

Deep Learning Framework for Sentiment Prediction using Residual Connections in Bidirectional – Gated Recurrent Unit

VARSHA NAGARAJU*, MUTHALAMBIKA CHALUVEGOWDA PADMA

*Department of Computer Science and Engineering, PES College of Engineering,
PES College Campus, Mandya, India*

**Corresponding author: nvarsha453@gmail.com*

(Received: 18 April 2025; Accepted: 26 November 2025; Published online: 12 January 2026)

ABSTRACT: Sentiment analysis plays an essential role in Natural Language Processing (NLP) for differentiating emotions and opinions expressed in various pieces of text. However, existing algorithms face challenges in handling complex language patterns and capturing long-term dependencies, thereby increasing overall computational cost. This research aims to design an improved sentiment analysis model that enhances accuracy and efficiency while addressing gradient-related limitations in deep networks. This research proposes a Residual Bidirectional Gated Recurrent Unit (RBi-GRU) algorithm for effective sentiment analysis, leveraging residual connections to improve accuracy and efficiency. Residual connections are incorporated into the Bi-GRU network to facilitate gradient flow across layers and mitigate the vanishing gradient problem during training. It also enables deeper networks by protecting data from earlier layers, which further enhances feature representation. Additionally, tokenization, stemming, and global vector-based word representations (GloVe) are employed during preprocessing to capture the semantic relationships and meanings of words, thereby improving contextual understanding in sentiment analysis. The developed RBi-GRU algorithm achieves 98.74% accuracy, 98.99% precision, 98.32% sensitivity, and 98.64% F1-score on the Sentiment140 dataset, compared with the Rectified Linear Unit-based Gated Recurrent Unit (ReLU-GRU).

ABSTRAK: Analisis sentimen memainkan peranan penting dalam Pemprosesan Bahasa Semula Jadi (NLP) bagi membezakan emosi dan pendapat yang dizahirkan dalam teks; namun, algoritma sedia ada menghadapi cabaran pengendalian corak bahasa yang kompleks serta kebergantungan jangka panjang, sekaligus meningkatkan masa pemprosesan. Kajian ini bertujuan mereka bentuk model analisis sentimen berketepatan tinggi dan cekap sambil menangani kekangan berkaitan kecerunan rangkaian mendalam. Sebuah algoritma Unit Kawalan Berulang Baki Dua Arah (RBi-GRU) dicadangkan dengan gabungan baki ke dalam rangkaian Bi-GRU bagi memudahkan aliran kecerunan antara lapisan dan mengurangkan masalah lenyap kecerunan semasa latihan, di samping membolehkan pembinaan rangkaian lebih mendalam dan meningkatkan perwakilan ciri. Selain itu, teknik prapemprosesan seperti penandaan token, pengakaran (stemming), serta penggunaan Representasi Kata Vektor Global (GloVe) diaplikasi bagi menangkap hubungan semantik dan makna kontekstual perkataan dengan lebih berkesan. Dapatan kajian eksperimen menunjukkan bahawa algoritma RBi-GRU mencapai ketepatan 98.74%, kejituan 98.99%, kepekaan 98.32%, dan skor F1 sebanyak 98.64% pada set data Sentimen140, sekaligus mengatasi prestasi model Unit Kawalan Berulang berasaskan Unit Pembetulan Linear (ReLU-GRU).

KEYWORDS: *Bidirectional – Gated Recurrent Unit, Global Vectors for Word Representation, Residual connection, Sentiment analysis, Stemming and Tokenization*

1. INTRODUCTION

The rapid growth of Natural Language Processing (NLP) techniques and the increasing availability of large text corpora have led to large-scale language models, enabling efficient application to social classification tasks [1-3]. Sentiment analysis, also known as opinion mining, is a crucial process in NLP that aims to identify opinion or emotional tones in text [4]. Sentiment analysis recognizes and classifies opinions in text as positive, neutral, and negative [5]. The primary aim of research in sentiment analysis is to classify text content into positive and negative categories based on emotional connotation [6,7]. Manual classification in sentiment analysis is time-consuming and requires multiple experts to label text [8], where Machine Learning (ML)-based algorithms become essential for online sentiment analysis [9]. Although these algorithms exhibit better performance, they are slow and time-consuming to train [10]. In recent times, the combination of ML and Deep Learning (DL)-based algorithms has become predominant in sentiment analysis due to their superior capability to handle non-linear relationships and learn intricate patterns [11,12]. DL-based algorithms, particularly neural networks, are well-suited to capturing nuanced sentiment expressions and learning discriminative features from the data [13]. ML-based algorithms like Support Vector Machine (SVM) and Naïve Bayes (NB) struggle to capture subtle contextual nuances, irony, and humor, which are crucial for NLP tasks. This limitation stems from their reliance on lexicons and hand-crafted features. In contrast, DL-based algorithms such as Recurrent Neural Networks (RNNs) and Gated Recurrent Units (GRUs) can model sequential information and track how sentiment evolves across text [14]. However, existing DL algorithms still face challenges, including handling complex language patterns, managing long-term dependencies, and being time-consuming. To overcome these drawbacks, this research proposes a DL-based algorithm designed to effectively capture intricate language patterns, maintain long-term dependencies, and operate more efficiently in less time.

