Scientific Journal of Informatics
Vol. 12, No. 4, November 2025
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Improving Sentiment Analysis with a Context-Aware RoBERTa–BiLSTM and Word2Vec Branch

Wahyu Hardyanto[1], Nila Prasetya Aryani[2], Defin Andestian[3], Sugiyanto[4], Wahyu Setyaningrum[5], M Fadil Mardiansyah[6], Muhamad Anbiya Nur Islam[7], Aji Purwinarko[8]

[1,2,4,]Study Program of Physics, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia
[3,5,6,7,8]Study Program of Informatics Engineering, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

Abstract.
Purpose: We improve the accuracy of Twitter/X sentiment analysis with a hybrid model combining Word2Vec and the Robustly Optimized BERT Pretraining Approach (RoBERTa). However, Twitter/X text is noisy (slang/OOV) and ambiguous, so the performance of the pre-trained transformer decreases. Word2Vec is also limited to local contexts. Integrative studies of both are still limited. The idea is that Word2Vec is strong for slang/novel vocabulary (distributional semantics), while RoBERTa excels in contextual meaning; combining the two mitigates each other's weaknesses.
Methods: The Sentiment140 dataset contains 1.6 million balanced tweets. The split is stratified; Word2Vec is trained solely on the training data. RoBERTa is pretrained (frozen in the first stage, then fine-tuned with some layers in the second stage). The Word2Vec and RoBERTa vectors are concatenated and processed using Bidirectional Long Short-Term Memory (BiLSTM) with sigmoid activation. Training utilizes TensorFlow and the Adam optimizer, incorporating dropout and early stopping. The decision threshold is optimized during the validation process.
Result: The hybrid model achieved an accuracy of 88.09%, an F1-score of 88.09%, and an Area Under the Curve (AUC) ≈ 95.19% on the Receiver Operating Characteristic (ROC). No overfitting was observed, and the hybrid model outperformed both single baselines. The confusion matrix and ROC curve corroborate the findings.
Novelty: The novelty lies in the fusion of distributional and contextual representations with a structured fusion mechanism. Limitations: Computational requirements and hyperparameter tuning are not yet extensive. Further directions: Systematic hyperparameter search and cross-validation across other large sentiment datasets to assess generalization.

Keywords: Sentiment analysis, BiLSTM, Word2Vec, RoBERTa
Received November 2025 / **Revised** December 2025 / **Accepted** December 2025

This work is licensed under a Creative Commons Attribution 4.0 International License.

INTRODUCTION

In the increasingly connected digital era, social media platforms like X (formerly known as Twitter) are widely used by everyone, where people post their opinions through tweets on products, current trends, politics, services, and communications [1]. Social media become a big data warehouse that provides detailed information from millions of individuals [2]. This data can be used to gain insights through sentiment analysis techniques. Sentiment analysis is a technique that involves the use of machine learning (ML) algorithms for Natural Language Processing (NLP) that allows the identification and classification of people's feelings (primarily) in large amounts of text data and determines whether the text expresses positive, negative, or neutral sentiments [3], [4]. Sentiment analysis using social media such as Twitter/X is a relatively inexpensive and potentially powerful complement to traditional survey methods, which are expensive and have limited sample sizes [5]. This kind of analysis has been widely used in various institutions, such as universities, business industries, and politicians, to provide marketing services, customer service, maintenance services, and user feedback [3], [6].

[1]*Corresponding author.
Email addresses: hardy@mail.unnes.ac.id (Hardyanto), nilaprasetya@mail.unnes.ac.id (Aryani), andestian107@gmail.com (Andestian), sugiyanto@mail.unnes.ac.id (Sugiyanto), wahyusetyaningrum27@students.unnes.ac.id (Setyaningrum), fadilmardiansyah@students.unnes.ac.id (Mardiansyah), fadilmardiansyah@students.unnes.ac.id (Islam), aji.purwinarko@mail.unnes.ac.id (Purwinarko)
DOI: 10.15294/sji.v12i4.35918

As technology advances, deep learning-based approaches are starting to replace traditional methods such as Naïve Bayes and Support Vector Machine (SVM) because of their ability to extract features and capture complex semantic relationships without the need for manual feature engineering [7]. Distributional models such as Word to Vector (Word2Vec) have been widely used in text representation because they can capture meaning based on the proximity of word contexts [8], [9]. On the other hand, transformer-based contextual models such as RoBERTa offer a deeper understanding of the meaning of words in the context of the whole sentence [10].

Several previous studies have investigated Twitter/Sentiment140 sentiment analysis using both traditional and hybrid methodologies. [1] utilized a TF-IDF feature-based machine learning methodology that incorporated various classifiers, such as Multinomial Naive Bayes, Logistic Regression, and Gradient Boosting, which were amalgamated into a Voting Classifier ensemble to improve performance beyond a singular baseline. [11] tested deep learning-based sequential models (RNN, Bi-RNN, and LSTM) on a large-scale Twitter dataset (1.6 million balanced tweets). They found that LSTM was the best architecture because it could capture long-term dependencies and deal with vanishing gradients. This is different from manual feature-based approaches. In a more modern direction, [12] proposed a hybrid RoBERTa–BiLSTM model that combines Transformer contextual representation with BiLSTM's ability to handle long dependencies; Evaluations on IMDb, Twitter US Airline, and Sentiment140 show improvements over baselines such as BERT/RoBERTa-based and RNN variants (GRU/LSTM). Most of these studies, conversely, examine only a singular representation paradigm (TF–IDF features, distributional embedding, or contextual embedding). This means that there is still room for more research on how to combine distributional and contextual representations in a single, structured hybrid framework, especially for dealing with Twitter/X-specific noise and slang/OOV.

