

Optimizing Sentiment Analysis: A Novel Hybrid Model Integrating PCC-HHO with BiLSTM-RNN for Enhanced Accuracy on Diverse Textual Datasets

Irfan Qutab
School of Software
Northwestern Polytechnical University
Xi'an, China

Zara Asghar
Department of Computer Science
University of Lahore
Sargodha, Pakistan

Muhammad Aqeel
School of Software
Northwestern Polytechnical University
Xi'an, China

Unaiza Fatima
School of Software
Northwestern Polytechnical University
Xi'an, China

Wahab Naqvi
School of Control Science and Engineering
Northwestern Polytechnical University
Xi'an, China

Muhammad Yasir Muneeb
School of Material Science and Processing Engineering
Northwestern Polytechnical University
Xi'an, China

Abstract— In today's digital age, the volume of textual content produced across platforms like messaging services (e.g., Telegram, WhatsApp), social media sites (e.g., Instagram, Facebook), and e-commerce platforms (e.g., Amazon) is rapidly increasing. Analyzing this data offers valuable insights into consumer sentiments, helping business owners understand public perceptions of their products, brands, or services to make informed decisions. This research presents a novel hybrid model for sentiment analysis that incorporates both Natural Language Processing (NLP) techniques and existing machine learning models. The proposed approach involves pre-processing, feature extraction, feature selection, and sentiment classification stages. In the pre-processing phase, unnecessary information is removed from the input text using NLP techniques. For feature extraction, a hybrid method combining TF-IDF, Skip N-gram models, and a feature selection process using Pearson Correlation Coefficient and Harris Hawks Optimization is used to construct a distinctive feature vector for each review. Sentiment classification is performed using a Bi-LSTM-RNN model, known for capturing contextual dependencies in the data. Additionally, the performance of traditional models such as SVM, Logistic Regression, and LSTM is compared. The proposed model was evaluated on two prominent datasets: the SemEval-2014 restaurant reviews dataset and the Sentiment140 Twitter dataset. Our hybrid model outperformed existing models, achieving state-of-the-art results with an accuracy of 96.54% on the SemEval-2014 dataset and 97.81% on the Sentiment140 dataset, demonstrating its effectiveness in accurately classifying sentiments in diverse textual data.

Keywords— Sentiment Analysis, NLP, Pearson Correlation Coefficient, Harris Hawks Optimization, Bi-LSTM.

I. INTRODUCTION

With the internet's evolution, social media platforms have surged in popularity as spaces for individuals to disseminate and exchange views about everyday experiences [1]. User-generated content, particularly reviews across various online applications, has increasingly reflected vivid emotional and opinionated sentiments over time [2]. This trove of review content holds significant potential for both consumers and businesses, offering insights into the quality of products through customer feedback [3]. Yet, the sheer volume of reviews available online makes it a daunting task to manually sift through and distill actionable insights from consumer feedback. Consequently, there's an urgent requirement for methodologies that can translate this information into a format that machines can process efficiently. Sentiment analysis emerges as an effective tool in addressing these challenges by providing computer-understandable interpretations of user sentiments [4]. Text mining is the process of uncovering meaningful and captivating insights from raw, unstructured text. The procedure commences with data pre-processing, advances to the extraction of pertinent attributes from the cleansed data, and culminates in the assessment of sentiment polarity based on these attributes through advanced deep learning (DL) techniques [5]. As a subset of machine learning (ML), deep learning employs a variety of neural network algorithms for the identification of patterns and the resolution of classification and regression quandaries. It characteristically involves multiple layers that are adept at deciphering complicated data representations, discerning high-

level features, and accurately categorizing or quantifying data characteristics. In sentiment analysis (SA) endeavors, DL has significantly elevated its proficiency. The operational prowess of specialized SA frameworks is notably augmented by deploying DL models that excel in feature extraction and discernment [6] [7]. Sentiment analysis data reveals that individuals often exhibit unique perspectives and emotional tendencies towards the same entity. Acknowledging this diversity is key to providing powerful tools for marketing and competitive analysis. Sentiment Analysis (SA) has become widely adopted across various industries, such as film recommendation services, online retail platforms, and surveys of public sentiment [8]. To enhance the accuracy of perceptron neural networks, certain optimization techniques, which include meta-heuristic algorithms, have been applied [9]. For instance, analyzing the sentiment in user reviews of products on an e-commerce site can yield valuable insights for developing models that map the relationship between users and products [10]. On the e-commerce platform, consumers have the advantage of obtaining tailored recommendations, while vendors have the ability to swiftly modify their offerings in response to insights from the model that maps the interactions between users and products. The proposed approach encompasses several key stages: pre-processing, feature extraction, feature selection, and sentiment classification. Initially, irrelevant data is removed from the input text reviews during the pre-processing phase using NLP techniques. For feature extraction, a hybrid approach is employed, combining TF-IDF and Skip N-gram models with a feature selection process that utilizes Pearson Correlation Coefficient (PCC) and Harris Hawks Optimization (HHO) to create a unique feature vector for each review. Finally, sentiment classification is carried out using a Bi-LSTM-RNN model, which effectively captures the contextual relationships within the data.

This section provides an overview of sentiment analysis and introduces the current study. Section 2 explores the related literature. In Section 3, the processes for building the models are explained, covering steps such as pre-processing, feature extraction, feature selection, and sentiment classification. The outcomes of the model tests are discussed in Section 4. Lastly, the conclusion is presented in the final section.