Jacob and Viswanatham [15] introduced a Rectified Linear Unit-based Gated Recurrent Unit (ReLU-GRU) for sentiment analysis and classification. Initially, preprocessing was performed using stemming, tokenization, part-of-speech tagging, and punctuation removal. Next, Latent Dirichlet Allocation (LDA), Term Frequency-Inverse Document Frequency (TF-IDF), and bag-of-words (BoW) were introduced for feature selection. Next, classification was performed using the Improved Atom Search Optimizer (ASO) and Simulated Annealing (SA). Finally, classification was performed using a ReLU-GRU to select relevant features into different categories. The introduced method selected relevant features and enhanced classification performance. However, this method required more computational power, which is a major drawback for the model. Loh et al. [16] presented MPNet-GRU, a hybrid DL method that combined three major elements: Bi-GRU, GRU, and MPNet.

This transformer-based pre-trained language model improved language understanding through masked and permuted language modelling. MPNet captured long-term dependencies both unidirectionally and bidirectionally. Integrating these techniques yielded an efficient and effective solution for sentiment analysis. The presented method enabled sophisticated, context-aware sentiment analysis with high classification performance; however, it did not capture relationships among words, thereby affecting overall classification performance. Parveen et al. [17] developed a Gated Attention Recurrent Network (GARN) method for sentiment analysis. Initially, the dataset was preprocessed to clean and filter the data.

The term weight-enabled feature extraction using a Log Term Frequency-enabled Modified Inverse Class Frequency (LTF-MICF) method was employed to extract sentiment-based features. Then, a Hybrid Mutation-based White Shark Optimizer (HMWSO) was

implemented for feature selection. At the same time, a GARN architecture was integrated with a Recurrent Neural Network (RNN) and attention mechanisms for sentiment classification. The developed method minimized system complexity through improved performance, but did not address vanishing gradients during training. Onan [18] implemented a Semantic Role Labeling–Ant Colony Optimization (SRL-ACO) framework to generate additional training data for NLP techniques. The implemented framework used SRL to identify semantic words in a sentence, while ACO was used to generate new sentences and preserve these roles. The implemented algorithm improved accuracy by generating additional data without requiring manual data annotation. However, the implemented method exhibited high memory usage during sentiment analysis. Kumar et al. [19] suggested an Enhanced Vector Space Model (EVSM) with Hybrid Support Vector Machine (HSVM) for sentiment analysis.

The EVSM was assigned to categorize text by mapping it to high-dimensional vector spaces and extracting relationships among words and their contextual meanings. The Decision Tree (DT) algorithm was used with SVM to refine the selection process. The vector space model employed appropriate data-retrieval techniques, thereby improving classification accuracy. The suggested method did not capture sequential features in text. Rahman et al. [20] introduced a hybrid DL model, RoBERTa-BiLSTM, that integrates the Robustly Optimized BERT Pretraining Approach (RoBERTa) with a Bidirectional Long Short-Term Memory (BiLSTM). RoBERTa was used to generate meaningful word embeddings, while a BiLSTM effectively extracted contextual semantics from long dependency texts. The model leveraged the strengths of both sequential and transformer-based methods to improve performance in sentiment analysis.

The analysis indicates that these existing algorithms face the following drawbacks: limited computational power, failure to capture relationships among words, failure to address vanishing gradients, and high memory usage. To address these drawbacks, this study proposes the RBi-GRU method for effective sentiment analysis that employs residual connections. The glove word embedding technique is used in the preprocessing phase to capture semantic relationships and meanings among words, thereby improving context understanding for sentiment analysis. By incorporating residual connections into the Bi-GRU network, the flow of gradients across layers is enhanced, thereby mitigating the vanishing gradient problem. The significant contributions of this research are described as follows:

- Tokenization, stemming, and Global Vectors for Word Representation (Glove) techniques are used in the pre-processing stage to capture the semantic relationships and meanings of words and improve the context understanding for sentiment analysis.
- The proposed Residual - Bidirectional Gated Recurrent Unit (Rbi-GRU) algorithm is employed in the classification phase to capture the sequential features from data for high classification performance.
- Residual connections are incorporated into the Bi-GRU network to improve gradient flow, while mitigating the issue of vanishing gradients. It also equips deeper networks by protecting data from earlier layers, thereby enhancing feature representation for effective sentiment analysis.

The remainder of this research paper is organized as follows: Section 2 details the functioning of the developed algorithm; Section 3 presents the experimental results and compares the proposed algorithm with existing models; and Section 4 discusses the results and advantages of the proposed algorithm. Finally, Section 5 concludes the research study.