However, most previous approaches use only one type of text representation, either distributional like Word2Vec [13] or contextual like Robustly Optimized BERT Pretraining Approach (RoBERTa) [14], thus missing the synergistic potential of combining the two. This study proposes a hybrid approach combining trainable Word2Vec with RoBERTa to form richer, more informative text representations. The model uses the Bi-directional Long Short-Term Memory (BiLSTM) architecture, which effectively processes data sequences and captures the bidirectional context in sentences [15], [16].

The Sentiment140 dataset, consisting of 1.6 million English tweets with a balanced distribution of positive and negative sentiments, is used as training data [17]. This dataset offers advantages in terms of volume, topic diversity, and sentiment label annotation that has been automatically provided through emoticon symbols [18]. Using this large-scale dataset, the model is expected to be more robust to natural language variations commonly used in social media.

While RoBERTa offers robust contextual representation, Twitter/X texts have domain-specific characteristics—such as slang, acronyms, noisy tokens, and OOV—that are not always well represented by generic pretrainers. On the other hand, Word2Vec trained locally on the target corpus can capture domain-specific distributional regularities, but does not model the global context of sentences. A research gap arises because there are still limited studies that combine these two paradigms into a structured hybrid framework, so the potential distributional–contextual synergy has not been optimally exploited. Based on this, we propose a Word2Vec–RoBERTa hybridization to enhance robustness to informal/OOV) tokens, with a more efficient adaptation strategy achieved by freezing most of RoBERTa and training lightweight Word2Vec branches.

The main contributions of this research include: (1) proposing a two-branch hybrid architecture that integrates distributional embedding (local Word2Vec) and contextual embedding (pretrained RoBERTa) in one coherent sentiment classification framework; (2) applying end-to-end trainable Word2Vec branches to adapt informal/OOV vocabulary representations to the Twitter/X domain, and using a staged partial fine-tuning strategy on RoBERTa (freeze → partially unlock the last layer) for domain adaptation without full fine-tuning; (3) designing a structured feature fusion mechanism through dimension alignment (128-dim per branch), fusion (concatenation and/or gated fusion), and a BiLSTM/MLP-based classification head to capture both local semantic signals and the global context of sentences.

METHODS

Proposed Method

Figure 1 illustrates how a hybrid model that utilizes both Word2Vec and RoBERTa is effective for binary sentiment classification. First, the tweet data is cleaned up by removing links, mentions, symbols, and punctuation. Then, it is divided into training, testing, and validation datasets. In the Word2Vec branch, a tokenizer is used to map the text. The Word2Vec model is then trained only on the training data to create an embedding matrix. A layered BiLSTM processes the embedding sequence with 128 units (with return_sequences=True) followed by 64 units. Then, the Normalization Layer normalizes it and projects it into a 128-dimensional feature vector. In parallel, the RoBERTa branch tokenizes the text (according to the specified configuration), passes the input to the RoBERTa encoder to obtain contextual representations, and then summarizes these representations using the pooling mode specified in the experiment configuration (e.g., BiLSTM pooling/mean/CLS). The results of the RoBERTa branch are then projected into a 128-dimensional space and normalized. The two 128-dimensional vectors are then combined using a fusion mechanism (see the Fusion Mechanism subsection) and processed by the classification head (Dense–Dropout–Sigmoid) to generate sentiment probabilities. The decision threshold is determined based on predictions made on the validation data (threshold optimization), and the final evaluation is performed on the test data using accuracy, F1 score, ROC–AUC, and a confusion matrix.