II. LITERATURE REVIEW

Alzubi et al. [11] introduced the Consensus-based Combining Method (CCM), a strategy for fusing multiple classifiers by aggregating their weighted outcomes. What sets CCM apart is its initial comparison of classifier outputs, which helps adjust their weights before reaching a final consensus. Onan et al. [12] proposed a deep learning-based sentiment analysis model for Twitter posts, using three word embedding techniques (word2vec, fast-Text and GloVe) with a convolutional neural network. The study tested this method on Twitter subsets ranging from 5,000 to 50,000 messages. In another study, the authors of [13] developed a fuzzy logic-based recommendation system to predict online shopping items in real-time by analyzing product reviews and applying demographic filtering, sentiment analysis, and ontology matching to offer personalized product suggestions. Zhao et al. [14] found that user-generated content, like reviews and ratings, plays a crucial role in shaping consumer

preferences within social recommendation systems. They highlighted that incorporating users' internal effects could further improve prediction accuracy. The authors applied Multinomial Logistic Regression to classify COVID-19-related tweets into positive, negative, and neutral sentiments. The model achieves 95.14% accuracy using Count-Vectorizer and Tf-idf for feature extraction, providing insights into public sentiment during the pandemic [15]. Jain et al. [16] conducted a comprehensive study on Twitter sentiment analysis using Machine Learning (ML) techniques, which involved collecting tweets, refining them with Natural Language Processing (NLP), and identifying key sentiment indicators through feature extraction. The sentiment models were validated using classifiers such as Support Vector Machines, Naive Bayes, and Decision Trees. Bose and colleagues [17] focused on analyzing food-related customer feedback on Amazon, categorizing emotions and sentiments to understand consumer behavior better and reduce business risks. Zhang and team [18] developed a semi-automated sentiment annotation platform using a BERT-based classification system, enhancing the robustness and accuracy of sentiment analysis, though they acknowledged the challenges in Aspect-Based Sentiment Analysis (ABSA) due to limited data. A sentiment classification model specifically designed for Roman Urdu text was introduced, categorizing text into four distinct groups: politics, sports, education, and religion [19]. The sentiment analysis of Roman Urdu text was explored and assessed, with the development of various systems aimed at addressing Roman Urdu sentiment classification. Additionally, a comparison of different researchers' outcomes in this domain was conducted [20].

Mai et al. [21] developed a framework to collect and analyze YouTube comments related to specific products, using a classification system to filter out irrelevant comments and then applying sentiment analysis to both specific topics (ASA) and general sentiment (SSA). However, their model operates with a limited dataset, leaving room for improvement in single-domain tasks. Wassan and colleagues [22] introduced a technique focused on product attributes in sentiment analysis, tested using Amazon customer reviews. Their method involved preprocessing steps like stemming, tokenization, and lemmatization to derive insights about positive or negative sentiments from the reviews. Alorini et al. [23] used NLP techniques to extract both positive and negative sentiments from COVID-19-related tweets, employing an LSTM-RNN framework for sentiment prediction with high accuracy. This approach has potential applications in spreading awareness about the virus. Alzubi et al. [24] proposed a method for detecting Android malware by combining a Support Vector Machine (SVM) classifier with the Harris Hawks Optimization (HHO) algorithm. Their method, tested on the CICMalAnal2017 dataset, showed superior performance compared to other metaheuristic algorithms. Similarly, Omar et al. [25] developed a method for the detection of Android malware by applying machine learning, optimizing SVM hyperparameters using the HHO algorithm to improve classification accuracy and feature weighting.

Current approaches face challenges such as subpar performance across different domains, diminished precision and effectiveness in sentiment analysis when working with

poorly labeled data, and the struggle to process complex expressions that require advanced interpretation beyond mere sentiment terminology and elementary analysis. To address these problems, we developed PCC-HHO-Bi-LSTM-RNN model. Our proposed model has achieved highest accuracy.

III. METHODOLOGY

The Proposed methodology of the system is depicted in Figure 1, consists of four-stage process. The procedure begins with data preprocessing, where raw data is converted into a usable format. Subsequent to this, algorithms dedicated to feature extraction are employed to extract features from user reviews.

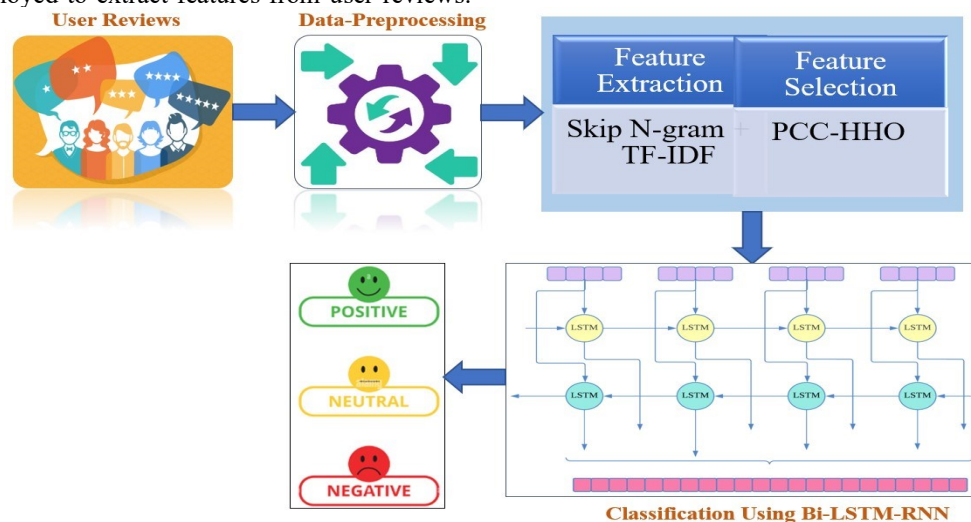


Figure 1. Methodology for sentiment classification of Comments

A. Dataset

In our study, we have used the Sentiment140 [26] and SemEval-2014 [27] restaurant reviews datasets, both of which are essential resources for sentiment analysis but differ in their data composition and focus. The Sentiment140 dataset consists 1.6 million tweets annotated with sentiment labels, where 0 represents negative, 2 represents neutral, and 4 represents positive sentiments. This dataset is particularly valuable for analyzing informal, short text data typical of Twitter, where language use often includes abbreviations, slang, and emoticons. On the other hand, the SemEval-2014 restaurant reviews dataset consists of 3,000 training comments and 800 testing comments. Each review in this dataset includes at least one annotated aspect term, making it highly suitable for aspect-based sentiment analysis in a structured, domain-specific context. The combination of these datasets allows our model to capture sentiment in both general social media content and specialized reviews, providing a comprehensive evaluation of its performance across different types of textual data.

B. Data Preprocessing

1) Text Preprocessing

- a) Remove URLs: Tweets often contain URLs, which are not useful for most analyses.
- b) Strip HTML tags: Sometimes tweets might contain HTML entities.

The ensuing phase involves feature selection, a critical component in Sentiment Analysis (SA), which involves pinpointing an optimal subset of features that preserves the integrity of the original dataset. This phase essentially entails choosing the most relevant features from an extensive array and evaluating them based on specific benchmarks. The final step utilizes the BiLSTM-RNN for assigning sentiment classifications into appropriate categories. The final classifications produced by this system are tagged with one of the following emotional descriptors: Positive, Neutral, or Negative.

