Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.033915
Article

Tech Science Press

Political Optimizer with Probabilistic Neural Network-Based Arabic Comparative Opinion Mining

Najm Alotaibi[1], Badriyya B. Al-onazi[2], Mohamed K. Nour[3], Abdullah Mohamed[4], Abdelwahed Motwakel[5,*], Gouse Pasha Mohammed[5], Ishfaq Yaseen[5] and Mohammed Rizwanullah[5]

[1]Prince Saud AlFaisal Institute for Diplomatic Studies, Riyadh, 13369, Saudi Arabia
[2]Department of Language Preparation, Arabic Language Teaching Institute, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Makkah, 24211, Saudi Arabia
[4]Research Centre, Future University in Egypt, New Cairo, 11845, Egypt
[5]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, 16242, Saudi Arabia
*Corresponding Author: Abdelwahed Motwakel. Email: a.ismaeil@psau.edu.sa

Received: 01 July 2022; Accepted: 14 October 2022

Abstract: Opinion Mining (OM) studies in Arabic are limited though it is one of the most extensively-spoken languages worldwide. Though the interest in OM studies in the Arabic language is growing among researchers, it needs a vast number of investigations due to the unique morphological principles of the language. Arabic OM studies experience multiple challenges owing to the poor existence of language sources and Arabic-specific linguistic features. The comparative OM studies in the English language are wide and novel. But, comparative OM studies in the Arabic language are yet to be established and are still in a nascent stage. The unique features of the Arabic language make it essential to expand the studies regarding the Arabic text. It contains unique features such as diacritics, elongation, inflection and word length. The current study proposes a Political Optimizer with Probabilistic Neural Network-based Comparative Opinion Mining (POPNN-COM) model for the Arabic text. The proposed POPNN-COM model aims to recognize comparative and non-comparative texts in Arabic in the context of social media. Initially, the POPNN-COM model involves different levels of data pre-processing to transform the input data into a useful format. Then, the pre-processed data is fed into the PNN model for classification and recognition of the data under different class labels. At last, the PO algorithm is employed for fine-tuning the parameters involved in this model to achieve enhanced results. The proposed POPNN-COM model was experimentally validated using two standard datasets, and the outcomes established the promising performance of the proposed POPNN-COM method over other recent approaches.

Keywords: Comparative opinion mining; Arabic text; social media; parameter tuning; machine learning; political optimizer

1 Introduction

Social media has emerged as a powerful platform for obtaining different types of data from different fields. The advancement and the application of computerized techniques in the extraction of information and knowledge from Arabic text in mass media are still in a nascent stage due to the complex nature of the language [1]. Various techniques such as normalization, learning algorithms, text cleaning and the Natural Language Processing (NLP) method have recently evolved. Arabic is a Semitic language that closely relates to Islam and the Arabic culture. The Arabic language is used to write the verses in the holy book of Muslims [2]. It is an official language for 422 million people spread across the globe. Along with cultural uniqueness and rich history, the Arabic language has a distinct structure and nature compared to other languages, such as English. For instance, Arabic is written from right to left [3,4]. With a total of 28 letters, the language has three vowels and diacritics. The shapes of the letters in Arabic text vary under their position in a word and the absence of capitalization [5]. Arabic is a structured language with a large vocabulary. Its morphology serves a significant role. Further, the words are frequently built in a complicated manner [6,7] and might contain agglutinative words, drop features and affixes. The Arabic language's characteristics and complexities make its opinion mining process a highly challenging one [8].

Businesses acquire insights about their customer purchase behaviour to gain an advantage over their competitors. These insights explain why customers are not purchasing a company's products and prefer their competitors' products [9]. Such crucial information gives the companies an overview of whether a new product or a new service is liked or not by the consumer. Public Facebook review pages, customer product review pages and blogs are filled with opinions of the end users instead of the entities that compare the outcomes against their competitors [10,11]. Mostly, the political reviews that compare a government's performance are important for the decision-makers in order to ensure that the people find their government's performance satisfactory and the country remains economically stable. This study examined different types of comparative sentences used in Arabic to extract and analyse the sentiments expressed in the public domain [12]. Comparative relations denote the extraction of the comparative opinion components. Many research works conducted in the comparative opinion mining domain are concerned with identifying comparative relationships and general classification of such relationships under pre-defined comparative classes [13]. Even though such efforts are highly significant in opinion mining, it becomes insufficient when the comparative opinions are recognized in relation to the argument [14]. Numerous research articles have been published on comparative opinions in the English language. However, the studies concerning the Arabic language focused only on the detection of comparative relations so far.

Aljedani et al. [15] proposed a hierarchical multi-label classification framework for the Arabic language. This study projected a Hierarchical Multi-label Arabic Text Classification (HMATC) technique with Machine Learning (ML). The study evaluated the impact of the feature set dimensions and the Feature Selection (FS) approaches on the classification performance. The outcomes of the Hierarchy of Multilabel ClassifiER (HOMER) approach were maximized by analysing different sets of multi-label classifiers, clustering methods and various clusters to enhance the performance of the hierarchical classification process. Boukil et al. [16] introduced a creative approach to Arabic text classification in which an Arabic stemming technique was used for extraction, selection and reducing the required features. Afterwards, the Term Frequency-Inverse Document Frequency (TF-IDF) method was leveraged as a feature weighting algorithm. At last, CNN, a Deep Learning technique, was used for classification. Though this technique is highly effective in other domains like pattern recognition and image processing, it is hardly utilized in text mining.

Bahassine et al. [17] modelled an enhanced algorithm for the classification of Arabic texts in which the Chi-square Feature Selection (FS) (referred to here as ImpCHI) method was used to enhance the classification performance. The study compared the outcomes of the enhanced chi-square method using three conventional FS metrics: the Chi-square method, mutual information and the information gain approach. Chantar et al. [18] projected an enhanced binary Grey Wolf Optimizer (GWO) wrapper FS method to handle the Arabic text classification process appropriately. The presented binary GWO was helpful as a wrapper-related FS method. The outcomes of the projected technique were compared with that of the established learning methods such as the Decision Tree (DT), k-Nearest Neighbour (KNN), Naïve Bayes (NB) and the Support Vector Machine (SVM) model. Muaad et al. [19] modelled a method to recognize and represent the Arabic text at the character level with the help of a Deep Convolutional Neural Network (CNN) method. In this study, the CNN method was authenticated through a 5-fold cross-validation test for classifying the Arabic text documents. In addition, the authors employed a mechanism to evaluate the Arabic texts. Finally, the ArCAR mechanism revealed the proposed method's capability in character-level Arabic text classification.

The current study proposes a Political Optimizer with Probabilistic Neural Network-based Comparative Opinion Mining (POPNN-COM) model to classify the Arabic text. The goal of the proposed POPNN-COM model is to recognize comparative and non-comparative texts in Arabic social media content. Initially, the POPNN-COM model involves different levels of data pre-processing to transform the input data into a useful format. For classification, the pre-processed data is fed into the PNN model to recognize the class labels. At last, the PO algorithm is employed for fine-tuning the parameters involved in the model to achieve enhanced results. The proposed POPNN-COM model was experimentally validated using two datasets, and the results were discussed under different measures.