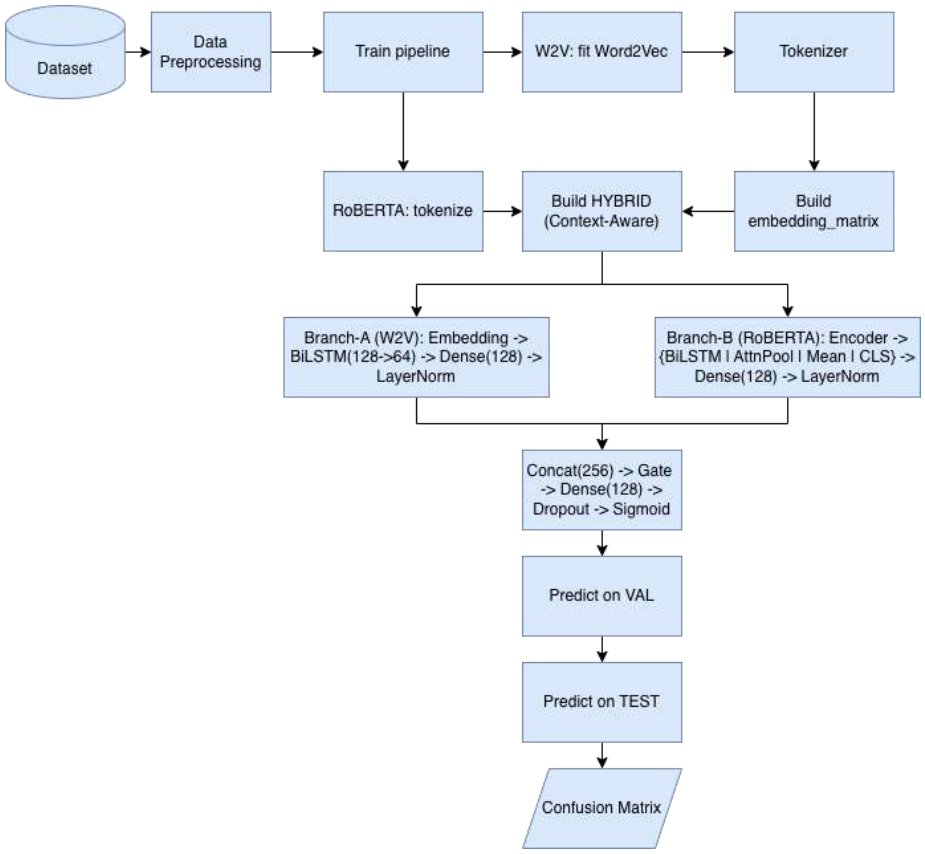

Figure 1. The proposed hybrid model architecture for Twitter/X sentiment analysis.

The pipeline includes data preprocessing, local Word2Vec training, RoBERTa tokenization, two modeling branches (Word2Vec–BiLSTM and RoBERTa–BiLSTM), feature fusion (concatenation and gated fusion), and binary classification with decision threshold optimization on validation data

Fusion Mechanism

After each branch produces feature vectors of the same dimension, namely $f_{w2v} \in \mathbb{R}^{128}$ from the Word2Vec branch and $f_{rb} \in \mathbb{R}^{128}$ from the RoBERTa branch, they are combined through concatenation. Afer concatenating the feature vektors from both branches, the resulting representation can be formally expressed through the following Equation 1, which defines the combined feature space:

$$z = [f_{w2v}; f_{rb}] \mathbb{R}^{256} \quad (1)$$

To weigh the feature contributions from both branches, we apply gated fusion to the concatenation vector. Briefly, the gate is calculated as Equation 2:

$$g = \sigma(W_g z + b_g) \quad (2)$$

and the fusion output is shown in Equation 3:

$$z' = z \odot g \quad (3)$$

where $\sigma(\cdot)$ is the sigmoid function and $\odot$ is the element-by-element multiplication operation. The vector z' then serves as the input to the classification head (Dense(128)–Dropout–Sigmoid). If the experimental configuration disables the gate, then $z' = z$ (fusion uses only concatenation).

Training Configuration & Fine-tuning Strategy

We trained with TensorFlow, using a batch size of 32 and a maximum sequence length of 128. Using Adam/AdamW, optimization was performed with early stopping based on validation metrics (such as val_auc) and restoring the best weights. We used dropout on the BiLSTM layer (0.5) and the classification head (0.3). We employed a staged fine-tuning scheme for RoBERTa to stabilize it and avoid full fine-tuning from the outset. In the first stage (Stage-1), the RoBERTa parameters were set in stone, and the training focused on the Word2Vec branch and the classification head, which had a higher learning rate. In the second stage (Stage-2), the last K encoder blocks of RoBERTa were unfrozen, and partial fine-tuning was performed using a lower learning rate. This scheme aims to achieve domain adaptation at a more manageable fine-tuning cost than unfreezing all RoBERTa parameters from the start.

Final Configuration Used for Reported Results

To ensure reproducibility, all metrics reported in the results section use the following final configuration: roberta-base pretrained transformer model, maximum sequence length of 128, and RoBERTa branch using BiLSTM pooling mode (RoBERTa hidden states → BiLSTM(128→64) → Dense(128) → LayerNorm). The Word2Vec branch uses a 100-dimensional embedding trained on the training data, then processed by BiLSTM(128→64) and projected into a 128-dimensional vector. Feature fusion is performed via concatenation (256 dimensions) and gated fusion before the classification head. The decision threshold does not use the default value of 0.5, but is optimized on the validation data by finding the threshold that maximizes the F1 score. Then, the best threshold is used to test the data.

Dataset Description

This study uses the Sentiment140 dataset, which has been widely used in social media-based sentiment analysis studies. This dataset was developed by [19] and contains 1,600,000 English-language tweets taken from X between April and June 2009. With such a large amount of data, this dataset provides the advantage of tweet topic diversity, covering various themes such as product opinions, news responses, social comments, and personal expressions. The dataset structure consists of six columns: target, id, date, flag, user, and text. Table 1 shows the Column Structure of the Sentiment140 Dataset. Annotated data (0 = negative, 4 = positive) can be used to detect sentiment, while text is cleaned from symbols, URLs, and punctuation as part of the pre-processing process. This dataset has a balanced class distribution, with 800,000 positive tweets and 800,000 negative tweets. In this study, we split the dataset in a training and testing ratio of 80–20, thus using 1,280,000 tweets to train the model and 320,000 tweets to test [11].

Table 1. Sentiment140 dataset column structure

No	Column Name	Description
1	target	Sentiment label (0 = negative, 4 = positive)
2	id	Unique ID of the tweet
3	date	Tweet timestamp (time and date format)
4	flag	Additional query or metadata (always NO_QUERY)
5	user	Twitter account username
6	text	Tweet text content

Word2Vec

The word representation process is carried out using the Word2Vec technique, one of the popular methods for producing word embeddings based on distributional learning. Meanwhile, word embeddings aim to convert text words into vectors for word representation, capturing semantic and syntactic relationships between words [20]. Word2Vec is trained using a collection of tweets that have gone through a preprocessing stage, and each sentence is converted into a list of tokens that aim to separate each word so that it can be included in the embedding model. The Word2Vec used is trained directly on the available

data, allowing the model to adjust the representation of words based on the specific context in the tweet corpus.