- c) Remove special characters and numbers: Cleanse tweets of any non-lexical characters.
- d) Handle emojis: Decide whether to remove emojis or convert them to text.
- 2) Normalize Case: Convert all text to lower case to ensure uniformity, as text data is case-sensitive.
- 3) Tokenization: Break down the tweets into individual words or tokens, as this allows for easier manipulation and analysis of the data.
- 4) Remove Stop Words: Filter out common words (like 'the', 'a', 'an', 'in') which don't contribute much to the meaning of the tweets.
- 5) Handle Mentions and Hashtags: Decide how to process '@mentions' and '#hashtags'. You might choose to remove them or convert them into standard words.
- 6) Stemming/Lemmatization: Stemming trims the words to their root form and lemmatization converts the words to their base or dictionary form to normalize the verbs and other words.

C. Feature Extraction (TF-IDF + Skip N gram Model)

Feature Extraction (FE) plays an essential role in the data mining workflow. This process involves selecting key feature words from the text to represent the textual information effectively. It also transforms these features from an unstructured format into a structured one. During the text feature extraction stage, any irrelevant or superfluous features

are discarded. Meanwhile, significant elements (such as words, sentences, or characters) are identified and assigned appropriate weights to accurately encapsulate the information contained within the text.

The TF-IDF technique transforms textual data into a significant numerical format, subsequently utilized for training deep learning models in classification tasks. This method serves as a measure to ascertain the importance of a word within a given document D. Here, TF represents the frequency of occurrence of a specific word t within the document D.

$$TF(t, d) = \frac{\text{Frequency of term } t \text{ in document } d}{\text{Total number of terms in document } d}$$

Term Frequency (TF) assesses how often a word or term appears in the dataset. Common words like stop words may occur often but don't necessarily add significant value. Conversely, the Inverse Document Frequency (IDF) method evaluates the importance of a term. This approach places greater emphasis on words that appear infrequently in the text d. The calculation of IDF is conducted using the following method:

$$IDF = \log \frac{\text{Number of documents containing term } t}{\text{Total number of documents}}$$

Term Frequency-Inverse Document Frequency (TF-IDF), operates as a vector space model (VSM). It assesses the relevance of a word in the context of a particular document. For example, in a document containing a total of hundreds of words, if the term "excellent" appears 8 times, its Term Frequency (TF) would be calculated as 8 divided by 100, resulting in 0.08. This method is particularly effective because it evaluates a word's importance within a broader dataset, offering a more insightful analysis than the Bag of Words (BoW) approach.

The Skip N Gram model, particularly noted for its efficacy, operates on the principle of predicting context words given a target word, effectively modeling the probability of a word w_o given a word w_t in its context. The objective function of Skip-Gram, aimed at maximizing the context likelihood, is given by:

$$\max_{(\theta)} = \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j} | w_t; \theta)$$

Here, θ represents the model parameters, T is the length of the text corpus, and c is the size of the training context. The conditional probability $P(w_{t+j} | w_t)$ is modeled using a softmax function:

$$P(w_o | w_t) = \frac{\exp(v'_{w_o} T v_{w_t})}{\sum_{w=1}^W \exp(v'_{w_o} T v_{w_t})}$$

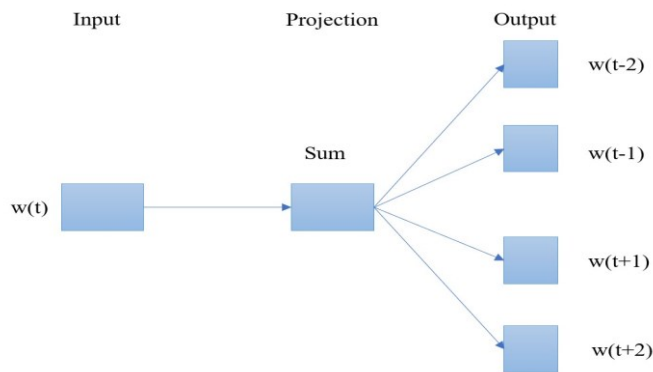


Figure 2. Skip N Gram Model's Structure

In this formulation, v'_{w_o} and v_{w_t} denote the 'input' and 'output' vector representations of words, respectively. The vectors are learned through training over a text corpus, leading to a learned embedding space where geometric relationships encode semantic similarities. The model is illustrated in the figure 2.

D. Feature Selection (PCC-HHO)

We employed a Harris Hawks Optimization (HHO) technique, guided by Pearson Correlation Coefficients (PCC), to extract features from user reviews. Initially, the PCC values of the features are utilized for preliminary dimensionality reduction. Following this, the HHO method is applied to select a concise set of non-repetitive features. This feature selection methodology is depicted in Figure 3.

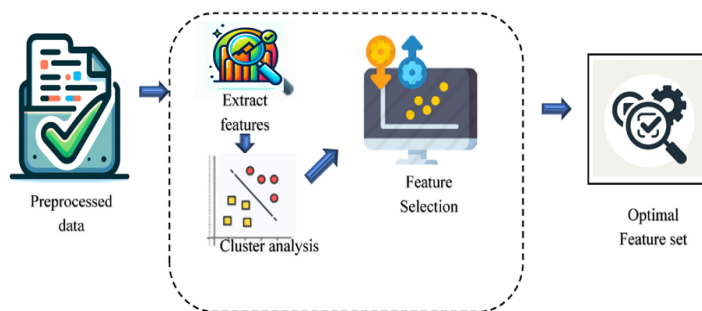


Figure 3. Selecting Features by using PCC-HHO

1) PCC (Person Correlation Coefficient)

The Pearson Correlation Coefficient (PCC) is a quantitative approach used for assessing the relationship between two continuous variables. In this context, it is employed to ascertain the correlation among features.

$$R_{pq} = \frac{\sum_{i=1}^n (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^n (p_i - \bar{p})^2 (q_i - \bar{q})^2}}$$

we calculated the correlation between each pair of features. This method allowed us to quantify the strength and direction of the relationships, identifying positive, negative, or no correlations. The results guided feature selection and dimensionality reduction, which were essential for improving model accuracy and performance. The clustering process was further refined using hierarchical clustering, which grouped similar features based on their correlation values.

