

## Research Article

# An Efficient Sentiment Classification Method with the Help of Neighbors and a Hybrid of RNN Models

Israa K. Salman Al-Tameemi <sup>1,2</sup> Mohammad-Reza Feizi-Derakhshi <sup>1</sup>  
Saeed Pashazadeh <sup>3</sup> and Mohammad Asadpour <sup>3</sup>

<sup>1</sup>Computerized Intelligence Systems Laboratory, Department of Computer Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz 51368, Iran

<sup>2</sup>State Company for Engineering Rehabilitation and Testing, Iraqi Ministry of Industry and Minerals, Baghdad 10011, Iraq

<sup>3</sup>Department of Computer Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz 51368, Iran

Correspondence should be addressed to Mohammad-Reza Feizi-Derakhshi; [mfeizi@tabrizu.ac.ir](mailto:mfeizi@tabrizu.ac.ir)

Received 18 June 2023; Revised 4 October 2023; Accepted 8 December 2023; Published 27 December 2023

Academic Editor: Roberto Natella

Copyright © 2023 Israa K. Salman Al-Tameemi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Some texts that are challenging to recognize on their own may become more understandable in a neighborhood of related texts with similar contexts. Motivated by this intuition, a novel deep text sentiment classification (DTSC) model is proposed to improve the model's performance by incorporating the neighborhood of related texts. Our framework uses a nonparametric approach to construct neighborhoods of related texts based on Jaccard similarities. Then, a new deep recurrent neural network architecture is proposed, comprising two distinct modules: bidirectional long short-term memory (Bi-LSTM) and gated recurrent unit (GRU). The proposed model aims to effectively capture informative features from the input text and its neighbors. The result of each module is processed through the maximum operation, which selects the most pertinent data. Finally, the extracted features are concatenated and subjected to classification to achieve accurate sentiment prediction. Previous studies have commonly employed a parametric approach to represent textual metadata. However, our approach utilizes a nonparametric approach, enabling our model to perform strongly even when the text vocabulary varies between training and testing. The proposed DTSC model has been evaluated on five real-world sentiment datasets, achieving 99.60% accuracy on the Binary\_Getty (BG) dataset, 98.32% accuracy on the Binary\_iStock (BIS) dataset, 96.13% accuracy on Twitter, 82.19% accuracy on the multi-view sentiment analysis (MVSA) dataset, and 87.60% accuracy on the IMDB dataset. These findings demonstrate that the proposed model outperforms established baseline techniques in terms of model evaluation criteria for text sentiment classification.

## 1. Introduction

The present generation of widely available and affordable web technologies generates significant social big data with perspectives that aid decision-making. Sentiment analysis (SA) is a computational study that analyses individuals' perspectives, opinions, and emotions toward a particular entity that may track individuals' moods and viewpoints by analyzing unstructured, multimodal, informal, noisy, and high-dimensional social data [1]. SA is a subset of natural language processing (NLP) that can be used in various real-world applications, including financial and stock price

forecasting [2], politics [3], and medicine [4, 5]. Many researchers have dedicated significant efforts to investigating textual SA [6–10] through various methodologies, resulting in notable advancements on social media platforms. One significant constraint of current sentiment classification systems is their predominant dependence on a post or tweet's textual content.

However, on social platforms such as Twitter, Flickr, and Instagram, there's a wealth of metadata available alongside the text in posts. People attach metadata to their shared content, including user-generated labels, as a means of engaging with others. This metadata, which includes

information about the attributes of social media posts and their authors, serves as a valuable source of context for dissecting the expressed sentiment within the text. Moreover, these metadata elements can function as an additional asset alongside the text-based features when it comes to successfully executing sentiment classification. For instance, the sentence “Nowruz Mubarak!” displayed in the green bounding box of Figure 1 carries an ambiguous meaning, and it is challenging to discern the true sentiment behind this sentence. Now, we consider the sentences in the red bounding box: “Happy Iranian New Year (Nowruz), Happy Nowruz, . . .” These sentences can help determine the true sentiment expressed by the input sentence, which can be positive. As a result, SA can greatly benefit from the metadata or neighbors of input sentences. Based on this intuition, we enhance SA by augmenting each input text with the neighborhood of related texts.

Previous studies have utilized metadata extracted from tweets, including metrics such as URL counts, hashtags, and mentions, as features for sentiment analysis. This approach assumes that texts sharing similar contents or features can convey similar sentiments [11, 12]. However, these studies relied on parametric models for text analysis, assuming consistent vocabularies between training and testing datasets. In the real world, metadata vocabularies can change as new tags emerge. Therefore, it is crucial to investigate the incorporation of metadata, or neighboring data, to adapt to these evolving vocabularies and enhance sentiment classification. In conclusion, a compelling rationale exists for utilizing both tags and metadata, facilitating the seamless integration of additional data sources to enhance the representation of the original input data.

The primary technical contribution of our study involves the nonparametric approach for generating text neighborhoods, as depicted in Figure 1. Subsequently, we employ a novel parametric model to learn the degree of informative representation derived from an initial input and its corresponding neighbors. This strategy enables our model to perform complex or impractical tasks using existing techniques and demonstrates a cutting-edge level of performance in text classification and SA. We specifically demonstrate the following capabilities of our model.

*1.1. Adapt to Changing Vocabularies.* Our model can handle various vocabularies during training and testing thanks to our nonparametric approach to finding near neighbors. Even when the training and testing vocabulary is entirely unrelated, our model still performs well. In other words, during training, the model learns the specific words in the training data and their relationships and similarities. However, it can still effectively classify sentiments in text even when it encounters entirely different words during the testing phase. This adaptability is attributed to the model’s nonparametric approach, allowing it to generalize to new categories of text metadata and adjust to changes in that metadata over time. So, despite variations in the text vocabulary, the model’s performance remains robust due to its unique design and training process.

*1.2. Handle Different Text Types with and without Metadata.* The proposed model can calculate neighbors from either the input data or the associated metadata. This feature ensures that our approach does not constrain the exclusive use of metadata or input data, showcasing the model’s remarkable adaptability across various text data types. It is essential to highlight the significance of both the presence and integration of metadata, as these aspects can significantly influence the model’s performance in sentiment analysis tasks. This reinforces the pivotal role of metadata, underscoring its potential to enhance the model’s accuracy, and stresses the need to consider it a crucial factor in data analysis.

*1.3. Efficient Deep Model.* An efficient model is designed to jointly learn representations from samples and their neighbors to generate intraclass-oriented meaningful representation.

The main objective of this study is to identify substantial correlations between the input text and the neighborhood of related texts within various sentiment datasets. It seeks to explore whether this metadata or neighboring texts can indicate potential biases in sentiment labeling. Lastly, we demonstrate how leveraging this metadata as features in a classifier might aid in enhancing sentiment classifiers and sentiment quantification.

The paper’s remaining sections are structured as follows: Section 2 presents the literature review. Section 3 describes the proposed model in depth. Section 4 illustrates the findings of the experiments. Section 5 concludes the study.

## 2. Literature Review

The significance of sentiment analysis (SA) increases in the context of natural language processing (NLP) when dealing with a substantial volume of user-generated textual content. Many supervised traditional machine learning (ML) classifiers are used to classify sentiment based on various features [13]. Ahuja et al. [14] examined the effects of TF-IDF word level and  $N$ -gram on the SS-Tweet sentiment analysis dataset. The TF-IDF word level approach in sentiment analysis using six machine learning classification algorithms outperforms  $N$ -gram features by 3-4%. Mee et al. [15] examined the relationship between textual qualities and Twitter user characteristics using regression and sentiment analysis, specifically the TF-IDF approach. Kunal et al. [16] recommended combining Tweepy and TextBlob, a Python framework, to analyze and classify tweets using the Naive Bayes (NB) classifier. Htet and Myint [17] developed a system for analyzing social media data from Twitter to assess individuals’ health, education, and business status. This system employs the maximum entropy (ME) classifier to identify specific requirements. Obiedat et al. [7] presented a hybrid approach that combines support vector machine (SVM), particle swarm optimization, and various oversampling techniques to address the issue of imbalanced data. The SVM has enhanced sentiment prediction using the restaurant reviews’ dataset.