In this study, Word2Vec was built using the gensim library, with main parameters such as vector_size=100, window=5, and min_count=5, which respectively set the dimensions of the word vector, the size of the context window and the minimum number of occurrences of a word to be included in the model. The word vectors generated from Word2Vec are arranged into an embedding matrix to initialize the embedding layer in the deep learning model and trained end-to-end. Each token of the text that the Tokenizer has previously mapped will be represented by the corresponding Word2Vec vector. This embedding layer then becomes the input for the BiLSTM layer, allowing the model to capture sequence-based semantic information.

RoBERTa

RoBERTa is used to extract contextual information from the text as a sub-word in a document [21], [22]. RoBERTa is an extension of the BERT model that is trained more extensively with a larger data volume, longer training time, and a more flexible dynamic masking technique [23]. The text processing process by RoBERTa begins with tokenization using the built-in RoBERTa tokenizer, which converts each raw text into a series of input IDs and attention masks. Input IDs are numeric representations of text tokens, while attention masks indicate which tokens the model should pay attention to during inference. These two components are then fed into the TFRoBERTa Model pre-trained on a large corpus.

Algorithm 1 RoBERTa-based Contextual Representation (sequence/pooled per *mode*)

Require: Text dataset $T = \{t_1, \ldots, t_n\}$, tokenizer $\mathcal{T}$, RoBERTa $\mathcal{R}$, mode $m \in \{\texttt{bilstm}, \texttt{attnpool}, \texttt{mean}, \texttt{cls}\}$

Ensure: Sequence outputs $X = \{x_1, \ldots, x_n\}$; optional pooled vectors $E = \{e_1, \ldots, e_n\}$ for $m \neq \texttt{bilstm}$

for each text $t_i \in T$ **do**
  $(input\_ids_i, attention\_mask_i) \leftarrow \mathcal{T}(t_i)$
  $H_i \leftarrow \mathcal{R}(input\_ids_i, attention\_mask_i)$
  $x_i \leftarrow H_i.last\_hidden\_state$ ▷ $x_i \in \mathbb{R}^{L\times 768}$ (sequence)
  if $m = \texttt{attnpool}$ **then**
    $e_i \leftarrow \mathrm{AttnPool}(x_i)$ ▷ $e_i \in \mathbb{R}^{768}$
  else if $m = \texttt{mean}$ **then**
    $e_i \leftarrow \mathrm{GAP}(x_i)$ ▷ global average pooling
  else if $m = \texttt{cls}$ **then**
    $e_i \leftarrow x_i[0]$ ▷ CLS slice
  else
    $e_i \leftarrow \mathrm{None}$ ▷ pure sequence for BiLSTM
  end if
end for
return (X, E)

Figure 2. RoBERTA algorithm

In the proposed architecture shown in Figure 2, RoBERTa generates a contextual representation of each token through its last hidden state output. This representation is then treated according to the selected mode: in $bilstm$ mode, the previous hidden state sequence is passed as is to the BiLSTM layer to capture sequential patterns; while in $attnpool$, $mean$, or cls modes, the sequence is condensed into a single fixed vector per text via attention pooling, global average pooling, or first token capture. The vectors or sequences from the RoBERTa branch are then combined with those from the Word2Vec branch via feature pooling (with the gated fusion option) and processed by an MLP-based classification head. In this way, the RoBERTa component provides a rich semantic understanding of the sentence context. At the same time, the other branches and fusion mechanisms refine the signal to improve the accuracy of sentiment analysis.

In detail, the Figure 2 can be explained as follows: given a text dataset $T = \{t_1, \ldots, t_n\}$, each text t_i is tokenized by the RoBERTa tokenizer $\mathcal{T}$ to produce two components: $input\_ids$ and $attention\_mask$. These two components are fed into the RoBERTa model $\mathcal{R}$ to obtain the final representation of each token, written as $x_i = H_i^{last} \in \mathbb{R}^{L\times 768}$ where L is the length of the sequence after padding or truncation.

The way x_i is summarized is determined by the mode $m \in \{bilstm, attnpool, mean, cls\}$. In bilstm mode, the sequence x_i is not pooled and is passed as is as sequential input to the next stage, so no vector e_i is generated. In $attnpool$ mode, the sequence x_i is condensed into a single vector $e_i \in \mathbb{R}^{768}$ using attention pooling, which gives more weight to important tokens. In mean mode, global average pooling is used to obtain $e_i \in \mathbb{R}^{768}$. In cls mode, the first token vector is taken as the sentence summary, i.e. $e_i = x_i[0] \in \mathbb{R}^{768}$.

The output of the algorithm is a pair (X, E), where $X = \{x_i\}$ contains the sequence of last hidden states for each text, and $E = \{e_i\}$ contains the pooling result vector for $attnpool$, $mean$, or cls modes, and is empty in $bilstm$ mode. This (X, E) pair can be used in the next stage, for example, for sequential feature extraction, fusion with other branches, or classification.