2. METHODOLOGY

This study introduces an effective DL-based algorithm for sentiment analysis using data from the Sentiment-140 dataset. This data is preprocessed using tokenization, stemming, and GloVe word embeddings, after which the preprocessed features are classified using the developed RBi-GRU method with residual connections. Figure 1 illustrates the functioning of the proposed model in sentiment analysis.

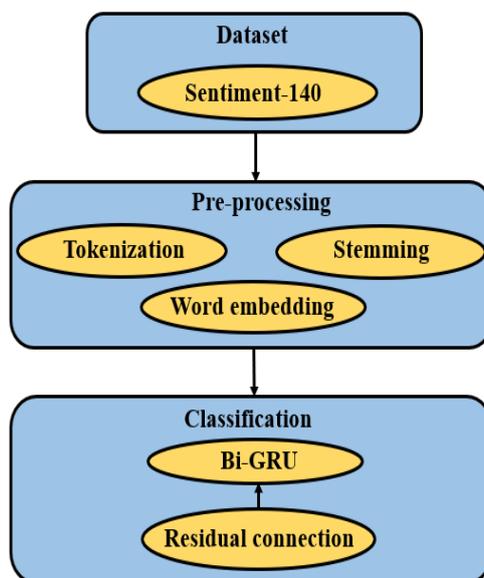


Figure 1. Process of sentiment analysis.

2.1. Dataset

The Sentiment140 dataset [21] comprises 1.6 million tweets, categorized as either detrimental or favorable. This allows one to view what individuals on Twitter perceive of a particular brand, subject, or product. The data is stored in a CSV file, without explicit emotion annotations. The dataset contains different components that define its file format, which are provided below. Figure 2 illustrates the data distribution.

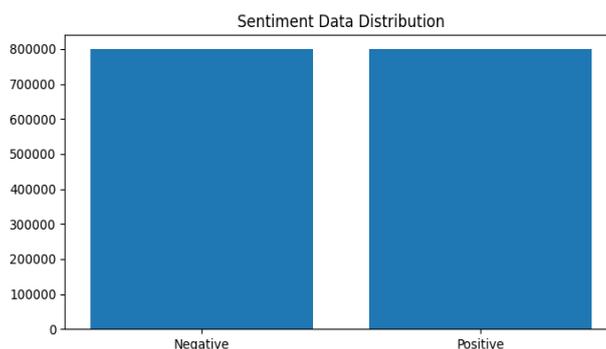


Figure 2. Data distribution.

- The polarity of the tweet (0 – negative, 2- neutral, and 4 – favourable)
- Tweet’s identifier
- Tweet’s creation data
- Question
- Tweet’s content

2.2. Pre-processing

The data is pre-processed using tokenization, stemming, and word embedding to improve data quality.

2.2.1. Tokenization

Tokenization involves splitting text into individual tokens or words. This process transforms unstructured text into a more structured format, enabling more effective analysis. By breaking down larger texts into smaller segments, tokenization facilitates the analysis of specific words and helps capture their semantic relationships. Unprocessed phrases and sentences are thus converted into smaller segments, called tokens, to enable analysis.

2.2.2. Stemming

Stemming is the process of reducing words to their root or base form. By converting related word variants to their base form, this process helps normalize textual data. For example, the words “running” and “ran” are both reduced to “run.” This approach groups related terms and preserves the core meaning of words, minimizing the influence of different word forms.

2.2.3. Word Embedding

Word embedding is the process of representing words as vectors based on a global co-occurrence matrix using GloVe. It uses an attenuation function based on the distance between two words in a context window to measure the weight.

2.3. Classification

This study employs a Bi-GRU with residual connections to extract and learn features from the data. The RBi-GRU combines the Bi-GRU layers with residual connections to enhance the neural network's effectiveness. The Bi-GRU is an RNN network that captures various patterns and sequential dependencies. GRU networks perform more effectively than traditional RNNs due to their gating mechanism, which enables the network to update and forget information from previous time steps. This mechanism includes updates and resets gates, which regulate the flow of information through the network. Figure 3 represents the architecture of the Bi-GRU model.

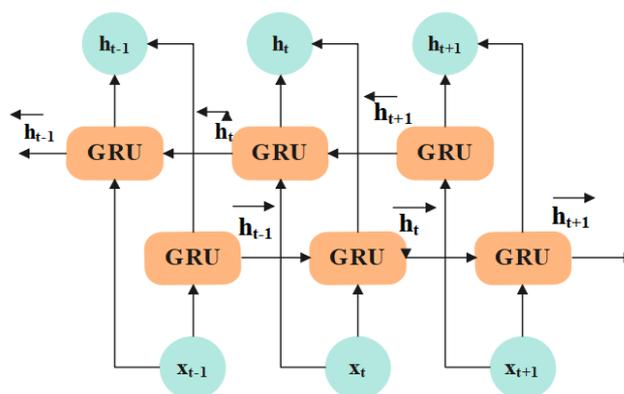


Figure 3. Architecture of Bi-GRU

The mathematical formula for update and reset gates is given in Eqs. (1) and (2),