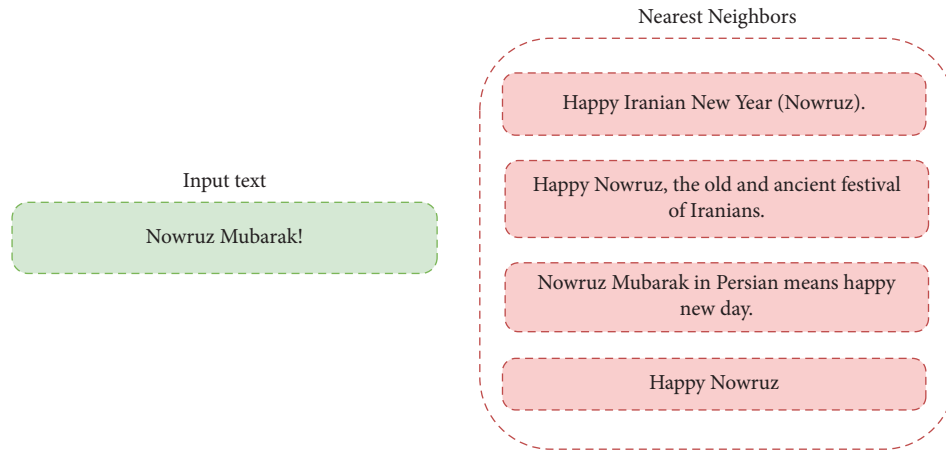


FIGURE 1: Even for humans, it might be challenging to understand some sentences without additional context. On social media, however, similar texts are frequently shared. Based on this intuition, we retrieve a neighborhood of sentences with similar words to help define the sentiment of the input text given an ambiguous sentence, such as “Nowruz Mubarak!”

Recently, text analysis has experienced advantageous outcomes by utilizing diverse deep learning (DL) models [18, 19], which have been extensively implemented in numerous research investigations. The categorization of short text was performed by Wang et al. [20] through the use of a convolutional neural network (CNN). Basiri et al. [21] proposed a CNN-recurrent neural network (RNN) model that utilizes attention mechanisms to capture past and future contexts in text sentiment analysis. The incorporation of bidirectional temporal information flow enhances the accuracy of classification in this approach. The present trend in the field of sentiment analysis involves the development of innovative text classification methods utilizing deep learning techniques such as CNN [22, 23] and long short-term memory (LSTM) [24]. While CNNs can collect and analyze local data, they may be less effective in capturing long-range dependency. The limitation of sequential modeling of texts across sentences can be overcome by utilizing the LSTM technique. In any case, its performance in collecting local information is suboptimal. The integration of CNN and LSTM becomes essential for enhancing the efficacy of text sentiment classification [25–27]. Li et al. [28] presented a new padding methodology that enhanced consistency in the dimensions of input data instances, thereby augmenting the amount of sentiment-oriented information incorporated in every review. Integrating a sentiment analysis model denoted as “lexicon,” which utilizes deep learning techniques, involved incorporating two-channel CNN-LSTM/Bi-LSTM family models through parallelization techniques. Abid et al. [29] presented a unified procedure for SA on the Twitter platform. This approach incorporates an RNN architecture to capture long-term dependencies efficiently and utilizes a CNN and Global Vectors (GloVe) for Word Representation as a word embedding technique. The experimental outcomes performed superior to the baseline model upon evaluating the Twitter corpora. Dang et al. [30] proposed the integration of LSTM networks, CNNs, and SVM in hybrid deep SA learning models. The proposed model was evaluated utilizing eight textual datasets

comprising tweets and reviews from various domains. The findings indicate that the hybrid models performed superior to the single models in sentiment analysis across all datasets.

Salur and Aydin [31] presented a new hybrid deep learning model that combines different types of word embeddings, namely, Word2Vec, FastText, and character-level embeddings, in combination with various deep learning models such as LSTM, GRU, Bi-LSTM, and CNN. The model under consideration amalgamates features from various deep learning word embedding methods and classifies textual information based on its emotional content. Zulqarnain et al. [32] proposed a novel methodology for SA by utilizing an encoder approach with a two-state GRU named E-TGRU. This framework was designed to enhance the effectiveness of SA. The study’s results indicate that, with adequate training data, the GRU model can proficiently acquire the vocabulary utilized in user opinions. The findings indicate that E-TGRU exhibited superior performance compared to GRU, LSTM, and Bi-LSTM. Li et al. [33] proposed a sentiment classification model for analyzing online restaurant reviews, integrating Word2Vec, bi-directional GRU, and the attention technique. The results show that the model’s performance surpassed established sentiment analysis models. Kamyab et al. [34] presented a novel approach to sentiment analysis utilizing attention mechanisms in conjunction with CNNs and two distinct bidirectional recurrent neural networks. The proposed method aims to enhance the understanding of sentiment in textual data. Initially, a preprocessor was utilized to improve the quality of the data. Then, max-pooling was used in conjunction with a CNN layer to reduce the dimensionality of features and retrieve contextual information. In addition, the study employed two autonomous bidirectional recurrent neural networks, namely, LSTM and GRU, to effectively capture long-term dependencies. Ultimately, the attention mechanism was implemented to highlight the degree of attention attributed to each word.

Mishra et al. [11] developed a sentiment analysis tool that is both cost-free and open-source, featuring a graphical user

interface. This tool enables users to perform two key functions: firstly, to retrain the weights of a given model by relabeling predictions and/or adding labeled instances and secondly, to tailor lexical resources to address errors in sentiment lexicons, such as false positives and false negatives. The proposed approach has the potential to offer advantages in iteratively improving or augmenting models in a readily available manner while disregarding the expenses associated with training a new model from the beginning and reducing predictive precision over time. Mishra and Diesner [12] analyzed the metadata characteristics at the user and tweet levels, identifying associations and relationships between these characteristics and the log odds for sentiment categories. The reliability of this analysis is strengthened by replicating the experiments on current tweets obtained from the user population present in our datasets. The results suggest that most patterns identified in this analysis exhibit high consistency. The metadata characteristics that have been identified are ultimately employed as features for a sentiment classification algorithm, resulting in an improved outcome for sentiment classification.

Deep learning-based sentiment analysis and the BERT technique have recently piqued the interest of researchers. Chiorrini et al. [35] suggested two BERT-based text classification approaches: BERT-base and cased BERT-base. Their investigation used two independent datasets for sentiment analysis and emotion recognition. They noted that BERT offers positive text classification results. Huang et al. [36] proposed an innovative DCNN-Bi-GRU (deep convolutional neural network bidirectional gated recurrent) text categorization model. The word semantic representation language model is trained using BERT. The DCNN-Bi-GRU hybrid model receives the dynamically generated semantic vector from the word context. This model is validated by the CCERT Chinese e-mail sample set and movie comment data set experiments. Bello et al. [37] proposed utilizing bidirectional encoder representations from transformers (BERT) for text classification in NLP and various variants. The experimental results indicate that the integration of BERT with CNN, BERT with RNN, and BERT with Bi-LSTM yields favorable outcomes in terms of accuracy rate, precision rate, recall rate, and  $F1$ -score when compared to the outcomes achieved by employing these deep learning models with Word2Vec or without any variation.

The earlier research has exhibited satisfactory results using different ML and DL models. However, achieving high accuracy in sentiment categorization remains a formidable challenge, particularly when dealing with data from social media platforms. One of the limitations of current sentiment classification systems is their heavy reliance on traditional techniques, such as bag-of-words (BOW) and  $N$ -gram approaches, which use term frequency as a feature. While these methods are straightforward and effective, they generate feature vectors that are often sparse and high-dimensional. This can lead to scalability issues and potential overfitting, even with regularization techniques. Furthermore, these models operate under the implicit assumption that sentiment expression in a post is solely conveyed through the text data, without

considering contextual factors. However, social media platforms like Twitter offer an opportunity to access rich metadata alongside the textual content of posts. This metadata includes information about the characteristics of social network posts and their authors, which can provide valuable contextual cues for analyzing sentiment in a tweet. Additionally, leveraging this metadata can complement the textual features in the sentiment classification task.

