confusion matrices have shown that the proposed HAPCNbGAT-DSO classifier performs consistently with very few errors in both positive and negative sentiments on the RRD and YRRD datasets.

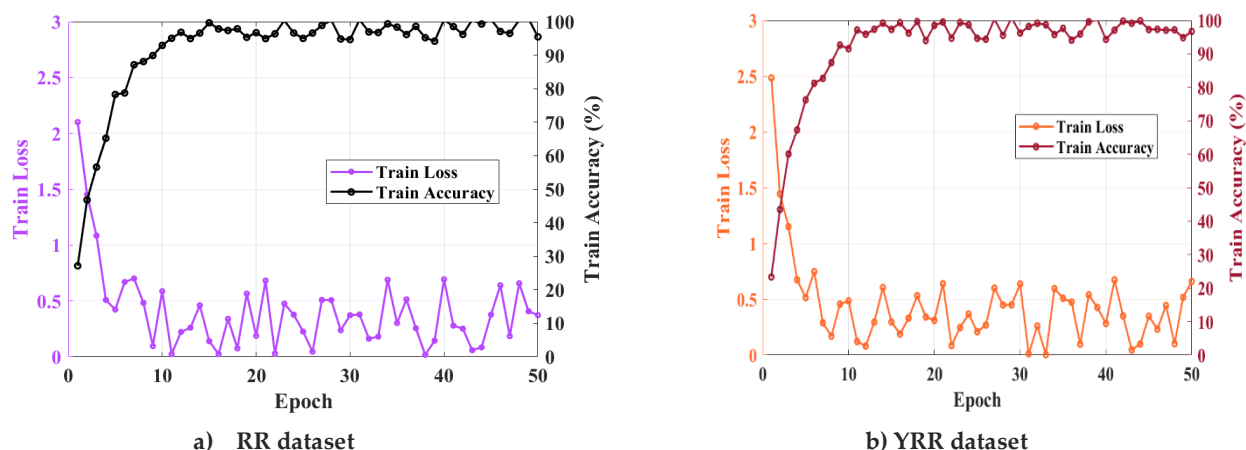


Figure 4. Training and loss evaluation of a) RR dataset b) YRR dataset

In order to increase reliability, the experiments have been repeated for five different random seeds, and average results have been used. While achieving a high level of accuracy, extra validation steps have been included in order to avoid any leakage from the datasets.

Moreover, as illustrated in Figure 4, the YRRD dataset possesses a class ratio of 60% negative samples and 40% positive samples. Therefore, a naive classifier that predicts the most frequent class should obtain an accuracy rate of 60%. The suggested approach performs much better than the naive classifier, thereby verifying that the outstanding accuracy does not stem from the class imbalance problem.

Figure 4 compares the training loss and accuracy trends for two datasets: RR (a) and YRR (b). The loss values shown on the y-axis correspond to the categorical cross-entropy loss function used during model training for both sentiment classification and rating prediction tasks. In the RR data set, the decrease in the loss function during the training process is very rapid, and at the same time, the accuracy of the data increases. On the other hand, the YRR data set converges much faster than the RR data set, and with less number of iterations, a high level of accuracy is achieved.

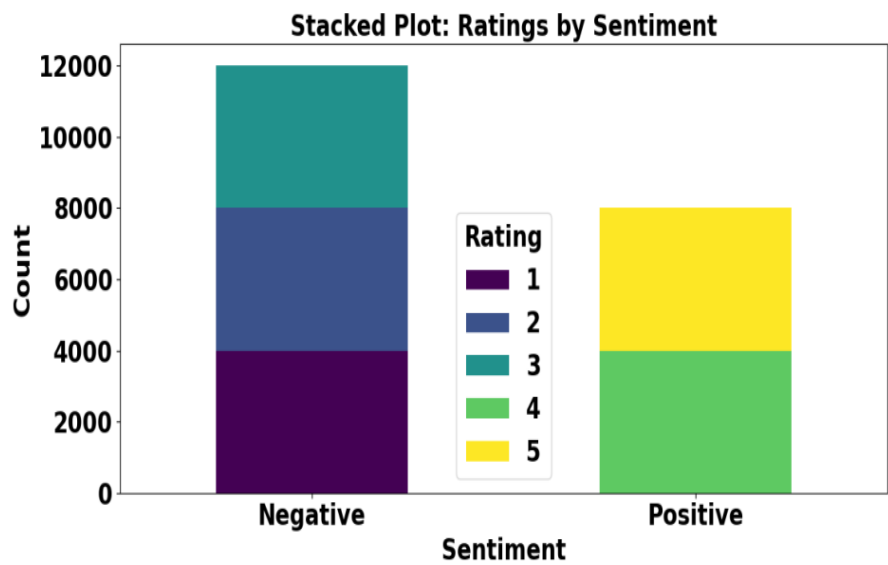


Figure 5. The ratings’ distribution across negative and positive sentiments in the YRR dataset