$$u_t = \sigma(W_{hu}h_{t-1} + W_{xu}v_t + b_u) \quad (1)$$

$$r_t = \sigma(W_{hr}h_{t-1} + W_{xr}v_t + b_r) \quad (2)$$

In the above Eqs. (1) and (2), u_t represents the update gate at a time t , which acts as a control mechanism that regulates network updates to their internal state by utilizing data from the present input v_t and past hidden state h_{t-1} . The decision to update is influenced by a sigmoid activation function employed with the linear combination of hidden and input states, with respective weight matrices of W_{xu} and W_{hu} and bias b_u . The reset gate is represented as r_t which defines the degree to which the network forgets or rests in the hidden state, depending on the present input. The reset decision is controlled by the respective weight matrices W_{xr} and W_{hr} , where b_r is a biased term in the reset gate.

In Bi-GRU, two GRU units process an input sequence in both the forward and backward directions, allowing the network to incorporate past and future content at each time step. This is particularly used for ligand sequences, which are sequential and exhibit long-range dependencies. The outcome of Bi-GRU for the present time step is represented as h_t , acquired by combining the forward hidden state \vec{h}_t and backward hidden state \overleftarrow{h}_t . The mathematical expression for the Bi-GRU output is given in Eq. (3).

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (3)$$

The Bi-GRU processes text in both forward and reverse directions, enabling it to capture past and future contexts within a sentence. The forward direction \vec{h}_t processes the sentence from start to end, capturing how previous words influence later ones. The backward direction \overleftarrow{h}_t processes the sentence from the end to the beginning, capturing how later words provide context for earlier ones. Next, the forward and backward hidden states are combined, offering a comprehensive understanding of the sentence by considering information from both directions in the sequence.

Residual connections are incorporated into Bi-GRU layers to enable direct data transmission across layers. This improves the network's ability to handle deep architectures by mitigating the vanishing gradient problem and enhancing training efficiency. As a result, the method can capture complex relationships among words and phrases in deeper layers without losing significant contextual information. In long, multi-clause sentences, residual connections help retain the overall sentiment while deeper layers refine subtle nuances. Figure 4 illustrates the architecture of the Bi-GRU with residual connections.

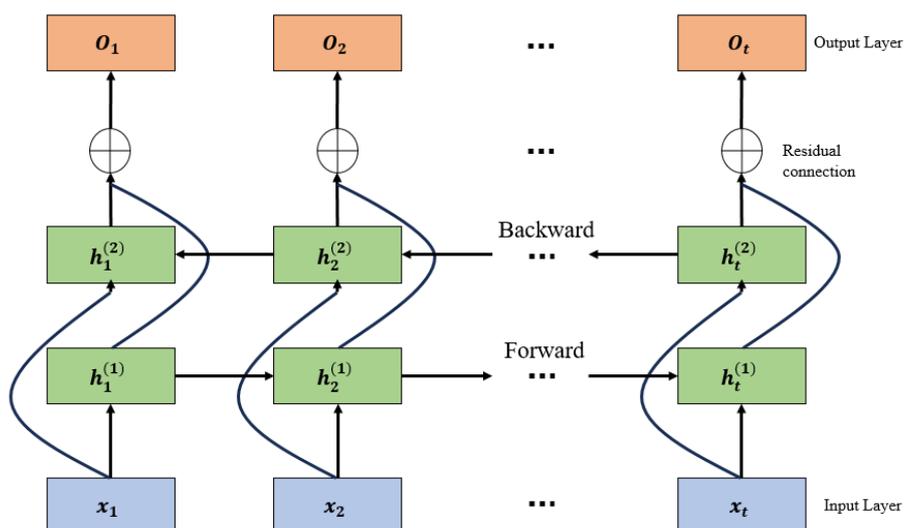


Figure 4. Architecture of Bi-GRU with residual connections

By measuring \vec{h}_t based on the candidate's hidden state \tilde{h}_t , the past hidden state and update gate are mathematically defined in Eq. (4), where \tilde{h}_t in one direction, the same is executed as \vec{h}_t in the opposite direction.

$$\vec{h}_t = (1 - u_t) \ominus h_{t-1} + u_t \ominus \tilde{h}_t \quad (4)$$

Where \tilde{h}_t is the candidate hidden state executed based on the reset gate r_t , h_{t-1} is the past hidden state and v_t is the present input vector, as mathematically formulated in Eq. (5).

$$\tilde{h}_t = \tanh(W_{hh}(r_t \ominus h_{t-1}) + W_{xh}v_t + b_h) \quad (5)$$

The RBi-GRU integrates a residual connection into the Bi-GRU layers. The residual connection enables data to be transferred directly between layers and to circulate through intermediate layers. This process avoids vanishing gradients that arise during training due to repeated gradient multiplication. Residual connection stacks multiple layers of Bi-GRU without accounting for the method's convergence time. The RBi-GRU supports skip connections, in which the input to the next layer is obtained by concatenating the result of the previous layer with the residual. The mathematical formula for the result of i th with residual connection is given in Eq. (6),

$$h_i = F(W[h_{i-1}, X] + b) + X \quad (6)$$

In the above Eq. (6), the h_i represents the result of i th the layer with residual connection, the X represents the output of the previous layer, the h_{i-1} represents the result of the past layer, the W represents a weight matrix, the b represents the bias vector and the F represents the non-linear activation function. Algorithm 1 represents the process of the proposed method.