Although few studies have relied on tweet metadata assuming consistent language patterns in training and testing data, the reality is that metadata vocabularies can change as new categories and trends emerge. Therefore, finding methods to incorporate this changing metadata or neighboring data is crucial to improve sentiment analysis accuracy. Motivated by these observations, we introduce an innovative deep text sentiment classification (DTSC) model. Our model features a nonparametric approach to generating text neighborhoods, making it adaptable to a wide range of signals and capable of generalizing to new categories of text vocabularies. Then, the input texts and their neighbors are converted into embedding vectors of lower dimensions, allowing neural networks to capture semantic word relationships. Importantly, this dense vector representation maintains a fixed size (embedding dimension), reducing model parameters and improving computational efficiency.

Additionally, we employ a novel parametric model to gauge the level of informative representation obtained from the input text and its associated neighbors. This unique approach equips our model with the ability to tackle complex tasks that may have been challenging with conventional techniques. As a result, our model demonstrates state-of-the-art performance in text sentiment classification.

### 3. The Proposed Model

A novel deep text sentiment classification (DTSC) model is proposed, as shown in Figure 2. The model considers the neighborhoods in which input text features are embedded. Our model uses a nonparametric approach to construct neighborhoods of related texts based on Jaccard similarities to develop a perfect system that can handle a wide range of signals, generalize to new categories of text metadata, and adjust to changes in that metadata over time. The input texts and their neighbors are transformed into embedding vectors to efficiently capture the semantic relationships between words and texts, making them more appropriate for in-depth analysis. Then, a new deep recurrent neural network architecture is proposed. Specifically, two distinct modules, Bi-LSTM and GRU, extract valuable representations from an input text and its neighbors. The outputs of each module (extracted features of a text and its neighbors) are fed through the maximum operation, which selects the most pertinent data. Finally, the extracted features are concatenated and deeply fused using multiple fully connected layers with a classifier layer to perform sentiment classification. The weights and biases are shared among the input text and its  $k$  neighbors. In other words, the input text and its neighbors are passed through a common architecture.

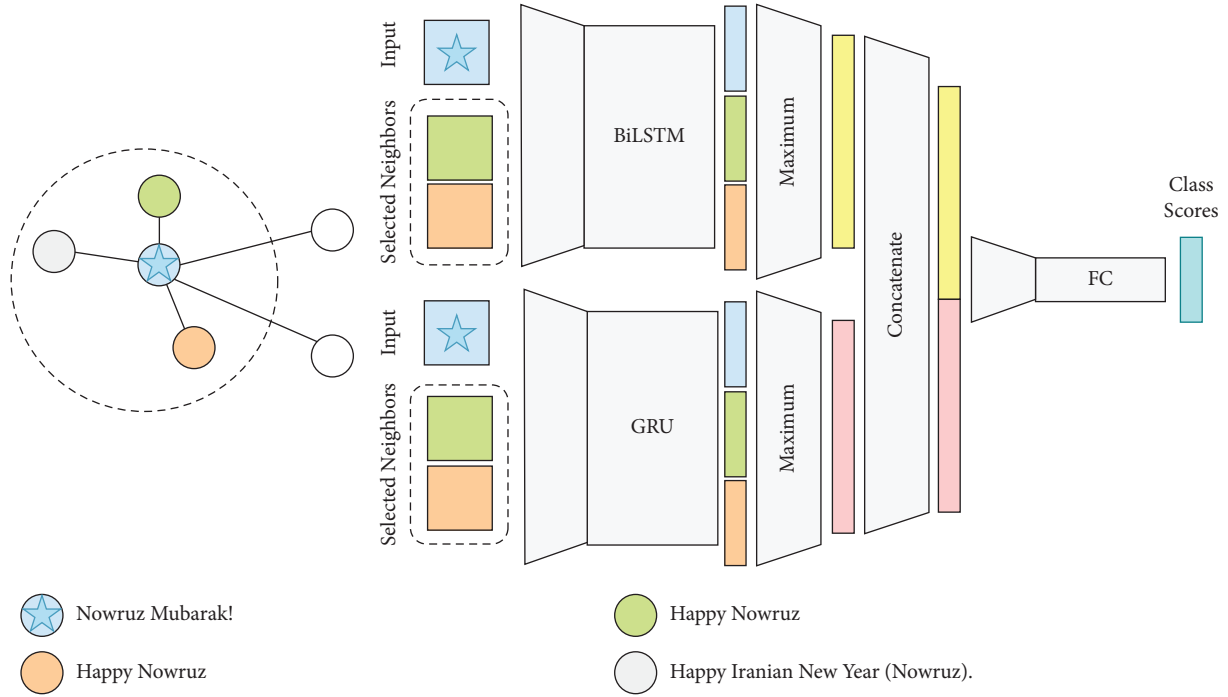


FIGURE 2: The DTSC model.

**3.1. Candidate Neighborhoods.** In the nonparametric approach, we assume that integrating neighboring data alongside the primary input during network training will yield extracted features that demonstrate similarity among samples belonging to the same class. In other words, it is assumed that if two texts are related or share similar content, their feature representations should also be similar after training. The key challenge lies in the selection of the nearest neighbors. Our approach leverages the Jaccard measure between words to calculate text similarity, allowing for the nonparametric creation of candidate neighborhoods. In particular, Jaccard similarity is employed to assess how similar or dissimilar individual words are between different texts. This similarity measurement involves the following steps.

**3.1.1. Jaccard Similarity Calculation.** In this context, it measures how similar the words in one text are to those in another. The Jaccard similarity is a nonparametric measure that quantifies the similarity between two sets by comparing the intersection (common words) of the word sets in both texts to their union (all unique words in both texts), and it ranges between 0 (no similarity) and 1 (perfect similarity). It does not assume any specific probability distribution for the data but calculates the proportion of common elements. Given two sets  $A$  and  $B$ , the Jaccard similarity ( $J\text{simi}$ ) is defined as

$$J\text{simi}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (1)$$

where

- (1)  $|A|$  is the cardinality (number of elements) of set  $A$ .

- (2)  $|B|$  is the cardinality (number of elements) of set  $B$ .
- (3)  $|A \cap B|$  is the number of elements in the intersection of sets  $A$  and  $B$  (i.e., the number of elements common to both sets).
- (4)  $|A \cup B|$  is the number of elements in the union of sets  $A$  and  $B$  (i.e., the total number of distinct elements in both sets).

Concretely, for  $x, x' \in X$ , we compute

$$J\text{simi}(x, x') = \frac{|w_x \cap w_{x'}|}{|w_x \cup w_{x'}|}, \quad (2)$$

where  $w_x$  and  $w_{x'}$  represent the set of words for  $x$ -th sample and its nearest  $x'$ -th neighbor. We set  $J\text{simi}(x, x) = 0$  for all  $x \in X$ , to prevent a text from appearing in its neighborhoods. If the Jaccard similarity is high, the texts share many common words and are more similar. Conversely, if the Jaccard similarity is low, it suggests that the texts have fewer words in common and are more dissimilar.

**3.1.2. Creating Candidate Neighborhoods.** In each batch of data, candidate neighbors for each sample are computed using the Jaccard similarity. The text's calculated similarities are then grouped into "neighborhoods" or clusters. Texts with a high Jaccard similarity between their words are placed in the same neighborhood because they are considered more similar. These neighborhoods are groups of related texts that share common features or themes. The only limitation that can be considered is that the number of neighbors should be smaller than the batch size. This is because the neighbors are selected from among the closest samples within a batch.

In summary, this method employs Jaccard similarity calculations to evaluate word overlap between texts, facilitating their organization into potential neighborhoods or clusters. However, a critical transformation is applied to enhance the suitability of these textual representations for further analysis. The input texts and their neighbors (with the number of neighbors defined by the user) undergo a crucial conversion process, and they are transformed into embedding vectors.