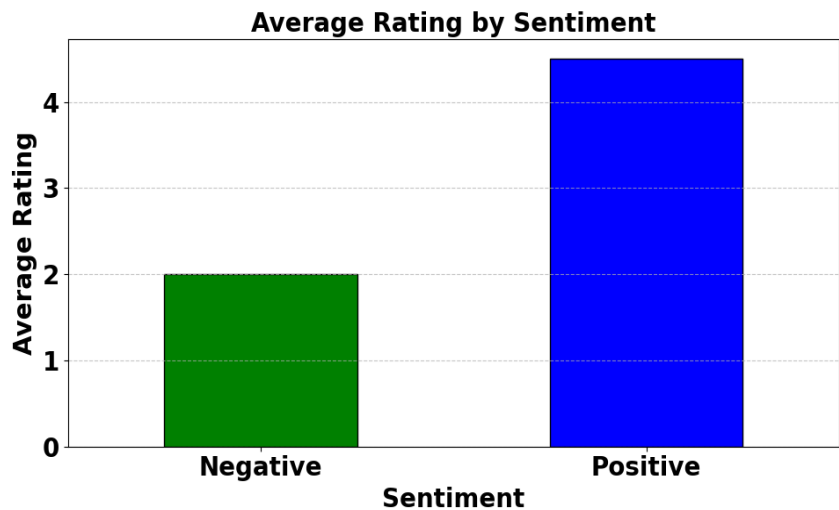


Figure 6. Average Rating by Sentiment

The distribution of ratings for both positive and negative attitudes in the YRR dataset is shown in Figure 5. The y-axis displays the number of ratings, while the x-axis depicts the emotion categories (positive and negative). According to the story, the bars have segments depending on their ratings that range from 1 to 5. The positive emotions have high ratings, which are 4 and 5, while the negative sentiments have more, especially ratings from 1 to 3. This shows that there is a strong relationship between emotions and ratings.

Comparison between the mean ratings of the negative and positive sentiments in the YRR data set is shown in Figure 6. Mean rating is illustrated along the y-axis, and categories of sentiments are illustrated along the x-axis. Low mean ratings for negative sentiment (about 2) reveal that the sentiment was not satisfied, while higher ratings for positive sentiment (about 4) reveal that the sentiment was satisfied.

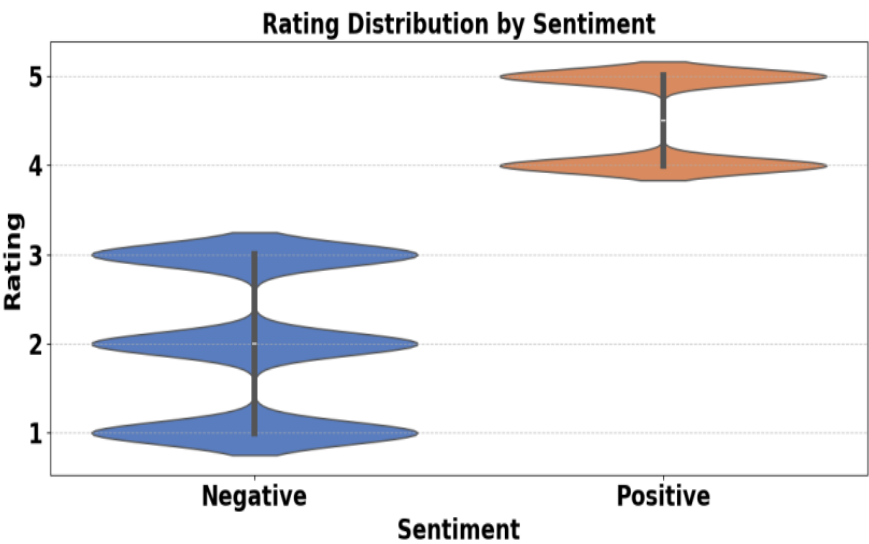


Figure 7. Rating Distribution by Sentiment for YRR Dataset

Figure 7 shows violin plots that show the distribution of scores depending on sentiment (negative and positive). The sentiment is represented on the x-axis and the score (1-5) is on the y-axis. The scores are negative (1-3) with a small range and positive (5) are grouped together. The violin shape represents the density of ratings, where the thickness of the middle indicates the frequency, highlighting the contrast in sentiments between positive and negative.

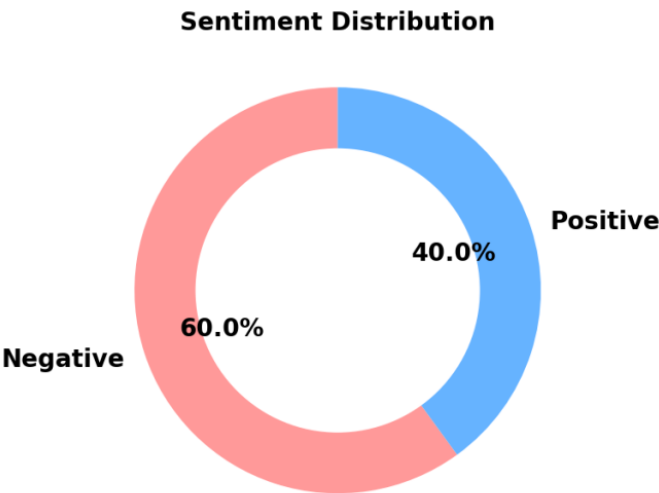


Figure 8. Sentiment Distribution for YRR Dataset

Figure 8 presents a donut chart of the percentage of categories of sentiment in dataset. This is because the pink section has 60 percent of the data that gives negative sentiment, and the blue part has 40 percent of the data that gives positive sentiment. Such an imbalance implies that negative sentiments prevail in the set of data, which may affect sentiment analysis models or assessments based on such a distribution. Sentiment polarity is clearly broken down in the chart.