Algorithm 1 – Process of the proposed method

<p>Input – Sentiment-140 dataset</p> <p>Output – classified output</p> <p>Split the dataset into a training set (80%) and a testing set (20%)</p> <p>Define hyperparameters – Epochs 50, batch size 64, learning rate 0.01, Adam optimizer, Categorical cross entropy loss function, 0.3 dropout rate, and 100 embedding dimensions.</p> <p>Initialize the RBi-GRU model with residual connections</p> <p>For every epoch from 1 to epochs do</p> <p style="padding-left: 20px;">For every batch in the training set</p> <p style="padding-left: 40px;">Forward propagate batch through embedding + RBi-GRU layers</p> <p style="padding-left: 40px;">Compute loss using the loss function</p> <p style="padding-left: 40px;">Backpropagate loss and update weights using the optimizer</p> <p style="padding-left: 20px;">End for</p> <p>Initialize Predictions = []</p> <p>For every sample x in test set, do</p> <p style="padding-left: 40px;">$y_{pred} = RBi - GRU(x)$</p> <p style="padding-left: 40px;">predictions.append(y_{pred})</p> <p style="padding-left: 20px;">End for</p> <p>End for</p>
--

3. EXPERIMENTAL ANALYSIS

The performance of the RBi-GRU technique is simulated in a Python environment, with configurations of 8 GB of RAM, an Intel Core i5 processor, and Windows 10 (64-bit). Performance metrics like accuracy, sensitivity, precision, and F1-score are considered to validate the performance of the developed RBi-GRU technique. The dataset is split into an 80% training set and a 20% test set. The mathematical expressions for performance metrics are given in Eqs. (7) and (8).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (7)$$

$$F1 - score = \frac{2 \times Sensitivity \times Precision}{Sensitivity + Precision} \times 100 \quad (8)$$

Table 1. Performance of RBi-GRU with different classes

Classes	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Positive	98.47	99.88	96.70	98.27
Negative	99.01	98.10	99.93	99.01
Average	98.74	98.99	98.32	98.64

Table 1 presents the performance of RBi-GRU across the dataset's classes, evaluated using various metrics. The developed RBi-GRU technique achieves 98.47% accuracy for the positive class, 99.01% for the negative class, and 98.74% overall.

In Table 2, the performance of the Bi-GRU without a residual connection is evaluated on the Sentiment140 dataset using various metrics. The RNN, LSTM, Bi-LSTM, and standard GRU are conventional algorithms used to evaluate the performance of the Bi-GRU. The Bi-GRU without a residual connection achieves 95.47% accuracy, 94.65% precision, 95.10% recall, and 94.87% F1-score on the sentiment140 dataset.

Table 2. Performance of Bi-GRU without the residual connection method

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
RNN	88.43	87.21	89.60	88.39
LSTM	90.87	89.43	91.25	90.34
Bi-LSTM	93.28	92.11	93.15	92.63
GRU	94.02	92.84	93.55	93.19
Bi-GRU	95.47	94.65	95.10	94.87

Table 3 presents the performance of the Bi-GRU with residual connections on the Sentiment 140 dataset, evaluated using various metrics. The RNN, LSTM, Bi-LSTM, and standard GRU are conventional algorithms used to evaluate the performance of the Bi-GRU with a residual connection. The Bi-GRU with a residual connection achieves 98.74% accuracy, 98.99% precision, 98.32% recall, and 98.64% F1-score on the sentiment140 dataset.

Table 3. Performance of Bi-GRU with the residual connection method

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
RNN	92.34	91.12	92.5	91.81
LSTM	94.21	93.18	94.5	93.84
Bi-LSTM	96.15	95.38	96.1	95.74
GRU	95.76	94.82	95.9	95.36
RBi-GRU	98.74	98.99	98.32	98.64

Figure 5 illustrates the memory usage of the developed Bi-GRU with residual connections, compared with the Bi-LSTM. Then, the total memory usage for each parameter, gradient, optimizer, and activation is evaluated. Here, it is observed that, compared with the Bi-LSTM, the Bi-GRU network consumes less memory.