This conversion is pivotal because it translates textual information into numerical vectors, enabling the model to effectively understand and work with the text. These embedding vectors capture the semantic relationships between words and texts, making them more appropriate for in-depth analysis. The input vectors and their corresponding candidate neighbors are introduced into a network featuring two distinct modules: Bi-LSTM and GRU. These modules leverage the embedded representations to extract valuable patterns and insights between the input samples and their neighbors, enabling efficient comprehension and classification of the sentiments expressed in the texts.

The rationale for implementing a nonparametric approach over parametric models lies in several key advantages:

- (1) **Adaptability to changing vocabularies:** nonparametric approaches can adapt seamlessly to varying vocabularies. This adaptability is crucial when working with text data from different sources or domains, where the vocabulary can be entirely unrelated between training and testing datasets. In contrast, parametric models often assume a fixed vocabulary, limiting their ability to handle such dynamic language usage.
- (2) **Complex and diverse data handling:** nonparametric approaches excel at managing complex and diverse data, which is common in real-world text applications. They can accommodate different word choices, expressions, and language styles, making them suitable for texts from various sources. Parametric models may struggle when confronted with data heterogeneity and nonlinear relationships.
- (3) **Enhanced performance in text classification and sentiment analysis (SA):** the combination of nonparametric neighborhood generation and a novel parametric model brings the best of both worlds. The nonparametric approach captures contextual information from neighbors, while the parametric model effectively learns informative representations. This synergy enables the model to perform complex tasks and achieve a cutting-edge level of performance in text classification and SA, surpassing the capabilities of traditional parametric models.
- (4) **Robustness in exploring correlations and bias detection:** beyond classification, the model's nonparametric foundation allows it to investigate correlations between the input text and related neighborhood texts. Furthermore, it explores the

potential for metadata to indicate biases in sentiment labeling. This capability is essential for understanding and addressing bias in sentiment analysis applications, ensuring more accurate and fair results.

In summary, the decision to employ a nonparametric model is justified by its adaptability to varying vocabularies, ability to handle complex and diverse data, improved performance, and robustness in exploring correlations and detecting biases. This hybrid approach, combining nonparametric and parametric techniques, is well suited for addressing the challenges posed by dynamic and diverse text data in the context of sentiment analysis and classification.

**3.2. Bi-LSTM.** The RNN [38] model has gained significant attention in NLP due to its complex architecture that facilitates effective feature extraction. The model demonstrates proficiency in processing short data sequences due to its singular memory, which renders it incapable of processing long-term dependency issues. As a result, the LSTM architecture is used as an extension of the RNN model to address the problem of long-term dependency in text SA. The LSTM model leverages the present word embedding and the preceding hidden state within the context of text sentiment data for every component or term to anticipate the upcoming hidden state. The hidden state  $h_t \in R^T$  ( $T$  is feature vector dimension) at time  $t$  is updated as follows:

$$\begin{aligned}
 i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\
 f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), \\
 o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_c x_t + U_c h_{t-1} + b_c), \\
 h_t &= o_t \circ \tanh(c_t),
 \end{aligned} \tag{3}$$

where  $\circ$  is the element-wise product symbol,  $\sigma(\cdot)$  is the sigmoid activation function,  $x_t$  represents the lower layer input at time step  $t$ , and  $\tanh$  is the tangent activation function.  $i_t$ ,  $f_t$ ,  $o_t$ , and  $c_t$  are input, forget, output, and memory gates, respectively. The parameters of the LSTM are  $W$ ,  $U$ , and  $b$ .

In sequence modeling tasks, it is beneficial to understand past and future contexts. Adding a second hidden layer in the unidirectional LSTM model expands the architecture, giving rise to the Bi-LSTM [39], which incorporates hidden connections that propagate in the reverse temporal sequence. The Bi-LSTM model consists of two sequences, which are as follows:

$$\begin{aligned}
 H_f &= (\vec{h}_1, \vec{h}_2, \vec{h}_3, \dots, \vec{h}_T), \\
 H_B &= (\overleftarrow{h}_1, \overleftarrow{h}_2, \overleftarrow{h}_3, \dots, \overleftarrow{h}_T).
 \end{aligned} \tag{4}$$

The output is shown in equation (4):

$$O = [H_f \oplus H_B], \tag{5}$$

where  $\oplus$  is the element-wise sum operation.

3.3. *GRU*. GRU is a distinctive variant within the family of RNNs [40]. The internal unit of the GRU is analogous to the LSTM internal unit [36], except that the GRU combines the forgetting and incoming ports into a single update port. Although it draws inspiration from the LSTM unit, this model maintains the LSTM’s capacity to overcome the vanishing gradient issue. The simplified internal architecture of GRUs facilitates their training process by reducing the computational complexity involved in enhancing the internal states. The hidden state  $h_t \in R^T$  ( $T$  is feature vector dimension) at time  $t$  is updated as follows:

$$\begin{aligned} z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z), \\ r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r), \\ \hat{h}_t &= \varnothing_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h), \\ h_t &= (1 - z_t) \circ \hat{h}_t + z_t \circ h_{t-1}, \end{aligned} \quad (6)$$

where  $\sigma$  is the sigmoid activation function and  $\varnothing_h$  is the hyperbolic tangent function. The operation  $\circ$  denotes the element-wise vector product,  $x_t$  denotes an input vector,  $h_t$  indicates the output vector,  $\hat{h}_t$  denotes the candidate activation vector,  $z_t$  is the update gate vector,  $r_t$  is the reset gate vector, and the learned parameters are  $W, U$ , and  $b$ .

In the proposed architecture, the  $i$ -th input text ( $x_i$ ) and its  $i$ -th neighbor ( $z_i$ ) go through RNN-based layers and new representations for the input sample ( $v_{x_i}$ ) and its  $i$ -th neighbor ( $v_{z_i}$ ) are generated (equations (7) and (8)).

For Bi-LSTM-based representation,

$$\begin{aligned} v_{x_i}^{\text{BiLSTM}} &= \text{BiLSTM}(x_i), \\ v_{z_i}^{\text{BiLSTM}} &= \text{BiLSTM}(z_i). \end{aligned} \quad (7)$$

- (1)  $v_{x_i}^{\text{BiLSTM}}$  represents the new representation of the input text  $x_i$ , generated by applying the Bi-LSTM model to  $x_i$ . This captures both the backward and forward dependencies of a word in the  $i$ -th input text ( $x_i$ ).
- (2) Similarly,  $v_{z_i}^{\text{BiLSTM}}$  represents the new representation of the neighbor text  $z_i$ , generated using Bi-LSTM to capture the backward and forward dependencies of a word in the  $i$ -th neighbor ( $z_i$ ).

For GRU-based representation,

$$\begin{aligned} v_{x_i}^{\text{GRU}} &= \text{GRU}(x_i), \\ v_{z_i}^{\text{GRU}} &= \text{GRU}(z_i). \end{aligned} \quad (8)$$

- (4)  $v_{x_i}^{\text{GRU}}$  represents the new representation of the input text  $x_i$ , generated by applying the GRU model to  $x_i$ . GRU also extracts contextual and high-level textual information with long-term dependencies of the  $i$ -th input text ( $x_i$ ).
- (5) Similarly,  $v_{z_i}^{\text{GRU}}$  represents the new representation of the neighbor text  $z_i$ , generated using GRU to extract contextual and high-level textual information with long-term dependencies of the  $i$ -th neighbor ( $z_i$ ).

The representations that were generated from the sample ( $v_{x_i}^{\text{BiLSTM}}, v_{x_i}^{\text{GRU}}$ ) and its neighbors ( $v_{z_i}^{\text{BiLSTM}}, v_{z_i}^{\text{GRU}}$ ) are merged after applying the maximum operation that selects the most pertinent data, as shown in the following equation:

$$f_\theta(x_i) = \max(v_{x_i}^{\text{BiLSTM}}, v_{x_i}^{\text{GRU}}) + \max(v_{z_i}^{\text{BiLSTM}}, v_{z_i}^{\text{GRU}}), \quad (9)$$

where “+” represents the concatenation operation and  $f_\theta(x_i)$  is the output feature generated by combining the features from the text  $x_i$  and its neighbors.