2 The Proposed Model

In the current study, a novel POPNN-COM model has been developed for opinion mining from the Arabic text. The goal of the proposed POPNN-COM model is to recognize comparative and non-comparative texts in the context of Arabic social media content.

2.1 Data Pre-Processing

In this stage, the POPNN-COM model involves different levels of data pre-processing to transform the input data into a useful format. The pre-processing stage aims to remove the noise from the text and reduce the number of features during text presentation. This process results in fewer storage requirements and high classification efficiency. The text pre-processing stage involves multiple sub-processes for the Arabic language. At first, the symbols such as "#" (hashtag), URLs, "@" (mention), EOL (End Of Line) and RT (retweet) are removed. Then, the Arabic stopwords are eliminated with the help of the Python NLTK (Natural Language Toolkit) library. The purpose of normalization is to convert every letter variant to a single form (viz., Ï !i). This process is applied to unify different letter formats, such as ya, alif and waw in Arabic. The speech effects and the single letters are repeatedly reiterated. For example, 'Hellooo' is one of the common features that appear in social media posts, which is mainly intended for emphasizing a conversation. Therefore, if such terms are removed, it produces a single form of the words, i.e., 'Hellooo' and 'Hellooooo' and so on, are mapped altogether against the word, 'Hello'.

TF-IDF is a weighting factor that reflects the significance of a word in a corpus document. It measures a certain word's overall number of occurrences in the provided document and is termed 'Term Frequency'. Nonetheless, certain words, such as the stop-words, commonly emerge in every single document; such content should be discounted in a systematic manner. This procedure is named ‘Inverse Document Frequency’. The lesser the value is, as a sign of differentiating a specific document, the higher the amount of documents will be, in which the word appears. Then, the TF-IDF approach is applied as a part of the Scikit-learn library and is evaluated as given below [20].

$$w_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i}\right), \tag{1}$$

Now, the number of times a word *i* occurs in the document *j* is represented by $\mathrm{tf}_{\mathrm{ij}}$, the number of documents that has the word *i* is denoted by df_{i}, and N refers to overall number of documents.

2.2 Opining Mining Using PNN Model

For classification, the pre-processed data is fed into the PNN model to recognize the class labels. The PNN method is the most commonly-utilized data mining approach and is employed in several pattern recognition and classification problems [21]. In these kinds of Neural networks (NN), the operation is pre-arranged in a multi-layered network that contains four layers such as the output layer, input layer, pattern layer and summation layer. In the input layer, a dimension (*p*) of the input vector reflects the dimension of a layer. In the pattern layer, the dimension of the number of instances in the training subset is equivalent to the dimension of a layer. The 3rd layer (summation) contains several classes in the group. The 4th layer (output) and the validation examples are categorized based on the number of classes. Fig. 1 depicts the infrastructure of the PNN technique.

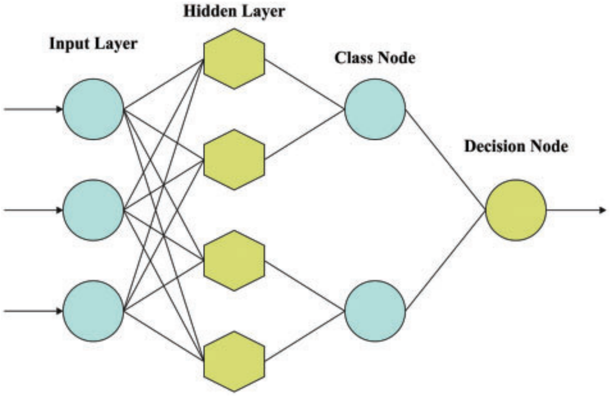


Figure 1: Architecture of the PNN approach

The operation formula of the PNN method includes all four important layers.

In the input layer, all the neurons have a prediction parameter. Its values are fed into every neuron in the hidden layer.

In the pattern layer, for all the training samples, a single layer formulates the product associated with an input vector x by involving the weight of the vector, $w_i, z_i = x \cdot w_i^T$. Then, the succeeding non-linear process is performed as given below.

$$\exp\left[\left(\frac{(-w_i - x).(w_i - x)^T}{(2\alpha^2)}\right)\right], \tag{2}$$

In Eq. (2), *i* denotes a pattern number, T indicates the overall number of the training patterns, X shows the *i-th* training pattern from a class, and *a* refers to a smoothing variable.

The summation layer aggregates the enhanced values of all the input classes and produces the network output as a probability vector (Eq. (3)):

$$\sum_i \exp\left[\left(\frac{(-w_i - x).(w_i - x)^T}{(2\alpha^2)}\right)\right]. \tag{3}$$

The output layer produces different classes that depend on the decision classes, such as Ω_r and Ω_σ. Here, $r \neq s$, $r, s = 1, 2, ..., q$ and a classification condition is shown in Eq. (4).

$$\sum_i \exp\left[\left(\frac{(-w_i - x).(w_i - x)^T}{(2\alpha^2)}\right)\right] > \sum_j \exp\left[\left(\frac{(-w_j - x).(w_j - x)^T}{(2\alpha^2)}\right)\right] \tag{4}$$

This node possesses a single weight C to denote the probability of the preceding membership and is composed of a number of trained instances under each class C. Further, it is represented as the cost parameter as given herewith.

$$C = -\left(\frac{h_s l_s}{h_r l_r} . \frac{n_r}{n_s}\right), \tag{5}$$

In Eq. (5), h_s indicates the previous prospect in which the present sample proceeds to a group, n whereas c_n represents the misclassified cost.

2.3 Parameter Optimization Using PO

At last, the PO algorithm is employed for fine-tuning the parameters involved in the model to enhance the outcomes. The PO algorithm is an intelligent optimization technique developed based on political election procedures followed by human society [22]. In the PO algorithm, every party member is regarded as a solution candidate. The election performance of the party members is regarded as the assessment function. The PO algorithm searches for an optimum solution with the help of a multiple-phase iteration technique. Initially, the whole population encompassing n^2 individuals is separated into n parties. There exists an n member, i.e., a solution candidate in all the parties. Furthermore, each party member acts as an election candidate, viz., one member from every party is carefully chosen to form a constituency. The blue-dotted line specifies the partition of a constituency, whereas the red-dotted line shows the partition of a political party. When mapping the population partition to align with the arithmetic modelling, the whole population is separated into n political parties, and each party contains n party members as follows.

$$P = \{P_1, P_2, P_3, \ldots, P_n\} \tag{6}$$

$$P_i = \{p_i^1, p_i^2, p_i^3, \ldots, p_i^n\} \tag{7}$$

Here, every party member acts as an election candidate. As a result, the whole population is viewed as n constituency, as shown in the following equations. Also, a party member, i.e., a member of the constituency, should be highlighted, though the logical division is dissimilar. The membership of every constituency is classified as given below.