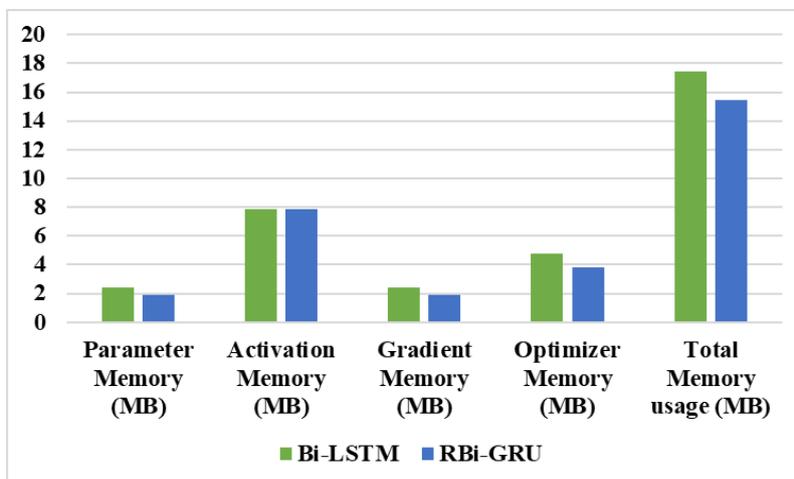


Figure 5. Memory usage of the developed RBi-GRU.

Table 4 presents an evaluation of different K-fold values across various performance metrics. The different k-fold values used are 2, 3, 5, 7, and 10, where, for a k value of 5, an accuracy of 98.74%, precision of 98.99%, recall of 98.32%, and f1-score of 98.64% is obtained, thereby proving superior to other k-fold values.

Table 4. Evaluation of K-fold values

K-fold values	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
2	94.12	93.85	94.10	93.97
3	95.14	94.56	95.00	94.72
5	98.74	98.99	98.32	98.64
7	96.85	96.24	96.56	96.40
10	97.20	96.78	97.00	96.88

Figure 6 presents a plot of accuracy versus epochs. In contrast, Figure 7 shows the loss vs. epochs plot, Figure 8 shows the ROC curve, and Figure 9 shows the confusion matrix, all for the Sentiment140 dataset.