BiLSTM

BiLSTM is used to process text representations generated from Word2Vec and RoBERTa. BiLSTM is a development method of LSTM that processes the sequence of data in sequential data back and forward so that the model can understand the context of words not only from the previous words but also from the words after them in the sentence [24], [25]. This process is critical in sentiment analysis tasks, where the meaning of a word can depend on the context of the sentence as a whole. Based on Figure 3, the input vector sequence from Word2Vec ($x_i \in \mathbb{R}^{L \times H}$) is first passed through a BiLSTM layer with the number of units ($u_1 = 128$) and setting return_sequences = True so that the output remains the sequence ($h_i^{(1)}$). This output is subjected to dropout at a rate ($p = 0.5$) to reduce overfitting, resulting in (d_i). Next, (d_i) is processed by a second BiLSTM with ($u_2 = 64$) which returns the final vector ($h_i^{(2)}$). This vector is then projected through a Dense layer of size 128 with ReLU activation to obtain (z_i), and normalized using LayerNorm to produce the final features ($f_i \in \mathbb{R}^{128}$). This feature set ($\{f_i\}$) is used in the next classification stage.

Algorithm 2 BiLSTM-based Feature Extraction (with projection & layer norm)

Require: Sequence set $X = \{x_i\}$ with $x_i \in \mathbb{R}^{L \times H}$, BiLSTM units (u_1, u_2), dropout rate $p = 0.5$
Ensure: Feature vectors $F = \{f_i\}$ with $f_i \in \mathbb{R}^{128}$
for each sequence $x_i \in X$ **do**
  $h_i^{(1)} \leftarrow \text{BiLSTM}(x_i, \text{units} = u_1, \text{return\_seq} = \text{True})$
  $d_i \leftarrow \text{Dropout}(h_i^{(1)}, \text{rate} = p)$
  $h_i^{(2)} \leftarrow \text{BiLSTM}(d_i, \text{units} = u_2)$ ▷ $h_i^{(2)} \in \mathbb{R}^{128}$
  $z_i \leftarrow \text{Dense}(h_i^{(2)}, 128, \text{ activation=ReLU})$
  $f_i \leftarrow \text{LayerNorm}(z_i)$
end for
return $F = \{f_i\}$

Figure 3. BiLSTM algorithm

Confusion Matrix

To examine the distribution of inaccuracies in the classification process, a quantitative framework of a confusion matrix is used by pairing the actual classes with the predicted classes [26]. The confusion matrix is shown in Table 2 and is represented in a tabular format, where rows indicate the actual classes, while columns indicate the predicted classes [27]. The confusion matrix has four main components consisting of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), where these components play an important role in determining various evaluative metrics such as true positive ratio (TPR), true negative ratio (TNR), false positive ratio, and false negative ratio [28].

Based on Table 2, the model performance metrics can be measured as follows in Equation 4, Equation 5, Equation 6 and Equation 7:

1) Accuracy functions as an indicator of the frequency of the classification model producing correct.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

2) Precision is used to determine how accurately the model predicts the positive class.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

3) Recall, or sensitivity, is used to determine how well the model can detect all positive cases.

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

4) F1-Score is the harmonic average of precision and recall.

$$F1 - Score = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \quad (7)$$

Table 2. Confusion matrix

		Actual Class	
		Positive (P)	Negative (N)
Predicted Class	Positive (P)	True Positive (TP)	False Positive (FP)
	Negative (N)	False Negative (FN)	True Negative (TN)

Statistical Analysis

To measure performance stability, we report metrics on the test data along with bootstrap-based 95% confidence intervals (CIs) [29], [30]. We resampled the test data (x_i, y_i) pairs B times, where B could be 1000. We found Accuracy, Macro-F1, and ROC-AUC for each resample. The lower and upper CIs were derived from the 2.5% and 97.5% percentiles of the bootstrap distribution for each metric, respectively.

RESULT AND DISCUSSION

After the Load and Clean Data stage, the Sentiment140 corpus contains 1,581,466 binary tweets that have been cleaned (lowercase, remove links, mentions/hashtags, punctuation, whitespace normalization) and deduplicated in the text column. The label distribution is very balanced: class 0 (negative) accounts for 790,185 tweets (≈49.97%) and class 1 (positive) accounts for 791,281 tweets (≈50.03%), with a difference of only 1,096 samples (≈0.07 percentage points) as shown in Figure 4. This balance is visible in the Label Distribution graph and benefits the training process by reducing bias towards one of the classes; therefore, aggressive balancing is not necessary (class weight is optional). The subsequent train/val/test data split is performed stratified to maintain the proportion of labels in each split.

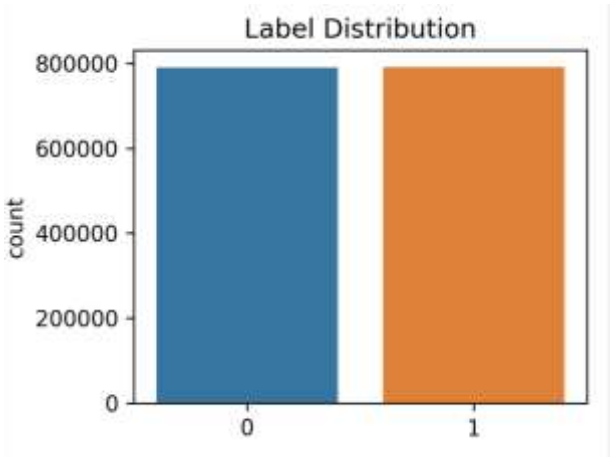

Figure 4. Label distribution

Figure 5 illustrates the confusion matrix for the hybrid model, demonstrating strong performance across the two sentiment classes. The hybrid model, combining locally trained Word2Vec and RoBERTa with a

BiLSTM sequence processor, performed best on Sentiment140. Out of a total of 316,294 tweets, the model successfully identified 136,180 negative tweets and 142,450 positive tweets. There were still errors, with 21,857 negative tweets incorrectly predicted as positive (false positives) and 15,807 positive tweets incorrectly predicted as negative (false negatives).