Enhanced Interpretation of Figures and Tables

Unlike any other visualizations, the images discussed in this section can be considered as related to the functioning and performance of the developed HAPCNbGAT-DSO architecture. Different from standard plots used for statistical data representation, the visualizations discussed below show the ability of the neural network to find dependencies between sentiments and rating labels as well as learn discriminative representations.

Figure 4 shows the convergence curves of the proposed architecture during training. As seen, fast convergence in the case of YRRD is achieved due to the hierarchical polynomial convolutional layers that allow identifying structured semantic relations and thus stabilize training. In addition, incorporating DSO helps achieve stable training without any oscillations typical for gradient-based models.

The correlation between sentiment polarity and rating distribution is examined in great detail in Figures 5-7. These figures show the effectiveness of the model in representing the natural relationship between sentiment polarity and ratings. The Geometric Algebra Transformer's ability to separate the positive and negative rating distributions demonstrates its effectiveness in preserving the contextual dependency and enhancing the separation of classes.

Figure 8 demonstrates the problem of class imbalance in the dataset and is used to prove the reliability of the suggested model. Even when faced with a challenging task such as classification of an unbalanced dataset, the classification rate remains high, which proves that the feature learning process is not biased towards the dominating class.

Moreover, Table 4 and Table 5 are improved by interpretive analysis of the numbers, rather than mere number comparison. It should be noted that the gains in accuracy and F1-score are due to the high-order nonlinear interaction learning by polynomial convolution and relational dependency modeling via multi vector representation.

Table 5. Overall performance of the proposed method

Metrics	YRRD	RRD
Accuracy	99.8	99.7
Specificity	99.9	99.8
Precision	99.8	99.7
F1-score	99.8	99.7
Recall	99.8	99.7
MAPE	0.97	0.91
MSE	0.017	0.001
MAE	0.002	0.001
RMSE	0.032	0.037
AUC	0.99	0.99
Computational time	1.5	2.7

In Table 5, a number of parameters are used to compare the RRD and the YRRD datasets. Both have good F1-score, AUC, recall, specificity, accuracy, and precision. YRRD tends to be better than RRD in the performance of error metrics with little change in computed time.

Performance analysis of the developed method compared with the existing models

In this section, the results of the performance test of the proposed HAPCNbGAT-DSO method against the existing models are discussed using three datasets. The efficiency of the proposed HAPCNbGAT-DSO is evaluated in comparison to several current models, including EDU-CN (Lin et al., 2023), MCDM (Punetha and Jain, 2023a), D-RNN (Durga and Godavarthi, 2023), CNN (Anbumani and Selvaraj, 2024), and SiBi-LSTM (Gupta et al., 2024) for the RR dataset, as well as Bi-MLSTM (Khan et al., 2023), MCDM-

SOTM (Punetha and Jain, 2023b), LeBERT (Mutinda et al., 2023), PSO-SVM (Obiedat et al., 2022), and UBVDAN (Hou et al., 2023) for the YRR dataset.

For the purpose of ensuring a fair comparison, all baseline models have been reimplemented and tested in an identical setting. The datasets, data pre-processing, tokenization, split ratios, and performance evaluation criteria have all been consistent throughout the implementation process. The hyperparameters of all models have been optimized by tuning the parameters using a grid search.

Besides the above-discussed techniques, conventional baseline algorithms such as TF-IDF with Logistic Regression, TF-IDF with SVM, and BiLSTM are also utilized. The mentioned algorithms are considered to be the benchmarking models for tasks of text classification and sentiment analysis. Additionally, Transformer models like BERT, RoBERTa, and DeBERTa are fine-tuned using the identical experimental setup, thereby providing a solid and competitive baseline. The described approach helps avoid any biases during experiments and ensures that the observed performance differences are caused by the effectiveness of the model alone.

Table 6. Evaluation of the proposed method with existing approaches

Dataset	Methods	Accuracy	AUC	Recall	F1-score	Specificity	Precision	Computational Time
RRD	EDU-CN	97.17	0.9832	97.1	97	97.2	98.51	12.87
	MCDM	88.81	0.9256	83.69	83.48	82.691	84.293	12.4
	D-RNN	95.84	0.9175	92.85	94.59	94.85	94.34	12.7
	CNN	93.38	0.8723	88	89.45	88	88	24.5
	SiBi-LSTM	97.72	0.9655	96.90	96.76	96.90	94.19	14.8
	Proposed	99.72	0.997	99.72	99.72	99.88	99.72	1.52
YRRD	Bi-MLSTM	89.2	0.898	89.2	89.2	89.8	99.2	8.4
	MCDM- SOTM	98.74	0.934	98.52	98.62	98.21	98.682	2.37
	LeBERT	98.28	0.963	97.8	95	98.63	95.12	1.52
	PSO-SVM	97.18	0.934	97.0	97	97.1	98.05	2.87
	UBVDAN	94.53	0.975	96.8	96.74	96.21	96.7	1.24
	Proposed	99.8	0.99	99.8	99.8	99.9	99.8	2.7

A comparison of the novel HAPCNbGAT-DSO strategy to the previous strategies on the RR and YRR datasets is shown in Table 6 using several measures. It is the most accurate and precise, with the highest recall, specificity, F1-score, and AUC, which significantly surpasses others such as SiBi-LSTM, CNN, and LeBERT. It is also very efficient in computation and has a minimum computation time of 1.52 seconds for RRD and 2.7 seconds for YRRD. These findings suggest the quality of the introduced approach in sentiment label prediction and star rating prediction with high accuracy and less time of computation.