$$C = \{C_1, C_2, C_3, \ldots, C_n\} \tag{8}$$

$$C_j = \{p_1^j, p_2^j, p_3^j, \ldots, p_n^j\} \tag{9}$$

Besides, the leader of the *i-th* party, after calculating the fitness of every member, is distinguished as p_i^*, and the group of every party leader is characterized by P^* as demonstrated in Eq. (10). Likewise, after the election process is over, C^* represents the winner from every constituency and is called as a 'parliamentarian' as illustrated in Eq. (11). Here, c_j^* represents the winner of the *j-th* constituency.

$$P^* = \{p_1^*, p_2^*, p_3^*, \ldots, p_n^*\} \tag{10}$$

$$C^* = \{c_1^*, c_2^*, c_3^*, \ldots, C_n^*\} \tag{11}$$

Campaigning is a fundamental phase in the election process and is accountable for the updated position of the search agent. In this work, a particular presentation is that a party member changes their position based on their leader P^* of the party and the winner C^* of the constituency. Furthermore, as discussed below, they acquire knowledge about the previous election over the original position-updating method. The major concept of the RPPUS method is to forecast a favourable area over the mathematical relationships among the subgroups using an optimum solution, i.e., constituency winner or party leader, with the existing and earlier fitness values of the searching agent.

$$p_{i,k}^j(t+1) = \begin{cases} m^* + r(m^* - p_{i,k}^j(t)), & if\ p_{i,k}^j(t-1) \le p_{i,k}^j(t) \le m^* or\ p_{i,k}^j(t-1) \ge p_{i,k}^j(t) \ge m^* \\ m^* + (2r-1)|m^* - p_{i,k}^j(t)|, & if\ p_{i,k}^j(t-1) \le m^* \le p_{i,k}^j(t)\ or\ p_{i,k}^j(t-1) \ge m^* \ge p_{i,k}^j(t) \\ m^* + (2r-1)|m^* - p_{i,k}^j(t-1)|, & if\ m^* \le p_{i,k}^j(t-1) \le p_{i,k}^j(t)\ or\ m^* \ge p_{i,k}^j(t-1) \ge p_{i,k}^j(t) \end{cases} \tag{12}$$

$$p_{i,k}^j(t+1) = \begin{cases} m^* + (2r-1)\,|m^* - p_{i,k}^j(t)\,|, & if\ p_{i,k}^j(t-1) \le p_{i,k}^j(t) \le m^* or\ p_{i,k}^j(t-1) \ge p_{i,k}^j(t) \ge m^* \\ p_{i,k}^j(t-1) + r\left(p_{i,k}^j(t) - p_{i,k}^j(t-1)\right), & if\ p_{i,k}^j(t-1) \le m^* \le p_{i,k}^j(t)\ or\ p_{i,k}^j(t-1) \ge m^* \ge p_{i,k}^j(t) \\ m^* + (2r-1)\,|m^* - p_{i,k}^j(t-1)\,|, & if\ m^* \le p_{i,k}^j(t-1) \le p_{i,k}^j(t)\ or\ m^* \ge p_{i,k}^j(t-1) \ge p_{i,k}^j(t) \end{cases} \tag{13}$$

In this expression, m^* shows the party's leader or the constituency's winner, r signifies an arbitrary number in the range of [0,1], and T characterizes the existing iteration count. Fig. 2 demonstrates the steps involved in the PO technique.

The party-switching stage is executed to create a balance between the exploitation and the exploration phases. It presents an adaptive variable λ termed a 'party switching rate'. Every party member tends to switch and gets selected into an arbitrarily-chosen party. The switching probability is represented by λ, which is primarily 1. It gets reduced linearly to 0 as follows.

$$\lambda = \left(1 - \frac{t}{T}\right) * \lambda_{\max} \tag{14}$$

Here, the fitness of every solution candidate is described, whereas the party leader and the constituency winner are upgraded.

$$q = \arg\min_{1 \le j \le n} f\left(P_i^j\right) p_i^* = P_i^q \tag{15}$$

$$q = \arg\min_{1 \le i \le n} f\left(P_i^j\right) c_j^* = p_q^j \tag{16}$$

The party-switching stage aims to alter a party's perception, whereas the parliamentary affair stage is the modification of a constituency's perspectives. A constituency winner interacts with others to enhance their fitness values. Every constituency winner uses the following formula to update their location based on the rest of the arbitrarily-designated constituencies. It is to be noted that the movement is employed when the fitness of c_j^* becomes highly efficient.

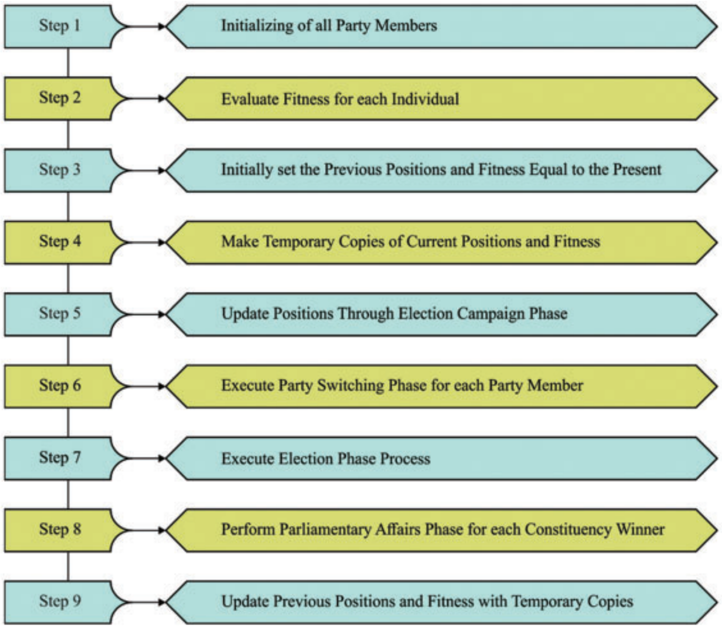

Step 1
Initializing of all Party Members
Step 2
Evaluate Fitness for each Individual
Step 3
Initially set the Previous Positions and Fitness Equal to the Present
Step 4
Make Temporary Copies of Current Positions and Fitness
Step 5
Update Positions Through Election Campaign Phase
Step 6
Execute Party Switching Phase for each Party Member
Step 7
Execute Election Phase Process
Step 8
Perform Parliamentary Affairs Phase for each Constituency Winner
Step 9
Update Previous Positions and Fitness with Temporary Copies