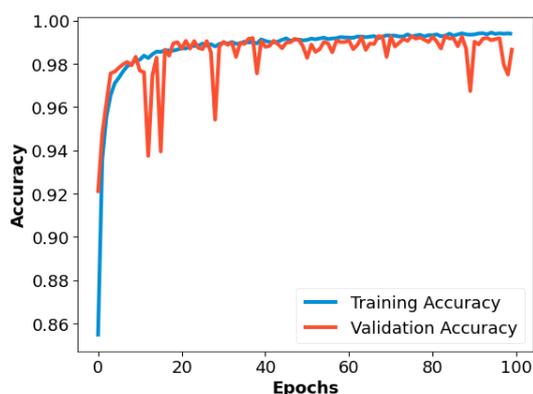


Figure 6. Accuracy vs epochs

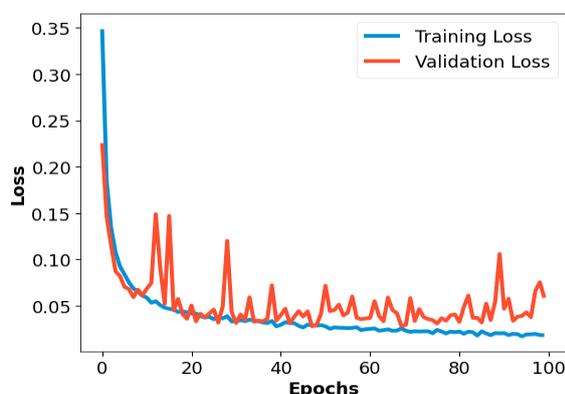


Figure 7. Loss vs epochs

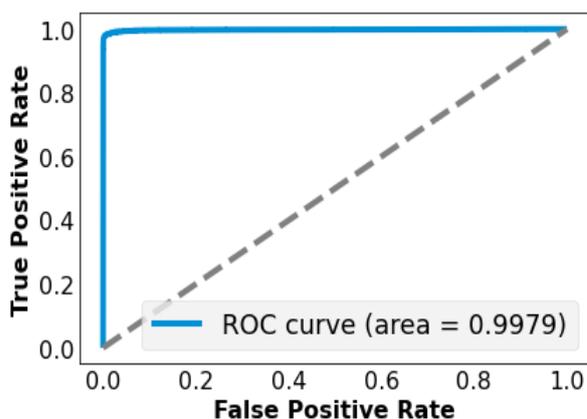


Figure 7. ROC curve

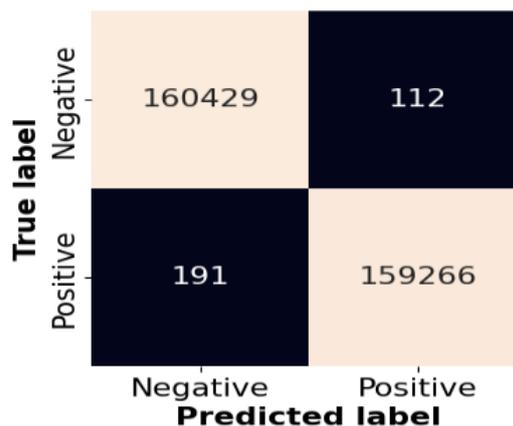


Figure 8. Confusion matrix

Table 5. Comparative Analysis of the RBi-GRU algorithm

Methods	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-score (%)
IASO-SA – GRU [16]	96.52	95.81	96.93	96.37
MPNet-GRU [17]	88.17	88	88	88
GARN [18]	97.86	96.65	96.76	96.70
RoBERTa-BiLSTM [20]	82.25	82.25	82.25	82.25
Proposed RBi-GRU	98.74	98.99	98.32	98.64

The performance of the developed RBi-GRU algorithm is compared with existing methods: IASO-SA–GRU [16], MPNet-GRU [17], GARN [18], and RoBERTa-BiLSTM [20] on the Senti140 dataset. The developed RBi-GRU algorithm achieves an accuracy of 98.74%, precision of 98.99%, sensitivity of 98.32%, and an F1-score of 98.64% on the Sentiment140 dataset. This model incorporates a residual connection with the Bi-GRU to enhance inter-layer connectivity and facilitate smooth gradient flow. This, in turn, improves the Bi-GRU model's training efficiency and classification performance. Table 5 presents a comparative analysis of the RBi-GRU algorithm.

4. DISCUSSION

The proposed algorithm operates in three phases and is applied to the Sentiment-140 dataset, with preprocessing and sentiment classification of the acquired data. The performance of this method is evaluated against the existing methods: RNN, LSTM, Bi-LSTM, and standard GRU on the Sentiment-140 dataset. The performance of the developed algorithm is compared with existing methods such as IASO-SA – GRU [16], MPNet-GRU [17], GARN [18], and RoBERTa-Bi-LSTM [20] on the Senti140 dataset. The analysis indicates that existing algorithms suffer from limited computational efficiency, an inability to capture relationships among words, failure to address vanishing gradients, and high memory consumption. To overcome these drawbacks, this study proposes the RBi-GRU method for effective sentiment analysis by incorporating a residual connection. The glove word embedding technique is used in the preprocessing phase to capture semantic relationships and meanings among words efficiently, thereby improving context understanding for sentiment analysis. Incorporating a residual connection into the Bi-GRU network enhances inter-layer gradient flow and mitigates the vanishing gradient problem. Despite its deep architecture, the RBi-GRU consumes less memory for parameters, activations, gradients, and optimizer states than the Bi-LSTM, making it a practical choice for large-scale sentiment analysis with limited computational resources.

The proposed RBi-GRU model also demonstrates improvements in sentiment analysis performance when compared to existing models. By incorporating residual connections into the Bi-GRU model, it efficiently addresses primary challenges in the recurrent neural network. These improvements enable the network to preserve significant contextual data across multiple layers, thereby improving understanding of nuanced sentiment expressions. Additionally, experimental results across multiple k-fold values demonstrate that the model maintains high stability and generalization ability with K=5, achieving the best performance-training stability balance. These findings show that residual connections in Bi-GRU improve overall prediction accuracy, minimize memory overhead, and enhance training convergence.

5. CONCLUSION

The DL-based algorithm is developed for effective sentiment analysis using the Sentiment-140 dataset. Tokenization, stemming, and GloVe word embeddings are employed during the preprocessing stage, thereby enhancing data quality and facilitating effective sentiment analysis. They also capture semantic relationships and meanings among words, improving context understanding. Then, classification is performed using the developed RBi-GRU algorithm, which includes a residual connection. The residual connection incorporated into the Bi-GRU network facilitates inter-layer gradient flow and mitigates the vanishing gradient problem. This process contributes to effective sentiment analysis. The developed RBi-GRU algorithm achieved 98.74% accuracy, 98.99% precision, 98.32% sensitivity, and 98.64% F1-score on the Sentiment-140 dataset. However, the loss of significant features reduced the effectiveness and overall performance of the developed algorithm. In the future, various DL-based algorithms with effective feature selection techniques will be developed to classify sentiments more accurately.