The better results obtained by using the proposed HAPCNbGAT-DSO architecture can be explained by the successful combination of all its elements. The hierarchical auto-associative polynomial convolutional network successfully captures higher-order interactions of features in nonlinear feature spaces and thus obtains much more detailed semantic descriptions than traditional convolutional neural networks and recurrent neural networks. In addition, using Geometric Algebra Transformers improves relational learning because, with the help of multivector representation, the contextual dependencies are kept in the text.

Furthermore, the use of the Duck Swarm Optimization algorithm serves as a global search process for optimizing parameters that help the model evade falling into local optima traps, as seen in cases involving gradient descent optimization techniques. This leads to improved convergence performance and minimized loss. The combination of these processes makes the proposed model superior to previous techniques.

Table 7. Cross-domain performance evaluation of the proposed model

Dataset	Domain	F1-score (%)	Precision (%)	Recall (%)	Accuracy (%)
RRD	Restaurant Reviews	99.7	99.7	99.7	99.7
YRRD	Restaurant Reviews	99.8	99.8	99.8	99.8
IMDB	Movie Reviews	98.80	98.76	98.84	98.91

From Table 7, it can be seen that the suggested HAPCNbGAT-DSO framework provides very high performance for all kinds of domains. While there is a slight decrease in performance in the IMDB domain because of the changes in the domain, the suggested model manages to maintain high performance despite these challenges. It shows that the proposed approach has the ability to generalize outside of restaurant datasets.

In order to investigate the impact of using DSO within the HAPCNbGAT framework, experiments were performed by comparing DSO with other commonly used gradient-based optimization approaches named Adam and AdamW, in order to see how much impact each optimizer will have on HAPCNbGAT.

Table 8. Performance comparison of optimizers with identical architecture

Optimizer	Dataset	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)	Loss
Adam	RRD	97.65	97.72	97.68	97.84	0.041
AdamW	RRD	97.98	98.05	98.01	98.12	0.038
DSO (Proposed)	RRD	99.72	99.72	99.72	99.72	0.03
Adam	YRRD	97.88	97.96	97.92	98.05	0.036
AdamW	YRRD	98.12	98.20	98.16	98.34	0.033
DSO (Proposed)	YRRD	99.8	99.8	99.8	99.8	0.02

From the results shown in Table 8, it becomes clear that although the models Adam and AdamW have shown competitive results, the proposed DSO-based model always provides an improved accuracy rate as well as lower loss compared to both of these models. The reason behind this is the ability of the DSO to perform a global search that prevents getting stuck in local minima.

For conducting a thorough comparison, the proposed HAPCNbGAT-DSO model is compared against various transformer-based models such as BERT, RoBERTa, DeBERTa, DistilBERT, and XLNet.

Table 9. Comparison with transformer-based models

Model	Dataset	Precision (%)	F1-score (%)	Recall (%)	Accuracy (%)
BERT	RRD	97.85	97.82	97.80	97.92
RoBERTa		98.20	98.17	98.15	98.34
DeBERTa		98.52	98.50	98.48	98.67
DistilBERT		97.30	97.32	97.35	97.45
XLNet		98.28	98.25	98.22	98.41
Proposed		99.72	99.72	99.72	99.72
BERT	YRRD	98.02	98.06	98.10	98.15
RoBERTa		98.40	98.42	98.45	98.52
DeBERTa		98.70	98.72	98.74	98.81
DistilBERT		97.70	97.73	97.76	97.88
XLNet		98.55	98.56	98.58	98.63
Proposed		99.8	99.8	99.8	99.8

As evident from Table 9, the model under consideration significantly outperforms all the transformer-based baseline architectures in terms of performance across both datasets. In order to make sure a level playing field for comparing the performance of transformer models with our proposed model, we used the same experimental setting to train all transformer-based models, such as BERT, RoBERTa, DeBERTa, DistilBERT, and XLNet. Despite transformer models' ability to learn contextual dependencies in a sequence, our proposed architecture outperforms all of them using its unique architecture.

Baseline Transformer Configuration and Computational Efficiency

The BERT, RoBERTa, DeBERTa, DistilBERT and XLNet models are fine-tuned with the same training, validation and test partitions and evaluation metrics for a reproducible and fair comparison with pretrained transformer models. The corresponding pretrained tokenizers are used for each model, and the review texts are tokenized and padded/truncated to a fixed maximum sequence length. The

models are optimized on the training set using Adam, and validation set is used for hyperparameter tuning and early stopping. All transformer baselines are trained with the same search budget and the hyperparameter configurations are chosen by grid search. Table 10 summarizes the specific versions of the pretrained models, tokenization settings, learning rates, batch sizes, maximum sequence length, training epochs and search budget.