2) HHO (Harris Hawks Optimization algorithm)

The Harris Hawks Optimization algorithm was developed by Heidari et al. [28], draws its inspiration from the group hunting tactics of Harris hawks. This algorithm is a nature-based metaheuristic that emulates the way these hawks surround, pursue, and capture their target. It operates in distinct stages: initial exploration, a transitional phase from exploration to exploitation, and finally, exploitation.

- Exploration Phase: In this initial stage, the hawks randomly scout for their target. The mathematical model for a hawk's position H_i at a given iteration t is expressed as:

$$H_i(t + 1) = H_{rabbit} - r \times |H_{rabbit} - 2 \times r \times H_i(t)|$$

Here, H_{rabbit} represents the prey's position (optimal solution so far), and r is a random vector ranging between 0 and 1.

- Transition from Exploration to Exploitation: The algorithm employs a dynamic variable q to fluidly transition between exploration and exploitation, thus ensuring a blend of diversification and intensification in the search process.
- Exploitation Phase: In the exploitation stage, the hawks execute a surprise attack on their prey. This phase is categorized into various strategies like soft besiege, hard besiege, and their rapid dive variations. An example of the soft besiege with rapid dive is formulated as:

$$H_i(t + 1) = \Delta H(t) - \mu \times |J \times H_{rabbit}(t) - H_i(t)|$$

In this equation, $\Delta H(t)$ denotes the average position of the hawks, μ is a random value between 0 and 1, and J signifies the random jump strength. This algorithm's phases collectively contribute to its robustness in identifying optimal solutions.

E. Proposed Algorithm for feature selection using PCC-HHO

Start

Input:

D : Preprocessed textual dataset

T : Target labels

Output:

S : Selected features for sentiment classification

1. Initialize:
2. $X, Y, W, A \leftarrow \emptyset$
3. Extract features F using TF-IDF and Skip N-gram models on D .
4. Compute Pearson Correlation Coefficients (PCC):
5. For each feature f_i in F :
6. Calculate the correlation coefficient R_i for f_i
7. $X \leftarrow X \cup R_i$
8. Feature Selection using Harris Hawks Optimization (HHO):
9. Initialize HHO:
10. Set initial hawk positions H_i
11. Define H_{rabbit} as the best position (optimal solution so far)

12. Exploration Phase:
13. For each hawk H_i :
14. Update $H_i(t + 1)$ based on exploration strategy.
15. Transition to Exploitation:
16. Update q (transition parameter)
17. Adjust hawk positions based on q .
18. Exploitation Phase:
19. For each hawk H_i in H :
20. Apply exploitation strategies (e.g., soft besiege, hard besiege)
21. Select the top features based on the optimized selection.

End

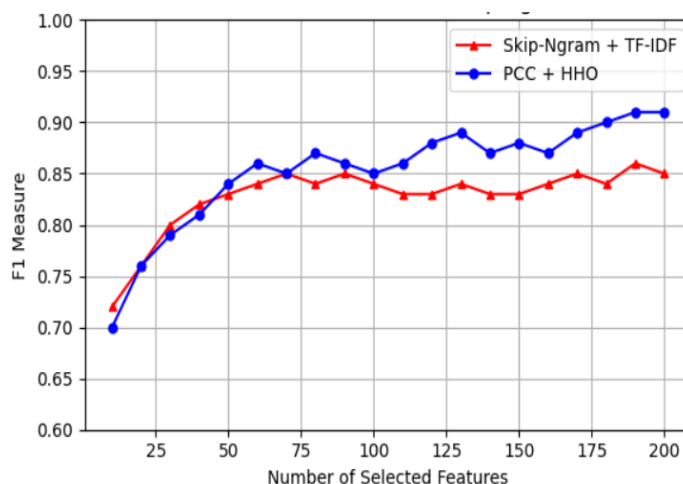


Figure 4. F1 Measures Vs. Number of Selected Features (Skip-Ngram + TF-IDF vs. PCC-HHO)

Figure 4. shows that PCC + HHO consistently outperforms Skip-Ngram + TF-IDF in terms of F1 measure as the number of selected features increases. PCC + HHO steadily improves, while Skip-Ngram + TF-IDF plateaus around 75 features and fluctuates, indicating possible inclusion of irrelevant features.

F. BiLSTM-RNN Model

Deep learning (DL), a subset of machine learning (ML), utilizes neural networks with multiple layers to identify patterns and solve complex tasks, such as sentiment analysis. Notable advancements in this area include Recurrent Neural Networks (RNNs), which process sequential data through feedback loops, enabling the retention of information from previous inputs. Long Short-Term Memory (LSTM), a specialized type of RNN, improves this process by using cell states and gate mechanisms to manage memory, allowing the network to effectively maintain and update information over time [29] [30]. Bidirectional LSTMs (Bi-LSTM) further enhance performance by processing data in both forward and backward directions, addressing challenges like gradient vanishing and better capturing the context within sequences [31] [32].

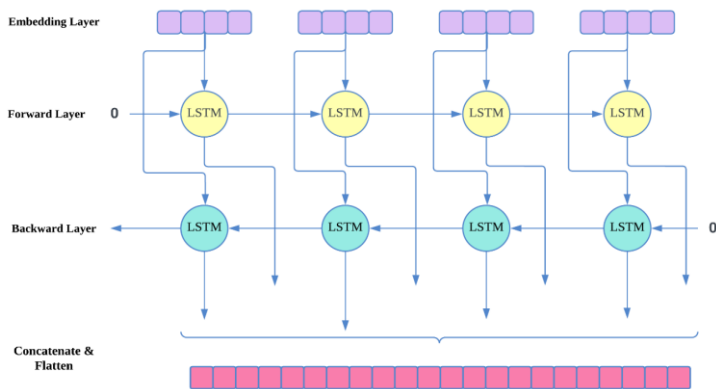


Figure 5. Bi-LSTM Unit

- **Embeddings:** In NLP and RNNs, embeddings are essential tools that convert words or categorical variables into continuous vector representations, capturing intrinsic meanings. Positioned at the input layer, these embeddings replace one-hot encodings by mapping words to lower-dimensional vectors, which are learned during model training, thereby enhancing the performance of ML and DL models.
- **Dropout layer:** The implementation will span both the Bi-LSTM and dense layers, as well as between the Bi-LSTM and the Multi-head attention layer. The Multi-head Attention operates by concurrently and iteratively applying an attention mechanism multiple time. It then generates the anticipated dimension by linearly amalgamating these individual attention results. Utilizing this layer enables a focused approach towards relevant future parameters, effectively sidelining those that are not pertinent to the analysis.