Figure 2: Steps involved in the PO algorithm

Algorithm 1: Pseudocode of the PO algorithm

```
Input: $n$ (amount of constituencies, party members, and political parties), $\lambda_{\max}$ (upper limit of party switching rates), $T_{\max}$ (the entire amount of iterations)
Output: final population $\mathcal{P}(T_{\max})$
Initializing $(n * n)$ candidate members $P$
Compute the fitness of all the members $p_i^j$
Compute the group of party leaders $P^*$ and the group of constituencies winners $C^*$ by utilizing in Eq. (8)
$t = 1$;
$P(t - 1) = P$;
$F(P(t - 1)) = f(P)$ ;
$\lambda = \lambda_{\max}$;
while $t \le T_{\max}$ do
    $P_{temp} = P$;
    $f(P_{temp}) = f(P)$
    for each $P_i \in P$, do
        for each $p_i^j \in P_i$ do
            $p_i^j$ = ElectionCampaign $(p_i^j, p_i^j(t - 1), p_i^j c_j^*)$ ;
            end
        end
PartySwitching $(P, \lambda)$ ;
Compute the fitness of all the members $p_i^j$
```

(Continued)

Algorithm 1: Continued

```
Compute the group of party leaders $P^*$ and the group of constituency winners $C^*$, by utilizing Eq. (8)
Parliamentary Affairs $(C^*, P)$ ;
$P(t-1) = P_{temp}$;
$F(P(t-1)) = f(P_{temp})$ ;
$\lambda = (\lambda - \lambda_{\max}/T_{\max})$ ;
$t = t + 1$;
end
```

3 Results and Discussion

The current chapter assesses the proposed POPNN-COM model's performance using two datasets, Corpus and Corpus$^+$. The details of the datasets are given in Table 1. For experimental validation, ten-fold cross-validation was executed.

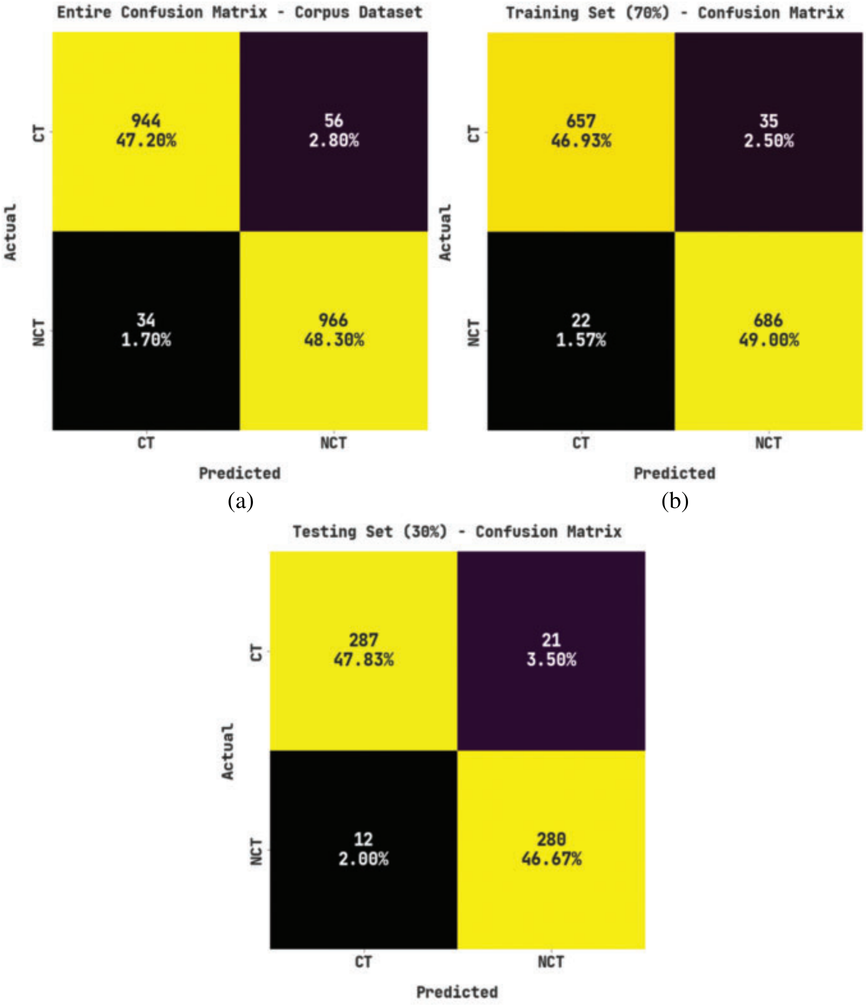
Table 1: Dataset details

Label	Description	No. of samples	
		Corpus	Corpus$^+$
CT	Comparative text (CT)	1000	1200
NCT	Non-comparative text (NCT)	1000	1200
Total no. of samples		2000	2400

Fig. 3 demonstrates the confusion matrices generated by the POPNN-COM model on the Corpus dataset using distinct Training (TR) and Testing (TS) datasets. The figure indicates that the proposed POPNN-COM model properly recognized both CT and the NCT samples in all aspects. Table 2 and Fig. 4 portray the CT and the NCT classification outcomes achieved by the proposed POPNN-COM model on the Corpus dataset. Upon the entire dataset, the proposed POPNN-COM model produced average $accu_y$, $prec_n$, $reca_l$, F_{score} and AUC_{score} values such as 95.50%, 95.52%, 95.50%, 95.50% and 95.50%, respectively. Simultaneously, upon 70% of the TR data, the presented POPNN-COM model achieved average $accu_y$, $prec_n$, $reca_l$, F_{score} and AUC_{score} values such as 95.93%, 95.95%, 95.92%, 95.93% and 95.92% correspondingly. Concurrently, on 30% of the TS data, the POPNN-COM approach resulted in average $accu_y$, $prec_n$, $reca_l$, F_{score} and AUC_{score} values such as 94.50%, 94.50%, 94.54%, 94.50% and 94.54% correspondingly.

Both Training Accuracy (TA) and Validation Accuracy (VA) values attained by the proposed POPNN-COM method on Corpus dataset are illustrated in Fig. 5. The experimental outcomes denote that the proposed POPNN-COM technique achieved the maximal TA and VA values. In contrast, the VA values were higher than the TA values.

Both Training Loss (TL) and Validation Loss (VL) values, achieved by the proposed POPNN-COM approach on Corpus dataset, are shown in Fig. 6. The experimental outcomes imply that the presented POPNN-COM algorithm accomplished the least TL and VL values. In contrast, the VL values were lesser than the TL values.