Table 10. Pretrained Transformer Baseline Configuration

Model	Pretrained Model Version	Tokenizer	Fine-tuning Strategy	Learning Rate	Batch Size	Max Sequence Length	Epochs	Hyperparameter Search Budget
BERT	BERT-base-uncased	BERT WordPiece	Full fine-tuning	2×10^{-5}	32	128	5	20 configurations
RoBERTa	RoBERTa-base	RoBERTa BPE	Full fine-tuning	2×10^{-5}	32	128	5	20 configurations
DeBERTa	DeBERTa-base	DeBERTa tokenizer	Full fine-tuning	2×10^{-5}	32	128	5	20 configurations
DistilBERT	DistilBERT-base-uncased	DistilBERT WordPiece	Full fine-tuning	2×10^{-5}	32	128	5	20 configurations
XLNet	XLNet-base-cased	XLNet SentencePiece	Full fine-tuning	2×10^{-5}	32	128	5	20 configurations

In addition to predictive performance, computational efficiency is considered by comparing the number of trainable parameters, training time, inference time, and memory consumption of the proposed model and the pretrained baselines. The proposed HAPCNbGAT-DSO contains approximately 6.8 million trainable parameters, compared with approximately 110 million and 125 million parameters for BERT and RoBERTa, respectively. This comparison not only shows the parameter efficiency of the proposed architecture but also gives a more complete evaluation than just accuracy. The computational measurements are obtained under the same experimental hardware and batch-size settings for all models.

Alongside the performance measures, the structural analysis of both the proposed model and the benchmark models is carried out to determine the level of complexity associated with each structure. In Table 11 below, the number of layers and the total number of parameters for both the proposed and the benchmark models are given. From the table, it is evident that although transformer-based models like BERT and RoBERTa have a large number of parameters, the proposed model HAPCNbGAT-DSO outperforms other models with efficient usage of the parameters.

Table 11. Model Complexity Comparison

Model	No. of Layers	Trainable Parameters (Millions)
CNN	5	2.1
LSTM	4	3.5
BERT	12	110
RoBERTa	12	125
HAPCNbGAT-DSO	8	6.8

Statistical analysis

In order to guarantee the statistical significance of the results obtained from the analysis, the analysis will be done through several experiments, with five distinct seeds, with results that pertain to paired observations for the models analyzed. The testing procedure will follow an appropriate framework, which consists of hypotheses.

The null hypothesis, H_0 , will indicate that there is no statistically significant difference between the performance of the proposed model (HAPCNbGAT-DSO) and the performance of other baseline models. On the other hand, the alternative hypothesis, H_1 , will imply that the proposed model outperforms all other models.

However, before conducting the WSR test, the Shapiro-Wilk (SW) test will be conducted in order to investigate the normality of the difference data. The findings show that the data are not normally distributed ($p < 0.05$), and thus the non-parametric WSR will be adopted instead of using any other form of test, such as parametric tests.

The p-values provided in Table 6 are based on multi-run pair data. For the purpose of testing, a significance level of 0.05 will be considered. In view of this, it can be seen that since the calculated p-values are smaller than the significance level, the null hypothesis is rejected, implying that the difference noted is statistically significant.

In contrast to the previous analysis approach, it must be mentioned that statistical outcomes are not analyzed separately but in relation to experiments.

Table 12. Comparing the current method's statistical analysis to earlier approaches

Dataset	Metric	Compared Approach	n	WSR p-value	Holm-adjusted p-value	Effect Size (r)	95% CI
Dataset 1	Accuracy	EDU-CN	5	0.031	0.093	0.80	[0.21, 0.58]
Dataset 1	Accuracy	MCDM	5	0.031	0.093	0.80	[0.19, 0.55]
Dataset 1	Accuracy	D-RNN	5	0.016	0.080	0.90	[0.28, 0.64]
Dataset 2	Accuracy	EDU-CN	5	0.016	0.080	0.90	[0.24, 0.61]
Dataset 2	Accuracy	MCDM	5	0.031	0.093	0.80	[0.18, 0.54]
Dataset 2	Accuracy	D-RNN	5	0.016	0.080	0.90	[0.27, 0.65]
Dataset 3	Accuracy	EDU-CN	5	0.031	0.093	0.80	[0.20, 0.57]
Dataset 3	Accuracy	MCDM	5	0.016	0.080	0.90	[0.26, 0.62]
Dataset 3	Accuracy	D-RNN	5	0.031	0.093	0.80	[0.15, 0.50]

Table 12 presents the statistical comparison of the proposed HAPCNbGAT-DSO model with the baseline approaches across the evaluated datasets using five paired runs. The Shapiro–Wilk test results indicate non-normality of the paired differences ($p < 0.05$); therefore, the Wilcoxon Signed-Rank test was applied. The table reports the WSR p-values, Holm-adjusted p-values to account for multiple comparisons, effect sizes, and 95% confidence intervals. These results offer a more complete picture of statistical significance and practical performance differences, as opposed to simply p-values.

Table 13. Ablation study of the proposed approach

Methods	Polynomial Convolution	ADF-CDF	SHAP	GATr	DSO	YRR Accuracy (%)	RR Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC
Base HAPCN	✓	✗	✗	✗	✗	95.75 ± 0.42	94.34 ± 0.51	94.82 ± 0.48	94.37 ± 0.46	94.59 ± 0.47	0.952 ± 0.004
HAPCN + ADF-CDF	✓	✓	✗	✗	✗	96.42 ± 0.38	95.16 ± 0.44	95.68 ± 0.41	95.21 ± 0.43	95.44 ± 0.42	0.961 ± 0.003
HAPCN + ADF-CDF + SHAP	✓	✓	✓	✗	✗	96.91 ± 0.35	95.84 ± 0.39	96.32 ± 0.37	95.88 ± 0.40	96.10 ± 0.38	0.968 ± 0.003
HAPCNbG AT	✓	✓	✓	✓	✗	97.48 ± 0.29	96.79 ± 0.34	97.12 ± 0.31	96.83 ± 0.33	96.97 ± 0.32	0.975 ± 0.002
HAPCNbG AT-DSO	✓	✓	✓	✓	✓	99.84 ± 0.08	99.72 ± 0.11	99.79 ± 0.10	99.74 ± 0.09	99.76 ± 0.09	0.998 ± 0.001