- **Dense Layer:** In the network layer, there is a dense layer which houses its neurons. These neurons obtain inputs from all preceding layers. The composition of this layer includes a matrix of weights denoted as 'w' a bias vector labeled 'a', and the activation values from the previous layer, referred to as 'b'. The features that define this dense layer are as follows:

$$d = b(x \cdot w) + a$$

In this context, 'b' signifies the parameter that depends on specific elements. The term 'w' represents the matrix of weights, and 'a' refers to the vector of biases associated with the respective layer.

- **Soft-max Classifier:** In the process of sentiment analysis, the resultant vector is directly fed into a Soft-max layer. The prediction result is then derived as follows:

$$x = softmax(V) = \frac{\exp(V)}{\sum_{k=1}^c \exp(V)}$$

IV. EXPERIMENTAL RESULTS

A. Evaluation Metrics

The table outlines key evaluation metrics used in sentiment classification: Accuracy, Precision, Recall, Specificity and F1-score. Accuracy measures the proportion of correctly predicted samples to the total sample size, while Precision assesses the proportion of true positive predictions. Recall evaluates the correctly predicted positive classes against all positive samples, and the F1-score provides a weighted average of Precision and Recall, balancing both metrics for comprehensive performance assessment.

TABLE I. Evaluation Metrics

Indicators	Formula	Purpose
Accuracy	$A = \frac{TN + TP}{TP + TN + FP + FN}$	The proportion of correctly predicted samples to the total sample size
Precision	$P = TP / (TP + FP)$	The proportion of samples with correct predictions that are genuinely positive
Recall	$R = TP / (TP + FN)$	The proportion of correctly predicted positive classes to all positive class samples
F1-Score	$F1 = \frac{2 * (precision * recall)}{precision + recall}$	Weighted summed average of Precision and Recall
Specificity	$S = TN / (TN + FP)$	The proportion of correctly predicted negative samples out of all actual negative samples

B. Classification of Sentiments by Using Bi-LSTM-RNN

Table II. present the performance metrics of our sentiment classification model on two datasets: SemEval-2014 and Sentiment140. The metrics evaluated include Precision, Recall, Specificity and F1-Score for positive, negative, and neutral sentiments. These results demonstrate the model's ability to accurately classify sentiments across different text categories, achieving high accuracy and consistency in both datasets.

Table II. Sentiment Analysis Performance Metrics

Dataset	Sentiments	Precision	Recall	Specificity	F1-Score	Accuracy
SemEval-2014	Positive	0.962	0.954	0.970	0.958	0.965
	Negative	0.950	0.944	0.965	0.947	0.954
	Neutral	0.959	0.962	0.968	0.961	0.958
Sentiment140	Positive	0.976	0.970	0.980	0.973	0.978
	Negative	0.966	0.960	0.975	0.963	0.966
	Neutral	0.975	0.980	0.985	0.977	0.975

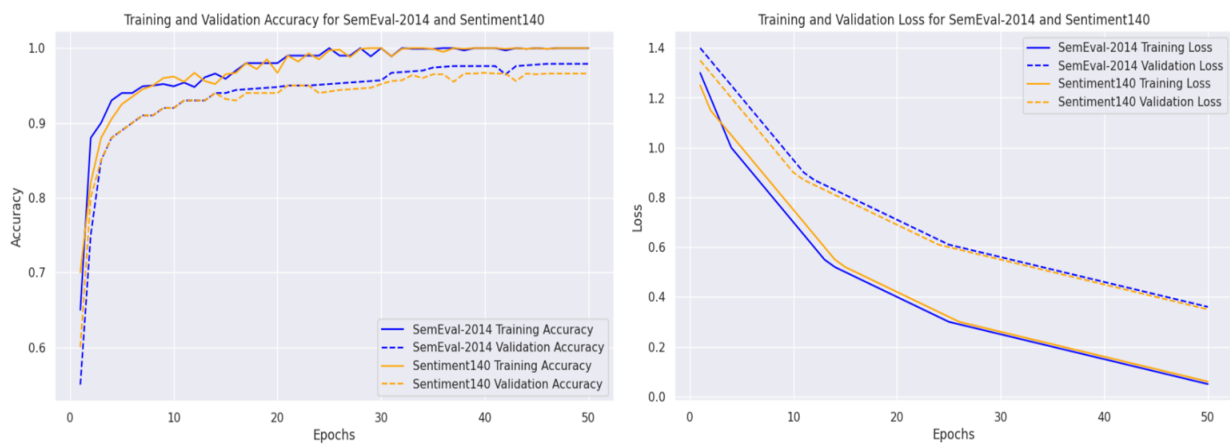


Figure 6. Training & Validation Accuracy and Loss

Figure 6 illustrates the training and validation accuracy curves over 50 epochs for the SemEval-2014 and Sentiment140 datasets using the Bi-LSTM-RNN model. The training and validation accuracy curves show a steady improvement, with both datasets reaching above 96% accuracy over 50 epochs.

The loss curves exhibit a consistent decrease, indicating effective learning and convergence of the model. The Bi-LSTM-RNN effectively captures the contextual dependencies in the text, leading to strong performance in sentiment analysis for both datasets.