Finally, the output feature  $f_\theta(x_i)$  is deeply fused using multiple fully connected layers, as shown in equation (10), with a classifier layer for sentiment classification. The weights and biases are shared among the input text and its  $k$  neighbors. In other words, the input text and its neighbors are passed through a common architecture.

$$y'_i = \sigma(W_x f_\theta(x_i) + b_x), \quad (10)$$

where  $y'_i$  is the label predictions of the  $i$ -th input text,  $\sigma$  is the activation function,  $b_x$  and  $W_x$  represent the biases and weights, and  $f_\theta(x_i)$  is the output feature of the RNN’s blocks.

The proposed method uses the cross-entropy objective function to train the network. The objective function calculates the loss, which is estimated between predicted labels (equation (10)) and the true label (equation (11)).

$$\mathcal{L} = - \sum_{i=1:N} y_i \log(y'_i), \quad (11)$$

where  $N$  is the number of classes and  $y_i$  indicates the accurate label of  $i$ -th text.

## 4. Experiment

The effectiveness of the proposed methodology is assessed and compared with several established baseline techniques, including Bi-LSTM, bidirectional-GRU (Bi-GRU), LSTM, GRU, CNN, and LSTM-CNN. The parameters and their respective values for the proposed method and other baseline approaches are displayed in Table 1.

The architectures of Bi-LSTM, LSTM, Bi-GRU, and GRU employ three layers of 128, 64, and 64 units. Furthermore, the classifier employs three fully connected layers with dimensions of 128, 64, 32, and 1 (or more based on class count). Additionally, a 128-dimensional word embedding is utilized.

In the 1D CNN model, three layers are employed, each consisting of 32 filters and a kernel size 5. A 1D max pooling layer is employed with a pooling size of three. A 1D global max pooling layer is employed in the convolutional-based model’s last layer. The classifier employs two layers, with the first consisting of 16 units and the second consisting of 1 unit (or more based on class count). In addition, a 128-dimensional word embedding is employed.

The LSTM-CNN model incorporates two layers with 32 filters and a kernel size 5. A 1D max pooling layer is employed with a pool size of 3. In the final layer of the

TABLE 1: Parameters.

Parameter	Value
Dropout rate	0.25
Batch size	32
Maximum number of epochs	20
Learning rate (Adam optimizer)	0.001

LSTM-CNN model, two LSTM layers with 32 units are employed. The classifier employed two layers, with 16 and 1 unit (or more based on class count). In addition, a 128-dimensional word embedding is employed.

*4.1. Evaluation Criteria.* To assess the effectiveness of the proposed model and conduct a comparative analysis with prior research, we employ the following evaluation metrics: precision, recall, *F1*-score, and accuracy. These metrics are defined in equations (12)–(15). In these equations, TP stands for true positive, FP represents false positive, TN stands for true negative, and FN represents false negative [41, 42].

$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FN + FP}, \quad (12)$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100, \quad (13)$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100, \quad (14)$$

$$\text{F1 - Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \times 100. \quad (15)$$

*4.2. Dataset.* Five social media datasets are used to evaluate the efficacy of the proposed DTSC model. The datasets are collected from various social media platforms with two (positive, negative) and three (positive, negative, and neutral) classes. These datasets have been partitioned into training and testing sets with an 80 : 20 ratio. Table 2 presents the complete statistical information for each dataset.

*4.2.1. Getty Images.* Getty images (<https://www.gettyimages.com/> (accessed Mar. 16, 2023)) offer a vast collection of creative resources, including photographs with detailed textual descriptions, videos, and audio content, catering to businesses and consumers, with over 477 million resources in its collection. The main advantage of Getty images is its user-friendly, effective query-based search engine with formal yet informative image descriptions. In particular, 3244 adjective-noun pairs (ANPs) from the visual sentiment ontology [43] are used as keywords to collect 20,127 sentiment sentences with two classes (positive, negative), named the “Binary\_Getty” (BG) dataset, which includes textual explanations and labels.

The initial labeling is accomplished using the sentiment scores associated with ANP keywords. We further employ the Valence-Aware Dictionary and sentiment Reasoner

TABLE 2: The complete statistics of each dataset.

Dataset	Positive	Negative	Neutral	Total
BG	10098	10029	—	20127
BIS	10587	8692	—	19279
IMDB	25000	25000	—	50000
Twitter	6075	5228	5770	17073
MVSA-Single	2683	1358	470	4511

(VADER) [44], a lexicon and rule-based SA tool (<https://github.com/chute/vaderSentiment> (accessed Mar. 16, 2023)), to label the preprocessed textual description. Then, we select only the text samples for which ANP and VADER sentiment scores are the same. Finally, three volunteers were chosen to assess the quality of our datasets. Each sample is graded 1 (suitable) or 0 (unsuitable). The results show that 95% of the samples are suitable and 5% are unsuitable; we only consider the samples with grade 1 (suitable) and ignore the others.

*4.2.2. iStock Images.* iStock images (<https://www.istockphoto.com/> (accessed Mar. 16, 2023)) is an online platform that offers a wide range of international royalty-free microstock photos, including images and their accompanying textual descriptions, graphics, clipart, videos, and audio tracks. The same procedure from Getty images is implemented; 3244 ANPs are used as keywords to retrieve 19,279 sentiment sentences with two classes (positive, negative), named the “Binary\_iStock” (BIS) dataset. The dataset includes textual explanations and labels. The same labeling procedure demonstrated for the BG dataset is used to establish the final labeling of the BIS dataset.

*4.2.3. Twitter Dataset.* Additionally, we gathered a new dataset from Twitter. English tweets are specifically gathered using the Twitter streaming application programming interface (API) (<https://developer.twitter.com/en> (accessed Mar. 16, 2023)), with user-generated hashtags as keywords. We carefully filtered out all texts that were too short (less than five words) or too long (more than 100 words). We use VADER, a lexicon and rule-based SA tool, to speed up the labeling process to predict text sentiment polarity. Based on the projected sentiment polarity, the tweets are manually categorized into neutral, negative, and positive sentiment polarities. Finally, 17,073 high-quality tweets were obtained.

*4.2.4. Multiview Sentiment Analysis (MVSA) Dataset.* The MVSA-Single dataset [45] comprises 5129 image-text pairs extracted from Twitter. After displaying each pair to a single annotator, the annotator assigned one of three polarities (neutral, negative, or positive) to the image-text pair. Like [46], we first delete tweets with contradicting textual and visual labels. In cases where one modality is labeled as neutral while the other is labeled as positive or (negative), the ultimate polarity assigned to multimodal data is positive or (negative). Thus, we obtain a new MVSA-Single dataset with 4511 text-image pairs. Here, we used only the



textual data from this dataset and considered a benchmark dataset collected mainly from the Twitter website, to demonstrate the outstanding performance of our proposed model using different social datasets.

**4.2.5. IMDB Dataset.** The IMDB (<https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>) is an online database of information related to films, television series, podcasts, home videos, video games, and critical reviews. It includes 50,000 IMDB movie reviews and the binary sentiment of each movie review: positive or negative.

To prepare text data for sentiment analysis (SA), it undergoes the following preprocessing steps: (1) lowercase, which involves changing all text to lowercase. (2) Remove irrelevant information, including punctuation, special characters, hashtags, multiple spaces, URL references, stop words, and numbers. (3) Emoticon translation involves translating all emoticons into their respective terms.

**4.3. Experiment 1: The Effect of the Neighborhood Technique.** The proposed method's performance is compared with and without the neighborhood technique in this experiment to ascertain the impact of the neighboring technique on our approach. In this experiment, the first part includes only the original input in the network training phase without the neighboring technique and neighbors are ignored. In the second part, the number of neighbors is considered to be 2. The outcomes are shown in Table 3.