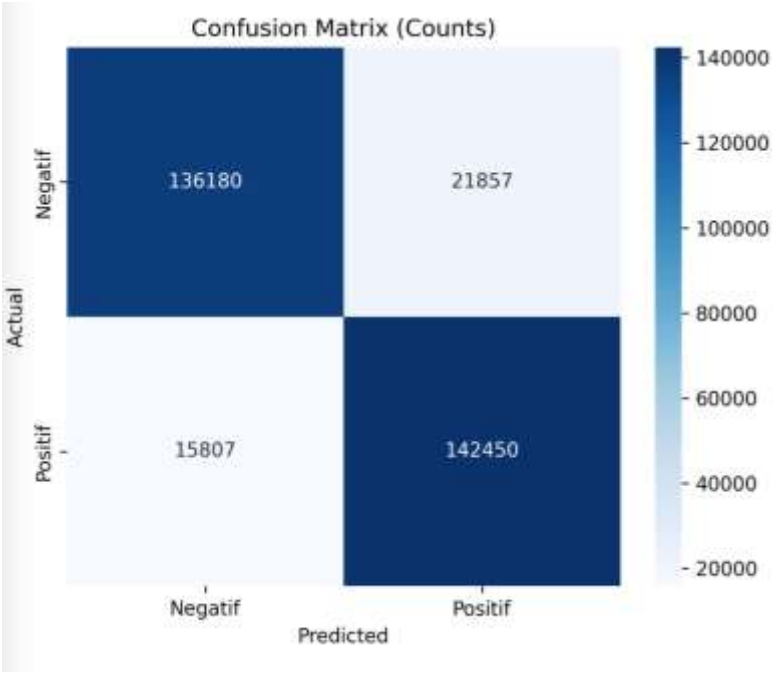

Figure 5. Confusion matrix for the hybrid model

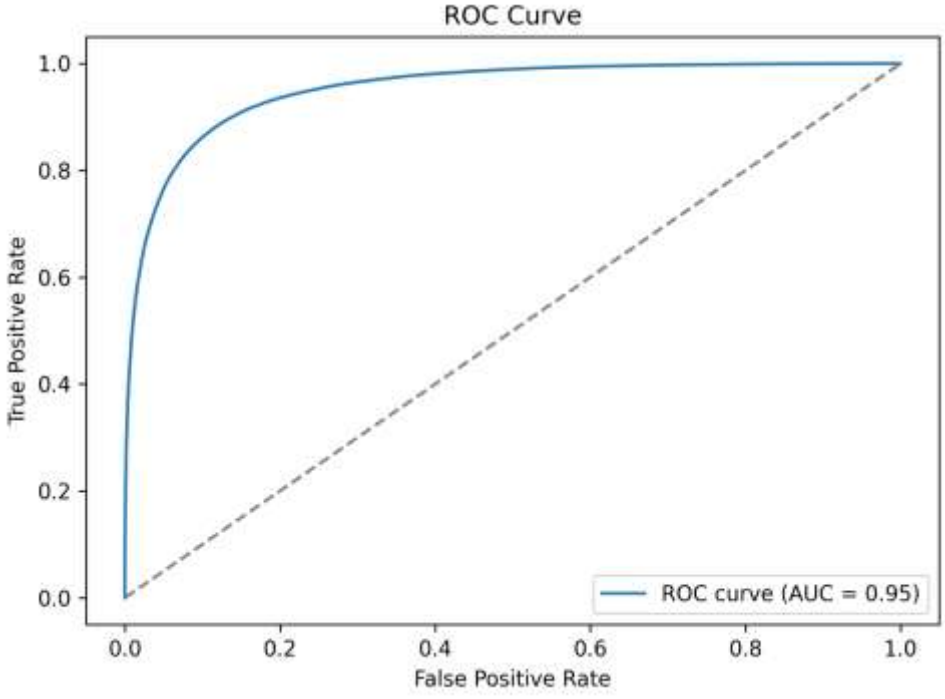

Figure 6. ROC curve for the hybrid model

The dominant diagonal block pattern indicates good class separation and aligns with the test metrics: accuracy of 88.09% and ROC–AUC of 95.19% (Figure 6). Specifically, recall for the positive class is high (90.01%), meaning the model rarely misses positive tweets. In contrast, positive precision (86.70%)

indicates a small portion of negative tweets still "look positive"—typically cases of sarcasm or ambiguous context. If the application is more sensitive to false positives, the decision threshold can be slightly raised from the F1-optimal point to suppress FP with the trade-off of slightly decreased positive recall.

Figure 6 shows the relationship between the True Positive Rate (TPR)/sensitivity and the False Positive Rate (FPR) for various decision threshold values. The diagonal dashed line is a reference for the random model; the further the curve is from this line and closer to the top-left corner, the better its class separation ability. In this graph, the curve curves sharply to the top-left, and the area under the curve (AUC) ≈ 0.95, meaning there is a 95% chance that the model will score a random positive tweet higher than a random negative tweet. In practice, at a low FPR, the model has achieved a high TPR—good for applications that want to suppress false positives without significantly losing positive detections.