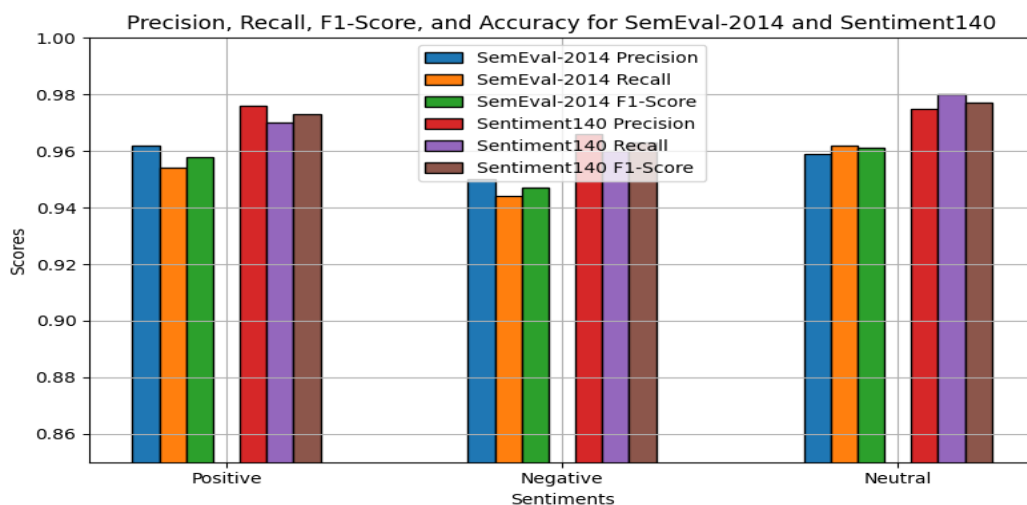


Figure 7. Precision, Recall & F1-Score of Sentiments

The bar graph compares the precision, recall, and F1-scores of SemEval-2014 and Sentiment140 across Positive, Negative, and Neutral sentiments. Sentiment140 generally performs better, particularly in precision and F1-scores for Positive and Neutral sentiments. The differences across metrics are small but notable, illustrating subtle performance variations between the datasets.

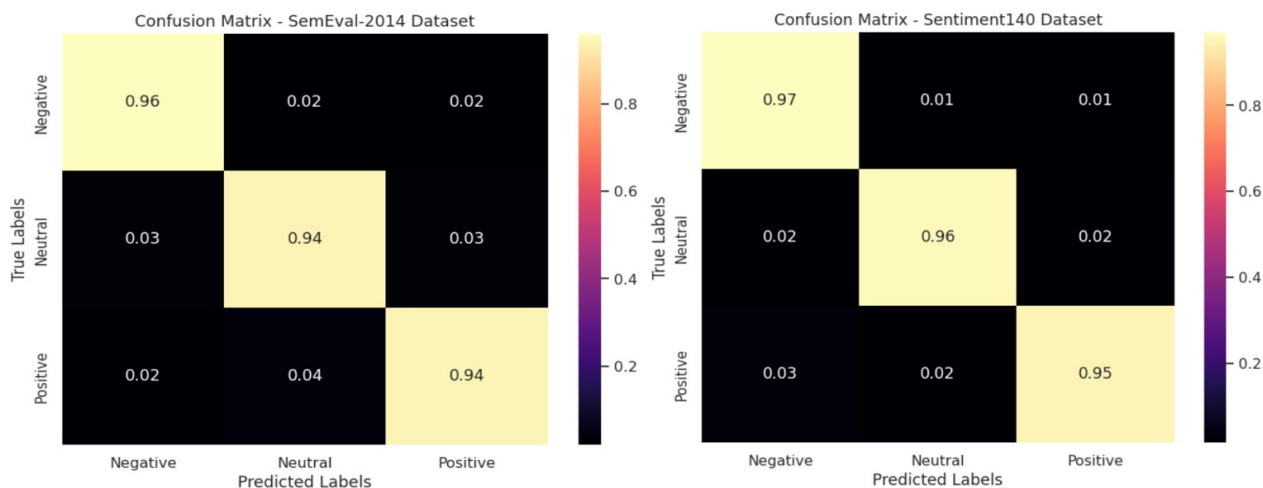


Figure 8. Confusion Matrices of Both Datasets

The confusion matrices in figure 8 shows the performance of the BiLSTM-RNN model on the SemEval-2014 and Sentiment140 datasets. Both matrices highlight strong classification accuracy for the Negative, Neutral, and Positive sentiments. These results demonstrate the model's effectiveness in sentiment analysis across different datasets. The performance comparison in Table 3 (SemEval-2014) and Table

4 (Sentiment140) shows that the PCC-HHO with BiLSTM-RNN model outperforms traditional models like SVM, Logistic Regression, KNN, Naive Bayes, and deep learning models like LSTM in precision, recall, F1-score, specificity, and accuracy. It delivers the most accurate and reliable results for sentiment analysis on both datasets.

Table III. Comparison Table for SemEval-2014 Dataset

Model	Precision	Recall	Specificity	F1-Score	Accuracy
PCC-HHO with BiLSTM-RNN	0.959	0.954	0.970	0.961	0.965
SVM	0.900	0.890	0.910	0.895	0.905
Logistic Regression (LR)	0.885	0.880	0.900	0.883	0.895
LSTM	0.920	0.910	0.920	0.915	0.918
KNN	0.870	0.860	0.875	0.865	0.880
Decision Tree (DT)	0.860	0.850	0.870	0.855	0.865
Naive Bayes	0.840	0.830	0.860	0.835	0.850

Table IV. Comparison Table for SemEval-2014 Dataset

Model	Precision	Recall	Specificity	F1-Score	Accuracy
PCC-HHO with BiLSTM-RNN	0.959	0.954	0.970	0.961	0.965
SVM	0.900	0.890	0.910	0.895	0.905
Logistic Regression (LR)	0.885	0.880	0.900	0.883	0.895
LSTM	0.920	0.910	0.920	0.915	0.918
KNN	0.870	0.860	0.875	0.865	0.880
Decision Tree (DT)	0.860	0.850	0.870	0.855	0.865
Naive Bayes	0.840	0.830	0.860	0.835	0.850

Comparison of models across both the SemEval-2014 (Figure 9) and Sentiment140 (Figure 10) datasets highlights the superior performance of the PCC-HHO with BiLSTM-RNN model in all metrics, including precision, recall, specificity, F1-score, and accuracy. While traditional models such as SVM, Logistic Regression, and LSTM perform reasonably well, PCC-HHO with BiLSTM-RNN consistently outperforms them, demonstrating its effectiveness in sentiment analysis tasks. The performance trends are similar across both datasets, showcasing the robustness of the model.