The proposed method with the neighboring technique outperforms the method without the neighboring mode. The use of the data neighbors enables the learning of better representations. The accuracy, precision, recall, and *F1*-score of the proposed model with using neighbors improve by an average of 1.20%, 1.52%, 1.42%, and 1.40% compared to without using neighbor modes. For the MVSA-Single dataset, using a neighbor has a more positive effect than other datasets. So, for the MVSA-Single dataset, the accuracy, precision, recall, and *F1*-Score of the proposed method improved by an average of 1.72%, 2.73%, 1.70%, and 2.63%, respectively. Using neighbors dramatically improves performance.

**4.4. Experiment 2: The Effect of Neighborhood Size.** In this experiment, to determine the effect of the neighborhood size in our approach, the performance of the proposed method is evaluated with different neighborhood sizes and batches. Our method is executed ten times for each dataset, and the average result is reported. The results are demonstrated in Table 4.

As demonstrated in Table 4, the efficacy of the proposed method increases with the increase in the neighborhood size. So, for sizes 2, 4, 8, 16, and 32, the average of all criteria has increased by 0.72%, 1.37%, 1.48%, 1.53%, and 1.54%, respectively, compared to size 0. Interestingly, all the criteria improve with the increase in neighbors. However, as the size of the neighborhood increases, the improvement percentage

decreases. For example, there is only a 0.03% difference between sizes 16 and 32.

Meanwhile, there is a 0.61% difference between sizes 2 and 4. The reason is quite apparent; with the increase in the size of the neighborhood, distant neighbors may also be selected. Although these distant neighbors are in the same class as the input sample, the text of the two samples may differ.

**4.5. Experiment 3: The Proposed Method vs. Baseline Models.** In this section, an evaluation of the proposed method and other established baseline methods is conducted across all datasets. The outcomes of the proposed method and other approaches are presented in Table 5.

Table 5 demonstrated that the LSTM model outperformed CNN and GRU, exhibiting comparatively inferior performance. The hybrid LSTM-CNN model demonstrated superior performance to the LSTM-only model, followed by bidirectional-based methods with relatively good performance, such as Bi-LSTM and Bi-GRU. The proposed method's average accuracy, precision, recall, and *F1*-score metrics are 92.77%, 93.73%, 91.82%, and 92.73%, respectively. On the second-best method (Bi-GRU), the average values for accuracy, precision, recall, and *F1*-score metrics are 92.37%, 93.32%, 91.42%, and 92.33%, respectively. According to CNN 1D's average recall of 90.54%, more samples from the specific class are typically misclassified. According to the suggested method's average recall of 91.82%, fewer samples are regularly misclassified compared to CNN 1D. As can be seen, the proposed model performs better than the current leading approach in all evaluation metrics. Specifically, it achieves the highest accuracy of 99.60% when evaluated on the BG dataset. This result provides strong evidence for the effectiveness of the DTSC model in enhancing the classifier's performance.

**4.6. Experiment 4: Generalization.** Our model has the benefit of handling scenarios where various types of metadata are accessible during training and testing with ease. Additionally, our model handles circumstances in which the words used over time may change. In other words, there may be some differences between the words in the training and test sets.

In the real world, the vocabulary or tags may change as new words become popular and older words fall out of favor. Any method that relies on user metadata should be able to handle these conditions. Ideally, to test our model's resilience to changes in user words over time, we should train it with texts from one time and test it with texts from another.

Instead of randomly dividing a dataset into training and test sets in this experiment, we use the BG dataset to train the model, generate neighborhoods in the training phase, and use the BIS dataset to test and generate neighborhoods in the testing phase. The results are reported in Table 6.

Table 6 presents a comparative analysis between the proposed method and baseline algorithms, namely, LSTM and GRU. The evaluation uses BG for training and BIS for testing in two distinct scenarios. The findings indicate that

TABLE 3: The effect of the neighborhood technique.

Neighboring	Metrics	Dataset				
		BG	BIS	MVSA-Single	Twitter	IMDB
Not taken into account	Accuracy	98.23	97.54	80.77	95.14	86.65
	Precision	98.46	98.29	80.20	95.18	89.45
	Recall	98.15	96.21	80.79	95.14	82.41
	<i>F1</i> -score	98.10	97.95	80.11	95.11	86.01
Taken into account	Accuracy	<b>99.60</b>	<b>98.32</b>	<b>82.19</b>	<b>96.13</b>	<b>87.60</b>
	Precision	<b>99.55</b>	<b>99.51</b>	<b>82.39</b>	<b>96.12</b>	<b>91.06</b>
	Recall	<b>99.65</b>	<b>97.54</b>	<b>82.16</b>	<b>96.13</b>	<b>83.64</b>
	<i>F1</i> -score	<b>99.60</b>	<b>98.52</b>	<b>82.22</b>	<b>96.12</b>	<b>87.20</b>

Bold values indicate that the proposed neighbourhood technique model has achieved better results.

TABLE 4: The effect of neighborhood size.

Neighborhood size	Metrics	Dataset				
		BG	BIS	MVSA-Single	Twitter	IMDB
0	Accuracy	98.23	97.53	80.74	95.17	87.01
	Precision	98.41	98.30	80.22	95.16	91.14
	Recall	98.16	96.21	80.74	95.15	83.29
	<i>F1</i> -score	98.10	97.95	80.10	95.11	87.00
2	Accuracy	98.74	98.05	82.28	95.65	87.15
	Precision	98.95	98.78	81.71	95.67	91.32
	Recall	98.67	96.73	82.28	95.66	83.78
	<i>F1</i> -score	98.62	98.46	81.62	95.63	87.12
4	Accuracy	99.23	98.53	82.77	96.15	88.25
	Precision	99.44	99.29	82.20	96.17	92.54
	Recall	99.18	97.24	82.78	96.16	84.34
	<i>F1</i> -score	99.10	98.97	82.10	96.13	88.31
8	Accuracy	99.34	98.65	82.87	96.24	88.36
	Precision	99.55	99.38	82.31	96.28	92.64
	Recall	99.27	97.33	82.87	96.24	84.47
	<i>F1</i> -score	99.22	99.06	82.23	96.22	88.40
16	Accuracy	99.37	98.72	82.94	96.31	88.40
	Precision	99.58	99.40	82.37	96.31	92.68
	Recall	99.31	97.40	82.93	96.32	84.50
	<i>F1</i> -score	99.25	99.10	82.26	96.27	88.44
32	Accuracy	99.42	98.73	82.96	96.33	88.45
	Precision	99.63	99.46	82.39	96.35	92.75
	Recall	99.35	97.41	82.96	96.34	84.12
	<i>F1</i> -score	99.30	99.14	82.30	96.31	88.12

the proposed methodology has yielded enhancements in performance, with average accuracy improvements of 1.14% and 1.09% observed for group one and respective improvements of 1.89% and 2.23% achieved for group two.

This suggests that leveraging additional metadata, specifically employing different vocabularies during training and testing, generates enhanced representations. Utilizing neighbors plays a crucial role in achieving these improvements. During the network training phase, this approach generates superior features characterized by a high level of generalizability. Even during the testing phase, when the input sample contains words not present in the training dictionary, the model produces superior representations with the assistance of its neighbors.

The results obtained from Tables 3 to 6 demonstrate the great benefits of the proposed model, which can be summarized as follows:

- (1) Improved accuracy: the model is designed to enhance accuracy in sentiment categorization, a task known to be challenging, especially in the context of social media data. By leveraging both textual content and extensive metadata, it aims to provide more accurate sentiment analysis.
- (2) Contextual understanding: unlike models solely relying on text data, this model considers the contextual aspects of social media posts. This contextual understanding is crucial for interpreting sentiment accurately, given the nuances and informal language often used in social media.
- (3) Utilizing metadata: the model harnesses the wealth of metadata, or neighboring data, available on platforms like Twitter, which includes information about posts and their authors. This additional

TABLE 5: Performance comparison of different models.