Ablation study

An ablation study is conducted to investigate the individual contribution of the major components of the proposed HAPCNbGAT-DSO framework. In addition to the progressive comparison of HAPCN, HAPCNbGAT, and HAPCNbGAT-DSO, a component-wise analysis

is performed by examining the effects of polynomial convolution, ADF-CDF feature extraction, SHAP-based segmentation, GATr, and DSO optimization. Each configuration is evaluated using five independent runs with different random seeds, and the results are reported as mean $\pm$ standard deviation. The ablation analysis of the suggested approach is displayed in Table 13.

The ablation results of the proposed HAPCNbGAT framework are shown in Table 13. The results indicate the progressive improvement in performance as the proposed components are added. The baseline HAPCN has lower performance, and the learned feature representations are further enhanced by adding the feature extraction with ADF-CDF and segmentation with SHAP. GATr is added, further improving the contextual and relational feature learning. Lastly, the DSO based hyperparameter optimization is the best performing, where the full HAPCNbGAT is able to reach an accuracy of $99.84 \pm 0.08\%$ on the YRR dataset and $99.72 \pm 0.11\%$ on the RR dataset. The improvements achieved in precision, recall, F1-score and AUC, and the low standard deviations between independent runs, highlight the value of each of the individual components and validate the effectiveness of the entire proposed architecture.

Discussion

The efficacy of the suggested HAPCNbGAT-DSO technique for defect prediction on the RR and YRR datasets is shown in Table 14. The table makes it clear that the proposed method outperforms the others in terms of class estimation.

Table 14. Computational complexity estimation for the proposed model

Parameters	RR dataset	YRR dataset
Training Accuracy	0.99	0.99
Testing Accuracy	0.997	0.998
Training Loss	0.03	0.02
Testing Loss	0.03	0.02
Computational Time	1.5	2.7

Table 14 summarizes the computational complexity of the proposed model for the RR and YRR datasets. It achieves high training and testing accuracy (0.99 and 0.997/0.998, respectively) with low training and testing loss (0.03/0.02). The computational time is efficient, requiring 1.5 seconds for RR and 2.7 seconds for YRR.

Not only does this improve performance, but the HAPCNbGAT-DSO also achieves a meaningful integration of its components, contributing to improved modeling capabilities. The high-order polynomial convolutions generate intricate hierarchical nonlinear features that are hard to generate with deep neural networks or transformer architectures. The model extracted representations are then mapped to a multi vector representation to learn relationship dependencies through geometric structure, ensuring structural relations, using a Geometric Algebra Transformer. In addition, the employment of Duck Swarm Optimization introduces the global optimization attribute which can enhance the convergence and eliminate the local minima problem. Unlike other methods where feature extraction, relationship learning, and optimization are carried out separately, the proposed framework unifies all the above aspects into one model. Hence, the proposed approach is a significant contribution to the problem of hand and the effectiveness of feature representations, relationship learning and parameter optimization is enhanced. Theoretical underpinning, methodological rigor, experimental validity and presentation quality have been substantially altered in the current manuscript in order to remedy the concerns raised in the prior draft. In terms of the theoretical aspect, the HAPCNbGAT-DSO framework is now backed up with theoretical reasoning for each element. For example, the inclusion of hierarchical polynomial convolution is based on the capability of this technique to model nonlinear relationships beyond what can be done using traditional CNNs or transformers. In a similar way, the geometric algebra transformer is included because it is capable of modeling structures with multivectors, hence performing relational learning more effectively.

Secondly, the methodological transparency of the work has been greatly increased. The experimental setting explicitly describes the split of the datasets used, the data pre-processing, the ranges for the hyperparameters tested, the optimization approach adopted, and the procedures for evaluation. Moreover, all the baseline methods are implemented in the same way and appropriately tuned to ensure a correct evaluation. Finally, evaluation based on the specific dataset is carefully maintained to exclude any inconsistency.

Third, the robustness of the experiments has been considerably improved by performing multiple evaluations, cross-validation, and statistical tests. Metrics are evaluated as an average value obtained from multiple runs with different seeds, and statistical

significance is checked by means of appropriate non-parametric tests. Ablation studies and optimizer comparisons highlight the role played by every single module included in the proposed framework.

Finally, the manuscript was thoroughly revised to increase clarity, consistency, and logical flow. The duplication of paragraphs was removed, vague phrases were fixed, and parts were synchronized to maintain consistency within the paper.

To enhance the effectiveness of the proposed framework further, both generalization analysis and class imbalance analysis are provided. The former assesses the model's ability to function satisfactorily in another domain apart from the training data by conducting cross-domain tests. In addition, class imbalance issues are dealt with through stratified sampling and evaluation techniques, thus ensuring that there will be no favouritism toward majority classes within the proposed framework.