Table 3. Ablation study on Sentiment140.

Variant	Accuracy (%)	Macro F1 (%)	ROC-AUC (%)
Word2Vec-only	82.89 [82.75, 83.02]	82.86 [82.73, 82.99]	91.14 [91.03, 91.24]
RoBERTa-only	86.85 [86.73, 86.97]	86.84 [86.72, 86.96]	94.38 [94.29, 94.46]
Hybrid (W2V+RoBERTa)	**88.09 [87.98, 88.21]**	**88.09 [87.97, 88.20]**	**95.19 [95.12, 95.27]**

Values are reported as percentages of the test data along with 95% CIs. CIs for Accuracy and Macro-F1 were calculated using bootstrapping (2.5%--97.5% percentile), while CIs for ROC--AUC were calculated from the standard error of the AUC using the number of positive and negative samples in the test data.

The ablation table in Table 3 clearly shows the contribution of each component. The Word2Vec-only variant was used as the baseline and achieved an accuracy of 82.89% (95% CI: [82.75, 83.02]), a Macro-F1 of 82.86% ([82.73, 82.99]), and an ROC–AUC of 91.14% ([91.03, 91.24]). When switching to RoBERTa-only, all metrics improved consistently—accuracy of 86.85% ([86.73, 86.97]), Macro-F1 of 86.84% ([86.72, 86.96]), and an AUC of 94.38% ([94.29, 94.46])—confirming the power of the transformer's contextual representation for Twitter/X text. The best performance is achieved by Hybrid (Word2Vec+RoBERTa) with an accuracy of 88.09% ([87.98, 88.21]), Macro-F1 of 88.09% ([87.97, 88.20]), and AUC of 95.19% ([95.12, 95.27]). Compared to RoBERTa-only, a 1.24-point increase in accuracy corresponds to a relative error reduction of approximately 9.4% and results in around 3,922 additional correct predictions on ∼approximately 316,000 test tweets. The narrow CIs ranges across all variants indicate that the performance estimates are stable across large test datasets, and the hybrid's superiority remains consistent across both threshold-based (Accuracy/Macro-F1) and threshold-free (ROC–AUC) metrics. Substantively, these findings support the synergy hypothesis: the Word2Vec branch trained on the target corpus injects distributional regularities that are more adaptive to slang/OOV and spelling variations, while RoBERTa maintains an understanding of sentence context; the combination of the two yields more robust classification decisions on noisy Twitter/X text.

Table 4. Accuracy comparison of several methods on the Sentiment140

Method	Accuracy (%)
Naïve Bayes + Logistic Regression + Gradient Boosting Classifier [1]	80.34
LSTM + Stopwords [11]	83.50
RoBERTa+BiLSTM [12]	82.25
Proposed method (W2V+RoBERTa+BiLSTM)	**88.09**

Table 4 compares the accuracy of several research methods on the Sentiment140 dataset. Three baseline approaches—Naïve Bayes + Logistic Regression + Gradient Boosting (80.34%) [1], LSTM with stopword filtering (83.50%) [11], and RoBERTa + BiLSTM (82.25%) [12] —provide an overview of performance when relying solely on a single representation type or a classical pipeline. The proposed method (combined Word2Vec + RoBERTa + BiLSTM) achieved the highest accuracy of 88.09%. This increase equates to an absolute increase of +7.75 points over the classical combination, +4.59 points over LSTM+stopwords, and +5.84 points over RoBERTa+BiLSTM. In terms of error reduction, the proposed method reduces error by approximately 39% compared to the classical baseline, 28% compared to LSTM+stopwords, and 33% compared to RoBERTa+BiLSTM. These results show that combining distributional representations (Word2Vec trained on the target corpus) with contextual representations (RoBERTa) on top of the BiLSTM architecture provides a real synergy: the model is more robust to slang/OOV while still fully understanding the sentence context.

Table 5. Examples of model predictions on the test set.

Type	y	$\hat{y}$	ρ	Error category	Tweet example (truncated)
TP	1	1	1.0000	-	Hi, welcome twitterers! Come often for updates and tweeting fun
TP	1	1	1.0000	-	#followfriday ... Cool tweeps to follow!
TN	0	0	0.0000	-	RIP Farrah Fawcett ... it kindoff hurts. ILY Mommy
TN	0	0	0.0000	-	Farrah Fawcett ... has died from cancer at 62.
FP	0	1	0.9999	noisy / ambiguous context	watching the kids ... Yay for summer finally!
FP	0	1	0.9999	noisy (capitalization/emphasis)	@dominiquedanyel HAPPY BELATED B-DAY
FP	0	1	0.9999	slang / emotional expression	Watching a BlackBuster movie ... WooHooo Harlem Nights
FP	0	1	0.9999	slang / conversational context	@untitleddesign fun times! Great crowd, good music. Lots of drinks
FP	0	1	0.9999	noisy (interjection)	@milagro88 Hi Mila! Yeah, TGIF! ... looking forward to the weekend
FN	1	0	0.0001	noisy (hashtag/technical term)	The Orange TV #iPhone app is really disappointing...
FN	1	0	0.0002	slang/noisy (emoji marker)	has tummy ache ... start contest soon (Y)
FN	1	0	0.0002	noisy (domain jargon)	Internet fail at work continues... It's so sad when IT stop me from working
FN	1	0	0.0002	noisy / implicit context	so bummed I'm going to miss @metricband playing in LA June 8th ...
FN	1	0	0.0003	noisy (expressive marker)	So early to be woken up ... *sigh*

From the Table 5, p denotes the predicted probability of the positive class. The “Error category” column is provided for FP/FN cases to illustrate common error sources in Twitter/X text (slang/OOV, negation, sarcasm, and noise).