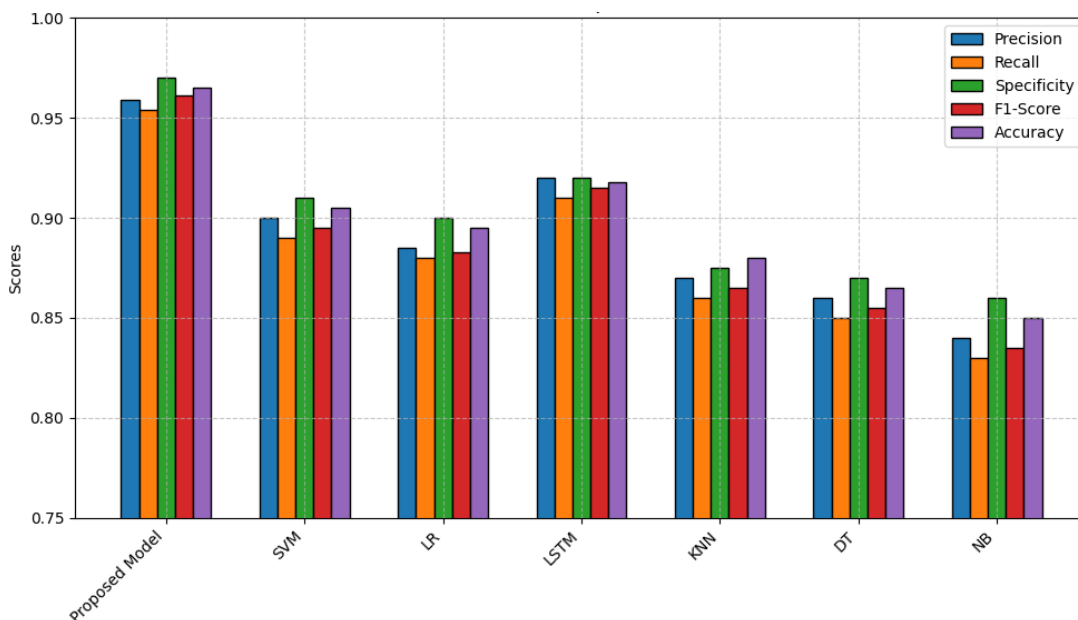


Figure 9. SemEval-2014: Comparison of Metrics

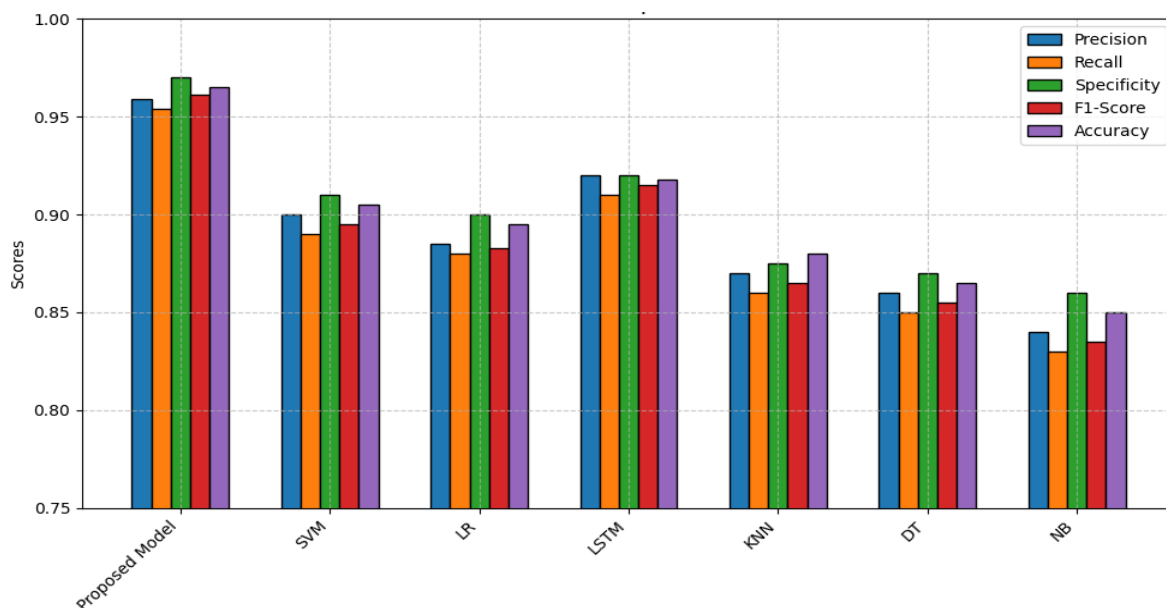


Figure 10. Sentiment140: Comparison of Metrics

V. CONCLUSION

In conclusion, this research presents a hybrid sentiment analysis model that effectively integrates advanced Natural Language Processing (NLP) techniques with an optimized feature selection process. By combining TF-IDF, Skip N-gram models, and a feature selection approach using Pearson Correlation Coefficient (PCC) and Harris Hawks Optimization (HHO), the model constructs distinctive feature vectors that enhance sentiment classification. The Bi-LSTM-RNN model successfully captures contextual dependencies, contributing to the model's high accuracy. Additionally, a comparison with existing models such as SVM, Logistic Regression, and LSTM highlights the superior performance of the hybrid approach. The proposed model achieved state-of-the-art results on both the SemEval-2014 and Sentiment140 datasets, with accuracies

of 96.54% and 97.81%, respectively. These results demonstrate the model's effectiveness in accurately classifying sentiments across diverse textual datasets, offering a robust and scalable solution for sentiment analysis.

ACKNOWLEDGMENT

I would like to express my heartfelt gratitude to my research colleagues for their invaluable guidance and support throughout the course of this research. Their expertise and encouragement have been instrumental in shaping the success of this study.

CONFLICT OF INTEREST

The author(s) declare that the publication of this article has no conflict of interest.