(a) (b)

(c)

Figure 3: Confusion matrices of the POPNN-COM approach on Corpus dataset (a) Entire dataset, (b) 70% of the TR data and (c) 30% of the TS data

Table 2: Analytical results of the POPNN-COM approach on the Corpus dataset under different measures

Labels	Accuracy	Precision	Recall	F-score	AUC score
Entire dataset					
CT	95.50	96.52	94.40	95.45	95.50
NCT	95.50	94.52	96.60	95.55	95.50
Average	95.50	95.52	95.50	95.50	95.50

(Continued)

Table 2: Continued

Labels	Accuracy	Precision	Recall	F-score	AUC score
Training set (70%)					
CT	95.93	96.76	94.94	95.84	95.92
NCT	95.93	95.15	96.89	96.01	95.92
Average	95.93	95.95	95.92	95.93	95.92
Testing set (30%)					
CT	94.50	95.99	93.18	94.56	94.54
NCT	94.50	93.02	95.89	94.44	94.54
Average	94.50	94.50	94.54	94.50	94.54

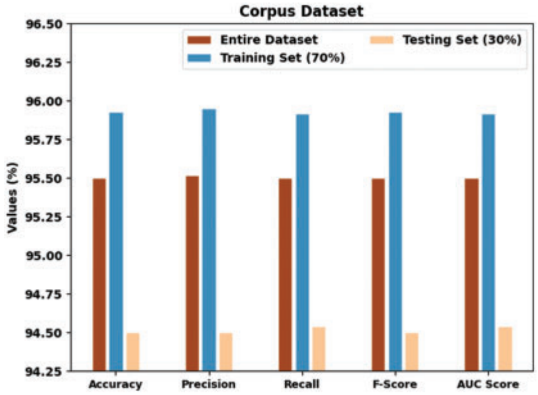


Figure 4: Analytical results of the POPNN-COM approach on the Corpus dataset

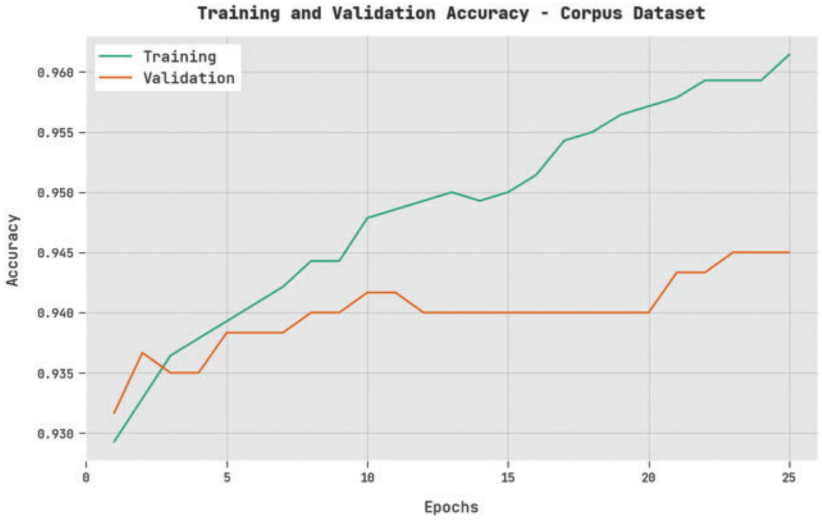


Figure 5: TA and VA analyses results of the POPNN-COM approach on the Corpus dataset

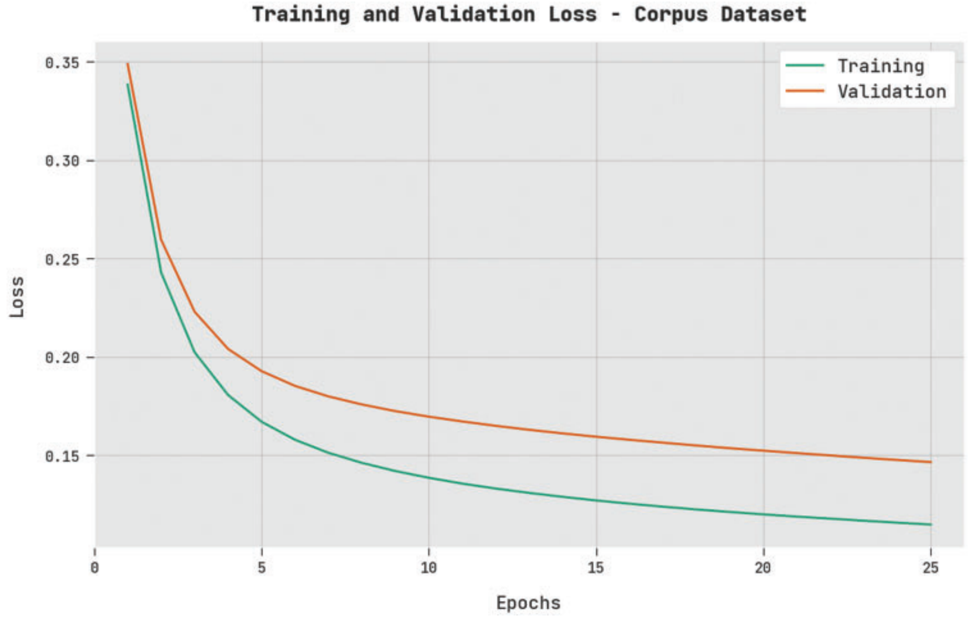


Figure 6: TL and VL analyses results of the POPNN-COM approach on Corpus dataset

A clear precision-recall analysis was conducted on the proposed POPNN-COM method using the Corpus dataset, and the results are portrayed in Fig. 7. The figure indicates that the proposed POPNN-COM method produced the enhanced precision-recall values under all the classes.