REFERENCES

- [1] Krishna MM, Duraisamy B, Vankara J (2023). Independent component support vector regressive deep learning for sentiment classification. *Meas.: Sens.*, 26:100678. <https://doi.org/10.1016/j.measen.2023.100678>
- [2] Havinga I, Marcos D, Bogaart P, Tuia D, Hein L (2024). Understanding the sentiment associated with cultural ecosystem services using images and text from social media. *Ecosyst. Serv.*, 65:101581. <https://doi.org/10.1016/j.ecoser.2023.101581>
- [3] Xu QA, Jayne C, Chang V (2024). An emoji feature-incorporated multi-view deep learning for explainable sentiment classification of social media reviews. *Technol. Forecasting Social Change*, 202:123326. <https://doi.org/10.1016/j.techfore.2024.123326>
- [4] Jain R, Kumar A, Nayyar A, Dewan K, Garg R, Raman S, Ganguly S (2023). Explaining sentiment analysis results on social media texts through visualization. *Multimedia Tools Appl.*, 82(15):22613-22629. <https://doi.org/10.1007/s11042-023-14432-y>
- [5] Petrescu A, Truica CO, Apostol ES, Paschke A (2024). EDSA-Ensemble: an event detection sentiment analysis ensemble architecture. *IEEE Trans. Affective Comput.*, 1-18. doi: 10.1109/TAFFC.2024.3434355
- [6] Halawani HT, Mashraqi AM, Badr SK, Alkhalaf S (2023). Automated sentiment analysis in social media using Harris Hawks optimisation and deep learning techniques. *Alexandria Eng. J.*, 80:433-443. <https://doi.org/10.1016/j.aej.2023.08.062>
- [7] Kaur G, Sharma A (2023). A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *Journal of big data*, 10(1):5. <https://doi.org/10.1186/s40537-022-00680-6>

- [8] Üveges I, Ring O (2023). HunEmBERT: a fine-tuned BERT-model for classifying sentiment and emotion in political communication. *IEEE Access*, 11:60267-60278. doi: 10.1109/ACCESS.2023.3285536
- [9] Barik K, Misra S, Ray AK, Bokolo A (2023). LSTM-DGWO-Based Sentiment Analysis Framework for Analyzing Online Customer Reviews. *Comput. Intell. Neurosci*, 2023(1):6348831. <https://doi.org/10.1155/2023/6348831>
- [10] Kastrati M, Kastrati Z, Shariq Imran A, Biba M (2024). Leveraging distant supervision and deep learning for twitter sentiment and emotion classification. *J. Intell. Inf. Syst*, 62(4):1045-1070. <https://doi.org/10.1007/s10844-024-00845-0>
- [11] Kumar T, Mahrishi M, Sharma G (2023). Emotion recognition in Hindi text using multilingual BERT transformer. *Multimedia Tools Appl*, 82(27):42373-42394. <https://doi.org/10.1007/s11042-023-15150-1>
- [12] Zou H, Wang Z (2023). A semi-supervised short text sentiment classification method based on improved Bert model from unlabelled data. *Journal of Big Data*, 10(1):35. <https://doi.org/10.1186/s40537-023-00710-x>
- [13] Zhou L, Zhang Z, Zhao L, Yang P (2023). Microblog sentiment analysis based on deep memory network with structural attention. *Complex Intell. Syst*, 9(3):3071-3083. <https://doi.org/10.1007/s40747-022-00904-5>
- [14] Yadav NB (2023). Harnessing customer feedback for product recommendations: an aspect-level sentiment analysis framework. *Hum.-Centric Intell. Syst*, 3(2):57-67. <https://doi.org/10.1007/s44230-023-00018-2>
- [15] Jacob N, Viswanatham VM (2024). Sentiment Analysis using Improved Atom Search Optimizer with a Simulated Annealing and ReLU based Gated Recurrent Unit. *IEEE Access*, 12:38944-38956. doi: 10.1109/ACCESS.2024.3375119
- [16] Loh NKN, Lee CP, Ong TS, Lim KM (2024). MPNet-GRUs: Sentiment Analysis with Masked and Permuted Pre-training for Language Understanding and Gated Recurrent Units. *IEEE Access*, 12:74069-74080. doi: 10.1109/ACCESS.2024.3394930
- [17] Parveen N, Chakrabarti P, Hung BT, Shaik A (2023). Twitter sentiment analysis using hybrid gated attention recurrent network. *Journal of Big Data*, 10(1):50. <https://doi.org/10.1186/s40537-023-00726-3>
- [18] Onan A (2023). SRL-ACO: A text augmentation framework based on semantic role labeling and ant colony optimization. *Journal of King Saud University-Computer and Information Sciences*, 35(7):101611. <https://doi.org/10.1016/j.jksuci.2023.101611>
- [19] Suresh Kumar K, Radha Mani AS, Ananth Kumar T, Jalili A, Gheisari M, Malik Y, Chen HC, Jahangir Moshayedi A (2024). Sentiment Analysis of Short Texts Using SVMs and VSMs-Based Multiclass Semantic Classification. *Applied Artificial Intelligence*, 38(1):2321555. <https://doi.org/10.1080/08839514.2024.2321555>
- [20] Rahman, M.M., Shiplu, A.I., Watanobe, Y. and Alam, M.A., 2025. RoBERTa-BiLSTM: A context-aware hybrid model for sentiment analysis. *IEEE Transactions on Emerging Topics in Computational Intelligence*.
- [21] Sentiment-140 dataset: <https://www.kaggle.com/datasets/kazanova/sentiment140>.