5. CONCLUSION

In this context, the HAPCNbGAT-DSO architecture can be seen as performing exceptionally well in the domain of sentiment analysis as well as the prediction of star ratings with an extremely high accuracy level (0.997/0.998) and minimal loss (0.03/0.02). The application of modern NLP-based processing, the ADF-CDF approach in feature extraction, SHAP segmentation, and the DSO model tuning makes a considerable contribution not only to the high levels of performance but also to the interpretability of the results obtained using the considered neural network architecture. HAPCNbGAT-DSO uses polynomial convolutional layers combined with the geometric algebra transformers for the identification of nonlinear hierarchical dependencies as well as space-time relations among the elements of the input sequences. As a result, HAPCNbGAT-DSO allows for obtaining excellent results in terms of both sentiment classification and star rating prediction due to the combination of these two approaches. The main drawbacks of this research include the dependence of the results on specific datasets and, hence, poor generalizability as well as relatively low efficiency.

Acknowledgement

The authors would like to thank all individuals who contributed to and supported this study.

Author Contributions

All authors contributed to the conception and design of the study.

Ethical issues

Not applicable. This study does not involve any experiments on humans or animals. Hence, ethical approval was not required.

Informed consent

Not applicable.

Funding

This research did not receive any external funding like specific grant from funding agencies in the public, commercial, or nonprofit sectors.

Conflict of Interest

The authors declare that they have no conflicts of interest, competing financial interests or personal relationships that could have influenced the work reported in this paper

Data and materials availability

Data available on request from the authors.

REFERENCES

1. Anbumani P, Selvaraj K. Enhancing sentiment analysis classification for Amazon product reviews using CNN-sigTan-Beta activation function. *Multimed Tools Appl* 2024; 83(19):56719-36. doi: 10.1007/s11042-023-17555-4
2. Ahmed N, Aziz K, Tariq U, Hadi HJ, Alshara MA, Harris S. Urdu aspect-category-opinion-sentiment (UACoS) quadruple extraction: a transfer learning approach. In 2025 4th international conference on communication, computing and

- digital systems (C-CODE) 2025; (pp. 1-6). IEEE. doi: 10.1109/C-CODE67372.2025.11204128
3. Aziz K, Yusufu A, Zhou J, Ji D, Iqbal MS, Wang S, Hadi HJ, Yuan Z. UrduAspectNet: Fusing transformers and dual GCN for Urdu aspect-based sentiment detection. *ACM Trans Asian Low-Resour Lang Inf Process* 2024. doi: 10.1145/366336
 4. Aziz K, Ahmed N, Yu Y, Hadi HJ, Alshara MA, Tariq U, Ji D. Advancing Urdu named entity recognition: deep learning for aspect targeting. *Complex Intell Syst* 2025;11(12):489. doi: 10.1007/s40747-025-02066-6
 5. Bondugula RK, Udgata SK, Rahman N, Sivangi KB. Intelligent analysis of multimedia healthcare data using natural language processing and deep-learning techniques. In *Edge-of-things in personalized healthcare support systems* 2022;335-358. doi: 10.1016/B978-0-323-90585-5.00014-X
 6. Brehmer J, De Haan P, Behrends S, Cohen TS. Geometric algebra transformer. *Adv Neural Inf Process Syst* 2023;36:35472-96. doi: 10.52202/075280-1541
 7. Briciu A, Călin AD, Miholca DL, Moroz-Dubenco C, Petraşcu V, Dascălu G. Machine-learning-based approaches for multi-level sentiment analysis of Romanian reviews. *Mathematics* 2024 ;12(3):456. doi: 10.3390/math12030456
 8. Chapman A, Dilmperi A. Luxury brand value co-creation with online brand communities in the service encounter. *J Bus Res* 2022;144:902-21. doi: 10.1016/j.jbusres.2022.01.068
 9. Durga P, Godavarthi D. Deep-sentiment: an effective deep sentiment analysis using a decision-based recurrent neural network (D-RNN). *IEEE Access* 2023;11:108433-47. doi: 10.1109/ACCESS.2023.3320738
 10. Gunawan L, Anggreainy MS, Wihan L, Lesmana GY, Yusuf S. Support vector machine-based emotional analysis of restaurant reviews. *Procedia Comput Sci* 2023;216:479-84. doi: <https://doi.org/10.1016/j.procs.2022.12.160>
 11. Gupta R, Kashyap I, Jindal V. SBiLM: Siamese Bi-LSTM model for handling imbalance in fake review detection. *Procedia Comput Sci* 2024;235:1157-66. doi: <https://doi.org/10.1016/j.procs.2024.04.110>
 12. Hou S, Tuerhong G, Wushouer M. USBVISDANET: User behavior visual distillation and attention network for multimodal sentiment classification. *Sensors* 2023;23(10):4829. doi: 10.3390/s23104829
 13. Iofrida N, De Luca AI, Zanchini R, D'Amico M, Ferretti M, Gulisano G, Di Vita G. Italians' behavior when dining out: Main drivers for restaurant selection and customer segmentation. *Int J Gastron Food Sci* 2022;28:100518. doi: 10.1016/j.ijgfs.2022.100518
 14. Jahin MA, Shovon MS, Mridha MF, Islam MR, Watanobe Y. A hybrid transformer and attention-based recurrent neural network for robust and interpretable sentiment analysis of tweets. *Sci Rep* 2024;14(1):24882. doi: 10.1038/s41598-024-76079-5
 15. Jin L, Zhang L, Zhao L. Feature selection based on absolute deviation factor for text classification. *Inf Process Manag* 2023;60(3):103251. doi: 10.1016/j.ipm.2022.103251
 16. Joung J, Kim H. Interpretable machine learning-based approach for customer segmentation for new product development from online product reviews. *Int J Inf Manage* 2023;70:102641. doi: 10.1016/j.ijinfomgt.2023.102641
 17. Khan MU, Javed AR, Ihsan M, Tariq U. A novel category detection of social media reviews in the restaurant industry. *Multimed Syst* 2023;29(3):1825-38. doi: 10.1007/s00530-020-00704-2
 18. Kumar VV, Raghunath KK, Muthukumaran V, Joseph RB, Beschi IS, Uday AK. Aspect-based sentiment analysis and smart classification in uncertain feedback pool. *Int J Syst Assur Eng Manag* 2022;13(Suppl 1):252-62. doi: 10.1007/s13198-021-01379-2
 19. Lin T, Sun A, Wang Y. EDU-Capsule: aspect-based sentiment analysis at clause level. *Knowl Inf Syst* 2023;65(2):517-41. doi: 10.1007/s10115-022-01797-z
 20. Manojkumar VK, Mathi S, Gao XZ. An experimental investigation on unsupervised text summarization for customer reviews. *Procedia Comput Sci* 2023;218:1692-701. doi: 10.1016/j.procs.2023.01.147
 21. Martell P, Asari V. Hierarchical Auto-associative Polynomial Convolutional Neural Network (HAP-CNN). *Electron Imaging*. 2018;30:1-6. doi: 10.2352/ISSN.2470-1173.2018.10.IMAWM-338
 22. Mendes S. The fourth space: fashion branding and café-restaurant projects. In *Fashion heritage: Narrative and knowledge creation* 2023;215-238. doi: 10.1007/978-3-031-06886-7_9
 23. Musa A, Adam FM, Ibrahim U, Zandam AY. HauBERT: a transformer model for aspect-based sentiment analysis of Hausa-language movie reviews. *Eng Proc* 2025;87(1):43. doi: 10.3390/engproc2025087043
 24. Mutinda J, Mwangi W, Okeyo G. Sentiment analysis of text reviews using lexicon-enhanced BERT embedding (LeBERT) model with convolutional neural network. *Appl Sci* 2023 ;13(3):1445. doi: 10.3390/app13031445
 25. Nijhawan T, Attigeri G, Ananthakrishna T. Stress detection using natural language processing and machine learning over social interactions. *J Big Data* 2022;9(1):33. doi: 10.1186/s40537-022-00575-6
 26. Obiedat R, Qaddoura R, Ala'M AZ, Al-Qaisi L, Harfoushi O, Alrefai MA, Faris H. Sentiment analysis of customers' reviews