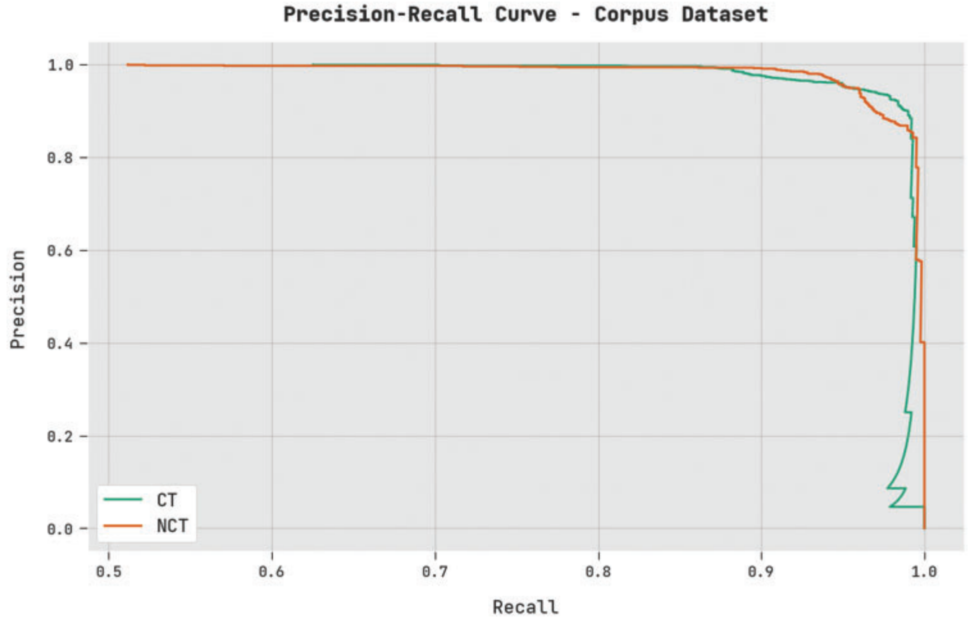
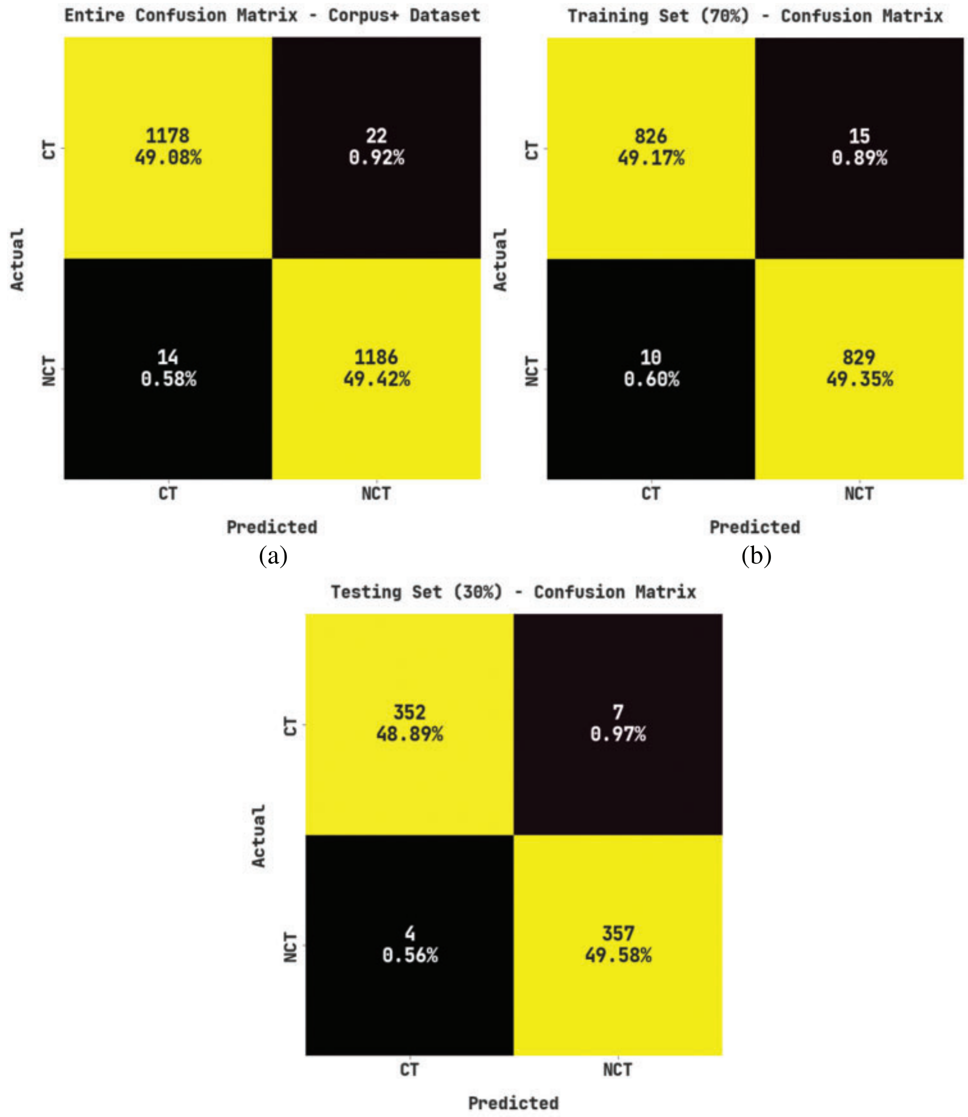


Figure 7: Precision-recall analysis results of the POPNN-COM approach on the Corpus dataset

Fig. 8 establishes the confusion matrices created by the proposed POPNN-COM model on Corpus$^+$ dataset. The figure represents that the proposed POPNN-COM model properly recognized both CT and the NCT samples in all aspects.



(a) (b) (c)

Figure 8: Confusion matrices of the POPNN-COM approach on Corpus$^+$ dataset (a) Entire dataset, (b) 70% of the TR data and (c) 30% of the TS data

Table 3 and Fig. 9 signify the CT and the NCT classification outcomes achieved by the proposed POPNN-COM model on Corpus$^+$ dataset. Upon the entire dataset, the proposed POPNN-COM model produced average $accu_y$, $prec_n$, $reca_l$, F_{score} and AUC_{score} values such as 98.50%, 98.50%, 98.50%, 98.50% and 98.50%, respectively. At the same time, on 70% of the TR data, the presented POPNN-COM model achieved average $accu_y$, $prec_n$, $reca_l$, F_{score} and AUC_{score} values such as 98.51%, 98.51%, 98.51%, 98.51% and 98.51% correspondingly. In parallel, on 30% of the TS data, the proposed

POPNN-COM model attained average $accu_y$, $prec_n$, $reca_l$, F_{score} and AUC_{score} values such as 98.47%, 98.48%, 98.47%, 98.47% and 98.47% correspondingly.

Table 3: Analytical results of the POPNN-COM approach on the Corpus$^+$ dataset under different measures

Labels	Accuracy	Precision	Recall	F-score	AUC score
Entire dataset					
CT	98.50	98.83	98.17	98.49	98.50
NCT	98.50	98.18	98.83	98.50	98.50
Average	98.50	98.50	98.50	98.50	98.50
Training set (70%)					
CT	98.51	98.80	98.22	98.51	98.51
NCT	98.51	98.22	98.81	98.51	98.51
Average	98.51	98.51	98.51	98.51	98.51
Testing set (30%)					
CT	98.47	98.88	98.05	98.46	98.47
NCT	98.47	98.08	98.89	98.48	98.47
Average	98.47	98.48	98.47	98.47	98.47

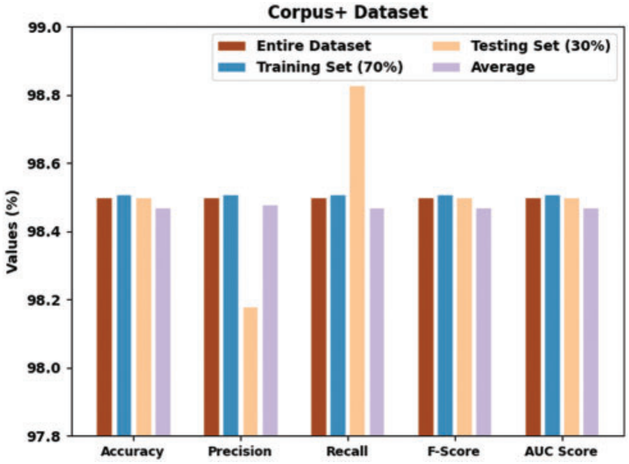


Figure 9: Analytical results of the POPNN-COM approach on the Corpus$^+$ dataset

Both TA and VA values, obtained by the proposed POPNN-COM method on the Corpus$^+$ dataset, are illustrated in Fig. 10. The experimental outcomes portray that the proposed POPNN-COM technique achieved the maximal TA and VA values. In contrast, the VA values were higher than the TA values.