Model	Metric	Dataset				
		BG	BIS	MVSA-Single	Twitter	IMDB
CNN	CNN	Accuracy	98.30	97.02	80.86	94.83
		Precision	98.23	98.21	81.09	94.82
		Recall	98.35	96.24	80.86	94.83
		<i>F1-score</i>	98.30	97.22	80.92	94.82
GRU	GRU	Accuracy	98.34	97.06	80.90	94.89
		Precision	98.30	98.24	81.03	94.89
		Recall	98.37	96.27	80.80	94.90
		<i>F1-score</i>	98.32	97.24	80.86	94.88
LSTM	LSTM	Accuracy	98.40	97.12	80.96	94.93
		Precision	98.33	98.31	81.19	94.92
		Recall	98.45	96.34	80.96	94.93
		<i>F1-score</i>	98.40	97.32	81.02	94.92
LSTM-CNN	LSTM-CNN	Accuracy	98.85	97.57	81.41	95.38
		Precision	98.78	98.76	81.64	95.37
		Recall	98.90	96.79	81.41	95.38
		<i>F1-score</i>	98.85	97.77	81.47	95.37
Bi-LSTM	Bi-LSTM	Accuracy	99.10	97.82	81.66	95.63
		Precision	99.03	99.01	81.89	95.62
		Recall	99.15	97.04	81.66	95.63
		<i>F1-score</i>	99.10	98.02	81.72	95.62
Bi-GRU	Bi-GRU	Accuracy	99.20	97.92	81.76	95.73
		Precision	99.13	99.11	81.99	95.72
		Recall	99.25	97.14	81.76	95.73
		<i>F1-score</i>	99.20	98.12	81.82	95.72
DTSC	DTSC	Accuracy	<b>99.60</b>	<b>98.32</b>	<b>82.19</b>	<b>96.13</b>
		Precision	<b>99.55</b>	<b>99.51</b>	<b>82.39</b>	<b>96.12</b>
		Recall	<b>99.65</b>	<b>97.54</b>	<b>82.16</b>	<b>96.13</b>
		<i>F1-score</i>	<b>99.60</b>	<b>98.52</b>	<b>82.22</b>	<b>96.12</b>

TABLE 6: Cross-dataset evaluation results.

Group	Method	Training dataset	Testing dataset	Accuracy	<i>F1_score</i>
One	LSTM	BG (all data)	BIS (train set)	94.39	94.70
		BG (all data)	BIS (test set)	94.94	95.35
		BG (all data)	BIS (all data)	94.57	94.90
	GRU	BG (all data)	BIS (train set)	94.66	94.94
		BG (all data)	BIS (test set)	94.71	94.10
		BG (all data)	BIS (all data)	94.67	94.97
	DTSC	BG (all data)	BIS (train set)	<b>95.96</b>	<b>96.23</b>
		BG (all data)	BIS (test set)	<b>95.25</b>	<b>95.64</b>
		BG (all data)	BIS (all data)	<b>95.92</b>	<b>96.21</b>
Two	LSTM	BG (train set)	BIS (train set)	94.40	94.70
		BG (train set)	BIS (test set)	92.89	93.25
		BG (train set)	BIS (all data)	94.69	94.97
	GRU	BG (train set)	BIS (train set)	93.78	94.05
		BG (train set)	BIS (test set)	94.06	94.45
		BG (train set)	BIS (all data)	93.21	93.46
	DTSC	BG (train set)	BIS (train set)	<b>95.50</b>	<b>95.94</b>
		BG (train set)	BIS (test set)	<b>96.14</b>	<b>96.58</b>
		BG (train set)	BIS (all data)	<b>95.68</b>	<b>96.11</b>

Bold values indicate the best result.

context can significantly improve the accuracy of sentiment analysis.

- (4) Flexibility: the model’s nonparametric generation of text neighborhoods and subsequent use of a novel parametric model make it adaptable to various tasks.

It can handle complex sentiment analysis tasks that may be impractical for other techniques.

- (5) Cutting-edge performance: the model demonstrates a state-of-the-art level of performance in text sentiment classification. It outperforms traditional

methods by effectively combining contextual information, thereby offering a valuable advancement in sentiment analysis technology.

In summary, the model's benefits include improved accuracy, enhanced contextual understanding, effective utilization of metadata, flexibility in handling diverse tasks, and cutting-edge performance, all of which contribute to more precise sentiment categorization, especially in the challenging domain of social media analysis.

## 5. Conclusion

Some texts that are difficult to recognize on their own may become more understandable in a neighborhood of related texts with similar contexts. Motivated by this theory, a novel deep text sentiment classification (DTSC) model was proposed to improve the classifier's performance by integrating the neighborhood of related texts. Our model uses the nonparametric approach to construct neighborhoods of related texts based on Jaccard similarities.

Moreover, two distinct deep learning-based recurrent neural networks (Bi-LSTM and GRU) were integrated to extract sophisticated features, capture temporal relationships, and generate SA insights. The result of each module was further processed through the maximum operation, which selects the most pertinent data. Finally, the extracted features were concatenated and subjected to classification to achieve accurate sentiment prediction. In contrast to the previous studies, our approach utilizes a nonparametric approach, enabling it to perform strongly even when the text vocabulary varies between training and testing. The effectiveness of the proposed model was evaluated on five real-world sentiment datasets of short English text along with a dataset of lengthy movie reviews. The DTSC model performs more accurately and efficiently in identifying and understanding the semantics of both short and lengthy texts when compared to baseline approaches. The proposed model demonstrated a high level of accuracy across the datasets. Specifically, it achieved a 99.60% accuracy on the Binary\_Getty (BG) dataset, a 98.32% on the Binary\_iStock (BIS) dataset, a 96.13% accuracy on Twitter, an 82.19% accuracy on the multiview sentiment analysis (MVSA) dataset, and an 87.60% accuracy on the IMDB dataset. These findings indicate that the proposed model performs better than the existing state-of-the-art techniques regarding model evaluation criteria for text sentiment classification. Future works primarily comprise (1) broadening the model's scope to encompass additional languages, such as Persian, and (2) leveraging transformer-based language models to produce more resilient embedding representations.

## Data Availability

The following information was supplied regarding data availability: (1) the computer codes and datasets are available at GitHub: [https://github.com/israakhalafsalman/LNN\\_SANNECDM](https://github.com/israakhalafsalman/LNN_SANNECDM) and (2) IMDB dataset of 50K movie reviews: <https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] I. Khalaf, S. Al-Tameemi, M.-R. Feizi-Derakhshi, S. Pashazadeh, and M. Asadpour, "A comprehensive review of visual-textual sentiment analysis from social media networks," 2022, <https://arxiv.org/abs/2207.02160>.
- [2] N. Jing, Z. Wu, and H. Wang, "A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction," *Expert Systems with Applications*, vol. 178, Article ID 115019, 2021.
- [3] M. Haselmayer and M. Jenny, "Sentiment analysis of political communication: combining a dictionary approach with crowdcoding," *Quality and Quantity*, vol. 51, no. 6, pp. 2623–2646, 2017.
- [4] K. Chakraborty, S. Bhatia, S. Bhattacharyya, J. Platos, R. Bag, and A. E. Hassanien, "Sentiment Analysis of COVID-19 tweets by Deep Learning Classifiers—a study to show how popularity is affecting accuracy in social media," *Applied Soft Computing*, vol. 97, Article ID 106754, 2020.
- [5] E. Zafarani-Moattar, M. R. Kangavari, and A. M. Rahmani, "A comparative study on transfer learning and distance metrics in semantic clustering over the COVID-19 tweets," 2021, <http://arxiv.org/abs/2111.08658>.
- [6] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," *Proceedings of the 31st International Conference on International Conference on Machine Learning*, vol. 4, pp. 2931–2939, 2014.
- [7] R. Obiedat, R. Qaddoura, A. M. Al-Zoubi et al., "Sentiment analysis of customers' reviews using a hybrid evolutionary SVM-based approach in an imbalanced data distribution," *IEEE Access*, vol. 10, pp. 22260–22273, 2022.
- [8] A. A. Al-Absi, D. K. Kang, and M. A. Al-Absi, "Sentiment analysis and classification using deep semantic information and contextual knowledge," *Computers, Materials & Continua*, vol. 74, no. 1, pp. 671–691, 2022.
- [9] M. Asgari-Chenaghlu, M. R. Feizi-Derakhshi, L. farzinvas, M. A. Balafar, and C. Motamed, "TopicBERT: a cognitive approach for topic detection from multimodal post stream using BERT and memory-graph," *Chaos, Solitons & Fractals*, vol. 151, Article ID 111274, 2021.
- [10] S. Kumar, M. Badruddin Khan, M. Hoque Abul Hasanat, A. Khader Jilani Saudagar, A. AlTameem, and M. AlKhathami, "Sigmoidal particle swarm optimization for twitter sentiment analysis," *Computers, Materials & Continua*, vol. 74, no. 1, pp. 897–914, 2023.
- [11] S. Mishra, J. Diesner, J. Byrne, and E. Surbeck, "Sentiment Analysis with Incremental Human-In-The-Loop Learning and Lexical Resource Customization," in *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pp. 323–325, Guzelyurt, Cyprus, August 2015.
- [12] S. Mishra and J. Diesner, "Detecting the Correlation between Sentiment and User-Level as Well as Text-Level Meta-Data from Benchmark Corpora," in *Proceedings of the 29th ACM Conference on Hypertext & Social Media*, pp. 2–10, Guzelyurt, Cyprus, July 2018.
- [13] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, Article ID 107134, 2021.