REFERENCES

- [1] F. Neri, C. Aliprandi, F. Capeci and M. Cuadros, "Sentiment analysis on social media.," In 2012 IEEE/ACM international conference on advances in social networks analysis and mining.IEEE, pp. 919-926, 2012.
- [2] B. Agarwal and N. Mittal, "Prominent feature extraction for sentiment analysis," Berlin: Springer International Publishing, pp. 21-45, 2016.
- [3] S. Naseem, T. Mahmood, M. Asif, J. Rashid, M. Umair and M. Shah, "Survey on sentiment analysis of user reviews.," In 2021 International Conference on Innovative Computing (ICIC). IEEE, pp. 1-6, 2021.
- [4] A. Dadhich and B. Thankachan, "Sentiment analysis of amazon product reviews using hybrid rule-based approach," In Smart Systems: Innovations in Computing: Proceedings of SSIC. Springer Singapore., pp. 173-193, 2022.
- [5] N. C. Dang, M. N. Moreno-García and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," Electronics, vol. 9, no. 3, p. 483, 2020.
- [6] J. Mutinda, W. Mwangi and G. Okeyo, "Lexicon-pointed hybrid N-gram Features Extraction Model (LeNFEM) for sentence level sentiment analysis," Engineering Reports, vol. 3, no. 8, 2021.
- [7] R. Al-Wajih, S. J. Abdulkadir, N. Aziz, Q. Al-Tashi and N. Talpur, "Hybrid binary grey wolf with Harris hawks optimizer for feature selection," IEEE Access, vol. 9, pp. 31662-31677, 2021.
- [8] A. Appathurai and P. Deepa, "Radiation induced multiple bit upset prediction and correction in memories using cost efficient CMC," Informacije MIDEM, vol. 46, no. 4, pp. 257-266, 2016.
- [9] A. A. Movassagh, J. A. Alzubi, M. Gheisari, M. Rahimi, S. Mohan, A. A. Abbasi and N. Nabipour, "Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model," Journal of Ambient Intelligence and Humanized Computing, pp. 1-9, 2021.
- [10] A. A. Movassagh, J. A. Alzubi, M. Gheisari, M. Rahimi, S. Mohan, A. A. Abbasi and N. Nabipour, "Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model," Journal of Ambient Intelligence and Humanized Computing, pp. 1-9, 2021.
- [11] O. A. Alzubi, J. A. A. Alzubi, S. Tedmori, H. Rashaideh and O. Almomani, "Consensus-based combining method for classifier ensembles," Int. Arab J. Inf. Technol, vol. 15, no. 1, pp. 76-86, 2018.
- [12] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," Concurrency and Computation: Practice and Experience, vol. 33, no. 23, 2021.
- [13] R. V. Karthik and S. Ganapathy, "A fuzzy recommendation system for predicting the customers interests using sentiment analysis and ontology in e-commerce," Applied Soft Computing, vol. 108, p. 107396, 2021.
- [14] G. Zhao, X. Lei, X. Qian and T. Mei, "Exploring users' internal influence from reviews for social recommendation," IEEE transactions on multimedia, vol. 21, no. 3, pp. 771-781, 2018.
- [15] I. Qutab, U. Fatima, M. Aqeel and I. A. Butt, "Analyzing COVID-19 Sentiments on Twitter: An Effective Machine Learning Approach," International Journal of Innovative Science and Research Technology, vol. 9, no. 8, pp. 841-850, 2024.
- [16] A. P. Jain and P. Dandannavar, "Application of machine learning techniques to sentiment analysis," International Conference on Applied and Theoretical Computing and Communication Technology (iCATecT). IEEE., pp. 628-632, 2016.
- [17] R. Bose, R. K. Dey, S. Roy and Sarddar, "Sentiment analysis on online product reviews," In Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2018. Springer Singapore., pp. 559-569, 2020.
- [18] Y. Zhang, J. Du, X. Ma, H. Wen and G. Fortino, "Aspect-based sentiment analysis for user reviews," Cognit.Comput, vol. 13, no. 5, p. 1114-1127, 2021.
- [19] I. Qutab, K. Malik and H. Arooj, "Sentiment classification using multinomial logistic regression on Roman Urdu text," Int. J. Innov. Sci. Technology, vol. 4, no. 2, pp. 323-335, 2022.
- [20] I. Qutab, K. Malik and H. Arooj, "Sentiment analysis for roman urdu text over social media, a comparative study," arXiv preprint arXiv:2010.16408, 2020.
- [21] L. Mai and B. Le, "Joint sentence and aspect-level sentiment analysis of product comments," Annals of Operations research, vol. 300, pp. 493-513, 2021.
- [22] S. Wassan, X. Chen, T. Shen, M. Waqar and N. Z. Jhanjhi, "Amazon product sentiment analysis using machine learning techniques," Revista Argentina de Clínica Psicológica, vol. 30, no. 1, p. 695, 2021.
- [23] G. Alorini, D. B. Rawat and D. Alorini, "LSTM-RNN based sentiment analysis to monitor COVID-19 opinions using social media data," In ICC 2021-IEEE International Conference on Communications. IEEE., pp. 1-6, 2021.
- [24] O. A. Alzubi, J. A. Alzubi, A. M. Al-Zoubi, M. A. Hassonah and U. Kose, "An efficient malware detection approach with feature weighting based on Harris Hawks optimization," Cluster Computing, pp. 1-19, 2022.
- [25] O. A. Alzubi, J. A. Alzubi, M. Alweshah, I. Qiqieh, S. Al-Shami and M. Ramachandran, "An optimal pruning algorithm of classifier ensembles: dynamic programming approach," Neural Comput. Appl, vol. 32, p. 16091-16107, 2020.
- [26] Y. Hao, T. Mu, R. Hong, M. Wang, X. Liu and J. Y. Goulermas, "Cross-domain sentiment encoding through stochastic word embedding," IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 10, pp. 1909-1922, 2019.
- [27] P. Maria, G. Dimitrios, P. John, P. Harris, A. Ion and M. Suresh, "SemEval- 2014 Task 4: Aspect Based Sentiment Analysis," In: Proc. 8th Int. Workshop Semantic Eval. (SemEval), p. 27-35, 2014.
- [28] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, "Harris hawks optimization: Algorithm and applications," Future generation computer systems, vol. 97, pp. 849-872, 2019.
- [29] G. Zhao, X. Lei, X. Qian and T. Mei, "Exploring users' internal influence from reviews for social recommendation," IEEE transactions on multimedia, vol. 21, no. 3, pp. 771-781, 2018.
- [30] A. P. Jain and P. Dandannavar, "Application of machine learning techniques to sentiment analysis," 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATecT) IEEE., pp. 628-632, 2016.
- [31] P. Patel, D. Patel and Naik, "Sentiment analysis on movie review using deep learning RNN method.," In Intelligent Data Engineering and Analytics: Frontiers in Intelligent Computing: Theory and Applications. Springer Singapore, vol. 2, pp. 155-163, 2020.
- [32] N. Wedjdane, R. Khaled and K. Okba, "Better Decision Making with Sentiment Analysis of Amazon reviews," International Conference on Information Systems and Advanced Technologies (ICISAT).IEEE, pp. 1-7, 2021.