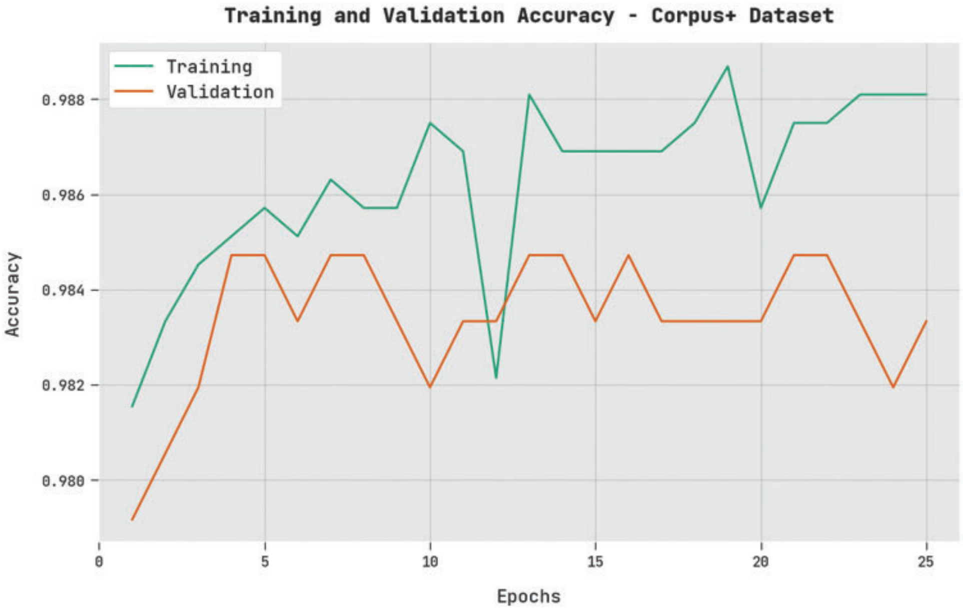


Figure 10: TA and VA analyses results of the POPNN-COM approach on the Corpus$^+$ dataset

Both TL and VL values, achieved by the presented POPNN-COM approach on the Corpus$^+$ dataset, are established in Fig. 11. The experimental outcomes denote that the proposed POPNN-COM algorithm accomplished the least TL and VL values. In contrast, the VL values were lower than the TL values.

Table 4 showcases the comparative analytical results achieved by the proposed POPNN-COM technique and other existing techniques on the Corpus$^+$ dataset. With respect to $accu_y$, the proposed POPNN-COM model achieved a maximum $accu_y$ of 94.50%, whereas the Naïve Bayes (NB), Jrip, J48, POS, KC and the POS-KC models accomplished the least $accu_y$ values such as 80.97%, 88.36%, 89.96%, 87.04%, 90.91% and 90.47% respectively. Also, with respect to $prec_n$, the POPNN-COM model accomplished a maximum $prec_n$ of 94.50%, whereas the NB method, Jrip, J48, POS, KC and the POS-KC models accomplished the least $prec_n$ values such as 74.74%, 91.09%, 92.97%, 92.78%, 98.98% and 92.73% correspondingly. Followed by, with respect to $reca_l$, the proposed POPNN-COM model attained a maximum $reca_l$ of 94.54%, whereas the NB, Jrip, J48, POS, KC and the POS-KC model achieved the least $reca_l$ values such as 84.72%, 77.12%, 78.60%, 75.48%, 83.14% and 84.16% respectively. With respect to F_{score}, the proposed POPNN-COM model achieved a maximum F_{score} of 94.50%, whereas the NB method, Jrip, J48, POS, KC and the POS-KC model accomplished the least F_{score} values such as 79.53%, 85.47%, 86.98%, 83.40%, 88.66% and 88.31% correspondingly.

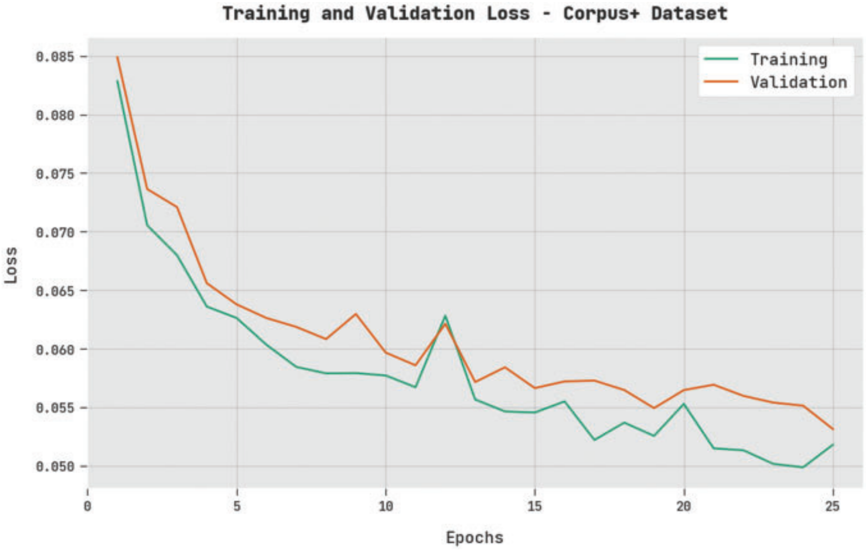


Figure 11: TL and VL analyses results of the POPNN-COM approach on the Corpus$^+$ dataset

Table 4: Comparative analysis results of the POPNN-COM approach and other recent algorithms on the Corpus$^+$ dataset

Corpus$^+$ dataset				
Methods	Accuracy	Precision	Recall	F-score
POPNN-COM	94.50	94.50	94.54	94.50
Naïve Bayes classifier	80.97	74.74	84.72	79.53
Jrip classifier	88.36	91.09	77.12	85.47
J48 classifier	89.96	92.97	78.60	86.98
POS classifier	87.04	92.78	75.48	83.40
KC algorithm	90.91	92.98	83.14	88.66
POS-KC	90.47	92.73	84.16	88.31

Table 5 portrays the comparison study outcomes of the proposed POPNN-COM technique and other existing techniques on the Corpus$^+$ dataset. With respect to $accu_y$, the POPNN-COM model achieved an increased $accu_y$ of 98.47%, whereas the NB model, Jrip, J48, POS, KC and the POS-KC model achieved the least $accu_y$ values such as 80.99%, 89.77%, 90.43%, 86.80%, 91.88% and 91.40% correspondingly. Moreover, with respect to $prec_n$, the proposed POPNN-COM model accomplished a maximum $prec_n$ of 98.48%, whereas the NB approach, Jrip, J48, POS, KC and the POS-KC model accomplished the least $prec_n$ values such as 73.70%, 93.89%, 97.85%, 92.53%, 94.84% and 92.89% correspondingly. Then, with respect to $reca_l$, the proposed POPNN-COM model exhibited an increase in the $reca_l$ value up to 98.47%. However, the NB approach, Jrip, J48, POS, KC and the POS-KC model accomplished the least $reca_l$ values such as 87.33%, 81.54%, 79.33%, 76.19%, 85.82% and 86.06%, correspondingly. With respect to F_{score}, the proposed POPNN-COM model achieved an increased F_{score}

of 98.47%. However, the NB approach, Jrip, J48, POS, KC and the POS-KC model accomplished the least F_{score} values such as 79.83%, 87.01%, 87.46%, 83.61%, 90.33% and 89.47%, correspondingly.