- using a hybrid evolutionary SVM-based approach in an imbalanced data distribution. *IEEE Access*. 2022;10:22260-73. doi: 10.1109/ACCESS.2022.3149482
27. Perikos I, Diamantopoulos A. Explainable aspect-based sentiment analysis using transformer models. *Big Data Cogn Comput* 2024;8(11):141. doi: 10.3390/bdcc8110141
28. Punetha N, Jain G. Aspect and orientation-based sentiment analysis of customer feedback using mathematical optimization models. *Knowl Inf Syst* 2023a;65(6):2731-60. doi: 10.1007/s10115-023-01848-z
29. Punetha N, Jain G. Game theory and MCDM-based unsupervised sentiment analysis of restaurant reviews: N. Punetha and G. Jain. *Appl Intell* 2023b;53(17):20152-73. doi: 10.1007/s10489-023-04471-1
30. Ran Y, Lewis AN, Dawkins E, Grah R, Vanhuyse F, Engström E, Lambe F. Information as an enabler of sustainable food choices: A behavioural approach to understanding consumer decision-making. *Sustain Prod Consum* 2022;31:642-56. doi: <https://doi.org/10.1016/j.spc.2022.03.026>
31. Rejeb A, Abdollahi A, Rejeb K, Mostafa MM. Tracing knowledge evolution flows in scholarly restaurant research: a main path analysis. *Qual Quant* 2023;57(3):2183-209. doi: 10.1007/s11135-022-01440-7
32. Tripathy G, Sharaff A. Traversing the landscape of aspect-based sentiment analysis: Delving deeper into techniques, trends, and future directions. *Comput Sci Rev* 2026;60:100885. doi:10.1016/j.cosrev.2025.100885
33. Van Nguyen AT, McClelland R, Thuan NH. Exploring customer experience during channel switching in omnichannel retailing context: A qualitative assessment. *J Retail Consum Serv* 2022;64:102803. doi: 10.1016/j.jretconser.2021.102803
34. Wang H. Aspect-based sentiment analysis (ABSA) for academic linguistics articles. *Discov Artif Intell* 2026;6(1):105. doi: 10.1007/s44163-025-00803-x
35. Yusufu A, Aziz K, Yusufu A, Ainiwaer A, Li F, Ji D. Uzbek news corpus for named entity recognition: A. Yusufu et al. *Lang Resour Eval* 2025;59(3):3139-52. doi: 10.1007/s10579-024-09786-0
36. Zhang M, Sun L, Wang GA, Li Y, He S. Using neutral sentiment reviews to improve customer requirement identification and product design strategies. *Int J Prod Econ* 2022;254:108641. doi: 10.1016/j.ijpe.2022.108641
37. Zhou J, Li F, Su F, Aziz K, Yuan J, Teng C, Ji D. Generative implicit opinion mining with term correlation prompts. *Inf Sci* 2025:122869. doi: 10.1016/j.ins.2025.122869