- [14] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on the sentiment analysis," *Procedia Computer Science*, vol. 152, pp. 341–348, 2019.
- [15] A. Mee, E. Homapour, F. Chiclana, and O. Engel, "Sentiment analysis using TF-IDF weighting of UK MPs' tweets on Brexit," *Knowledge-Based Systems*, vol. 228, Article ID 107238, 2021.
- [16] S. Kunal, A. Saha, A. Varma, and V. Tiwari, "Textual dissection of live twitter reviews using naive Bayes," *Procedia Computer Science*, vol. 132, pp. 307–313, 2018.
- [17] H. Htet and Y. Myint, "Social media (Twitter) Data analysis using maximum entropy classifier on big data processing framework (Case study: analysis of health condition, education status, states of business)," *Journal of Pharmacognosy and Phytochemistry*, vol. 7, pp. 695–700, 2018.
- [18] M. Asgari-Chenaghlu, M. R. Feizi-Derakhshi, L. Farzinvas, M. A. Balafar, and C. Motamed, "CWI: a multimodal deep learning approach for named entity recognition from social media using character, word and image features," *Neural Computing & Applications*, vol. 34, no. 3, pp. 1905–1922, 2022.
- [19] Z. Jahanbakhsh-Nagadeh, M. R. Feizi-Derakhshi, and A. Sharifi, "A semi-supervised model for Persian rumor verification based on content information," *Multimedia Tools and Applications*, vol. 80, no. 28–29, pp. 35267–35295, 2021.
- [20] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, "Convolutional recurrent neural networks for text classification," *2019 International Joint Conference on Neural Networks (IJCNN)*, vol. 2019, 2019.
- [21] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "Abcdm: an attention-based bidirectional cnn-rnn deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.
- [22] Y. Kim, "Convolutional neural networks for sentence classification," 2014, <https://arxiv.org/abs/1408.5882>.
- [23] I. K. Salman Al-Tameemi, M.-R. Feizi-Derakhshi, S. Pashazadeh, and M. Asadpour, "Multi-model fusion framework using deep learning for visual-textual sentiment classification," *Computers, Materials & Continua*, vol. 76, no. 2, pp. 2145–2177, 2023.
- [24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [25] A. M. Alayba and V. Palade, "Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation," *Journal of King Saud University Computer and Information Sciences*, vol. 34, no. 10, pp. 9710–9722, 2022.
- [26] S. R. Ankita, S. Rani, A. K. Bashir, A. Alhudhaif, D. Koundal, and E. S. Gunduz, "An efficient CNN-LSTM model for sentiment detection in #BlackLivesMatter," *Expert Systems with Applications*, vol. 193, Article ID 116256, 2022.
- [27] I. K. S. Al-Tameemi, M. R. Feizi-Derakhshi, S. Pashazadeh, and M. Asadpour, "Interpretable multimodal sentiment classification using deep multi-view attentive network of image and text data," *IEEE Access*, vol. 11, pp. 91060–91081, 2023.
- [28] W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, "User reviews: sentiment analysis using lexicon integrated two-channel CNN-LSTM family models," *Applied Soft Computing*, vol. 94, Article ID 106435, 2020.
- [29] F. Abid, M. Alam, M. Yasir, and C. Li, "Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter," *Future Generation Computer Systems*, vol. 95, pp. 292–308, 2019.
- [30] C. N. Dang, M. N. Moreno-García, and F. De La Prieta, "Hybrid deep learning models for sentiment analysis," *Complexity*, vol. 2021, Article ID 9986920, 16 pages, 2021.
- [31] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, pp. 58080–58093, 2020.
- [32] M. Zulqarnain, S. Abd, R. Ghazali, N. Mohd, M. Aamir, and Y. Mazwin, "An improved deep learning approach based on variant two-state gated recurrent unit and word embeddings for sentiment classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, pp. 594–603, 2020.
- [33] L. Li, L. Yang, and Y. Zeng, "Improving sentiment classification of restaurant reviews with attention-based bi-gru neural network," *Symmetry*, vol. 13, no. 8, p. 1517, 2021.
- [34] M. Kamyab, G. Liu, A. Rasool, and M. Adjeisah, "ACR-SA: attention-based deep model through two-channel CNN and Bi-RNN for sentiment analysis," *PeerJ Computer Science*, vol. 8, p. e877, 2022.
- [35] A. Chiorrini, C. Diamantini, A. Mircoli, and D. Potena, *Emotion and Sentiment Analysis of Tweets Using BERT, EDBT, Ioannina, Greece, 2021*.
- [36] H. Huang, X. Y. Jing, F. Wu, Y. F. Yao, X. Y. Zhang, and X. W. Dong, "DCNN-BiGRU text classification model based on BERT embedding," in *Proceedings of the 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*, pp. 632–637, Shenyang, China, October 2019.
- [37] A. Bello, S. C. Ng, and M. F. Leung, "A BERT framework to sentiment analysis of tweets," *Sensors*, vol. 23, no. 1, p. 506, 2023.
- [38] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, Article ID 132306, 2020.
- [39] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019.
- [40] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: encoder-decoder approaches," 2014, <https://arxiv.org/abs/1409.1259>.
- [41] A. Golzari Oskouei, M. A. Balafar, and C. Motamed, "EDCWRN: efficient deep clustering with the weight of representations and the help of neighbors," *Applied Intelligence*, vol. 53, no. 5, pp. 5845–5867, 2022.
- [42] A. Golzari Oskouei, M. A. Balafar, and C. Motamed, "RDEIC-LFW-DSS: ResNet-based deep embedded image clustering using local feature weighting and dynamic sample selection mechanism," *Information Sciences*, vol. 646, Article ID 119374, 2023.

- [43] D. Borth, R. Ji, T. Chen, T. Breuel, and S. F. Chang, "Large-scale visual sentiment ontology and detectors using adjective noun pairs," *Proceedings of the 21st ACM international conference on Multimedia*, vol. 37, pp. 223–232, 2013.
- [44] E. Hutto and C. J. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," *Eighth International AAAI Conference on Weblogs and Social Media*, vol. 8, p. 18, 2014.
- [45] T. Niu, S. Zhu, L. Pang, and A. Elsaddik, "Sentiment analysis on multi-view social data," *MultiMedia Modeling*, vol. 9517, pp. 15–27, 2016.
- [46] N. Xu and W. Mao, "MultiSentiNet: a deep semantic network for multimodal sentiment analysis," *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, vol. 318, pp. 2399–2402, 2017.