Table 5: Comparative analysis results of the POPNN-COM approach and other existing algorithms on Corpus$^+$ dataset

Corpus$^+$ dataset				
Methods	Accuracy	Precision	Recall	F-score
POPNN-COM	98.47	98.48	98.47	98.47
Naïve bayes classifier	80.99	73.70	87.33	79.83
Jrip classifier	89.77	93.89	81.54	87.01
J48 classifier	90.43	97.85	79.33	87.46
POS classifier	86.80	92.53	76.19	83.61
KC algorithm	91.88	94.84	85.82	90.33
POS-KC	91.40	92.89	86.06	89.47

4 Conclusion

In the current study, a novel POPNN-COM model has been developed for opinion mining from the Arabic text. The proposed POPNN-COM model aims to recognize comparative and non-comparative texts in Arabic social media content. Initially, the POPNN-COM model involves different levels of data pre-processing to transform the input data into a useful format. For classification, the pre-processed data is fed into the PNN model to recognize the class labels. At last, the PO algorithm is employed for fine-tuning the parameters involved in the model to enhance the results. The proposed POPNN-COM model was experimentally validated using two datasets, and the outcomes establish the promising performance of the presented POPNN-COM algorithm over other recent approaches. Thus, the POPNN-COM model has been established as an effective tool for comparative opinion mining. It can be used in the future for real-time applications. Further, future studies can also extend the presented model for hate speech detection.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2022R263), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4310373DSR56.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] A. Elnagar, R. Al-Debsi and O. Einea, “Arabic text classification using deep learning models,” *Information Processing & Management*, vol. 57, no. 1, pp. 102121, 2020.

[2] S. L. M. Sainte and N. Alalyani, “Firefly algorithm based feature selection for arabic text classification,” *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 3, pp. 320–328, 2020.

[3] A. M. F. Al-Sbou, “A survey of Arabic text classification models,” *International Journal of Informatics and Communication Technology*, vol. 8, no. 1, pp. 25–28, 2018.

[4] F. N. Al-Wesabi, “Entropy-based watermarking approach for sensitive tamper detection of arabic text,” *Computers Materials & Continua*, vol. 67, no. 3, pp. 3635–3648, 2021.

[5] A. Wahdan, S. A. L. Hantoobi, S. A. Salloum and K. Shaalan, “A systematic review of text classification research based on deep learning models in Arabic language,” *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 6629, 2020.

[6] F. N. Al-Wesabi, “A hybrid intelligent approach for content authentication and tampering detection of arabic text transmitted via internet,” *Computers Materials & Continua*, vol. 66, no. 1, pp. 195–211, 2021.

[7] M. A. Ahmed, R. A. Hasan, A. H. Ali and M. A. Mohammed, “The classification of the modern arabic poetry using machine learning,” *TELKOMNIKA*, vol. 17, no. 5, pp. 2667, 2019.

[8] F. N. Al-Wesabi, “A smart english text zero-watermarking approach based on third-level order and word mechanism of Markov model,” *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1137–1156, 2020.

[9] A. El Kah and I. Zeroual, “The effects of pre-processing techniques on Arabic text classification,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 1, pp. 41–48, 2021.

[10] F. N. Al-Wesabi, A. Abdelmaboud, A. A. Zain, M. M. Almazah and A. Zahary, “Tampering detection approach of arabic-text based on contents interrelationship,” *Intelligent Automation & Soft Computing*, vol. 27, no. 2, pp. 483–498, 2021.

[11] H. El Rifai, L. Al Qadi and A. Elnagar, “Arabic text classification: The need for multi-labeling systems,” *Neural Computing and Applications*, vol. 34, no. 2, pp. 1135–1159, 2022.

[12] F. S. Al-Anzi and D. AbuZeina, “Beyond vector space model for hierarchical Arabic text classification: A Markov chain approach,” *Information Processing & Management*, vol. 54, no. 1, pp. 105–115, 2018.

[13] A. Adel, N. Omar, M. Albared and A. Al-Shabi, “Feature selection method based on statistics of compound words for arabic text classification,” *The International Arab Journal of Information Technology*, vol. 16, no. 2, pp. 178–185, 2019.

[14] F. A. Abdulghani and N. A. Z. Abdullah, “A survey on arabic text classification using deep and machine learning algorithms,” *Iraqi Journal of Science*, vol. 63, no. 1, pp. 409–419, 2022.

[15] N. Aljedani, R. Alotaibi and M. Taileb, “HMATC: Hierarchical multi-label Arabic text classification model using machine learning,” *Egyptian Informatics Journal*, vol. 22, no. 3, pp. 225–237, 2021.

[16] S. Boukil, M. Biniz, F. Adnani, L. Cherrat and A. Moutaouakkil, “Arabic text classification using deep learning technics,” *International Journal of Grid and Distributed Computing*, vol. 11, no. 9, pp. 103–114, 2018.

[17] S. Bahassine, A. Madani, M. Al-Sarem and M. Kissi, “Feature selection using an improved Chi-square for Arabic text classification,” *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp. 225–231, 2020.

[18] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah *et al.,* “Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification,” *Neural Computing and Applications*, vol. 32, no. 16, pp. 12201–12220, 2020.

[19] A. Y. Muaad, M. A. Al-antari, S. Lee and H. J. Davanagere, “A novel deep learning arcar system for arabic text recognition with character-level representation,” in *Proc. of The 1st Int. Electronic Conf. on Algorithms*, pp. 14, 2021.

[20] M. C. Manning, D. K. Chou, B. M. Murphy, R. W. Payne and D. S. Katayama, “Stability of protein pharmaceuticals: An update,” *Pharmaceutical Research*, vol. 27, no. 4, pp. 544–575, 2010.

[21] G. Capizzi, G. L. Sciuto, C. Napoli, D. Polap and M. Wozniak, “Small lung nodules detection based on fuzzy-logic and probabilistic neural network with bioinspired reinforcement learning,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 6, pp. 1178–1189, 2020.

[22] Q. Askari and I. Younas, “Political optimizer based feedforward neural network for classification and function approximation,” *Neural Processing Letters*, vol. 53, no. 1, pp. 429–458, 2021.