To complement the quantitative evaluation in Figures 5–6 and Tables 3–4, we analyzed examples of mispredictions (FP/FN) to identify remaining error patterns in Twitter/X text. Table 5 summarizes examples of correct (TP/TN) and incorrect (FP/FN) predictions, as well as the error source categories for the FP/FN cases. In general, FP were dominated by tweets with strong positive expression markers (e.g., "Yay," capital letters, or interjections), such that the strongly positive lexical signal led the model to predict the positive class even though the label stated negative, which could also be influenced by the noise of Sentiment140's auto-labeling. In contrast, FN were prevalent in tweets with implicit context and incomplete conversational structure, as well as the presence of non-standard markers, such as hashtags, mentions, and domain-specific symbols or jargon, which increased semantic ambiguity. These findings confirm that while the Word2Vec–RoBERTa hybrid architecture improves robustness to informal and OOV vocabulary variations, remaining challenges—especially labeling noise, pragmatic context, and exceptional cases such as sarcasm/negation—still require further addressing in future research (e.g., through data curation, richer labeling schemes, and pragmatic context analysis).

However, there are limitations: the Sentiment140 labels are derived from emoticons and therefore potentially contain noise; the neutral class is not modeled; and the phenomenon of sarcasm or code-mixing remains a source of remaining errors. In the future, domain-adaptive pretraining on a more sophisticated Twitter corpus, emoji/expression normalization, attention pooling in the RoBERTa branch, and multi-seed testing can further reduce errors and strengthen conclusions. Thus, the primary benefit of this study is the quantitative evidence that the Word2Vec–RoBERTa hybridization offers real and measurable advantages for large-scale sentiment analysis in social media.

CONCLUSION

This study proposes a hybrid architecture that combines Word2Vec (for capturing vocabulary and slang/OOV patterns) with RoBERTa (for sentence context understanding) on top of BiLSTM as a sequence processor. Tests on Sentiment140 show that the hybrid approach delivers the best performance. Ablation results confirm that both branches complement each other: Word2Vec enhances robustness to non-standard tokens, while RoBERTa maintains global context understanding. This model is therefore more robust to noisy and variable social media text, and is ready for use in large-scale opinion monitoring or customer service applications.

However, this study still has limitations: the data is in English with emoticon-based auto-labeling, the task is still binary (positive/negative), and it has not specifically explored aspects such as sarcasm or neutrality. Looking ahead, we see several promising development directions, including expansion to multi-class

scenarios (including neutral/sarcasm), probability calibration and cost-based threshold adjustment, augmentation and active learning for new domains, and multilingual adaptation (including Indonesian) with efficient fine-tuning (e.g., LoRA). With these improvements, we anticipate that the system's performance and usability will continue to improve in real-world applications.

As a follow-up, future research will evaluate model stability through multiple random initializations (multi-seed) and expand domain adaptation with more resource-efficient domain-adaptive pretraining or partial fine-tuning strategies. Furthermore, we will add probability calibration evaluations (e.g., Brier score/ECE) to ensure that the output scores are reliably interpretable, and explore active learning to allow the system to adapt more quickly to new domains or topics with minimal annotation costs. In application scenarios, this model has potential uses for industry sentiment monitoring (e.g., customer service and brand reputation), public opinion analysis of policies or events, and real-time social media analytics for trend detection and early warnings, with decision thresholds adjusted according to operational needs.

REFERENCES

[1] H. Pal and B. Bhushan, "Sentiment Analysis on Twitter Dataset using Voting Classifier," in *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/ICEECT61758.2024.10739316.

[2] H. Mehmetcik, E. L. Morgan, M. Kölük, and G. Yüksel, "Socializing IR: Turkish IR Scholars and their Twitter Interactions," *All Azimuth*, vol. 13, no. 1, 2024, doi: 10.20991/allazimuth.1416584.

[3] A. Umair, E. Masciari, and M. H. Ullah, "Vaccine sentiment analysis using BERT + NBSVM and geo-spatial approaches," *Journal of Supercomputing*, vol. 79, no. 15, 2023, doi: 10.1007/s11227-023-05319-8.

[4] A. M. Alsugair and N. S. Alghamdi, "Sentiment Analysis of Arabic Tweets using ARABERT as a fine tuner and feature extractors," in *2024 11th IEEE Swiss Conference on Data Science (SDS)*, IEEE, May 2024, pp. 31–36. doi: 10.1109/SDS60720.2024.00012.

[5] R. Al-Sahar, W. Klumpenhouwer, A. Shalaby, and T. El-Diraby, "Using Twitter to Gauge Customer Satisfaction Response to a Major Transit Service Change in Calgary, Canada," *Transp Res Rec*, vol. 2678, no. 3, 2024, doi: 10.1177/03611981231179167.

[6] A. Jazuli, Widowati, and R. Kusumaningrum, "Auto Labeling to Increase Aspect-Based Sentiment Analysis Using K-Nearest Neighbors Method," in *E3S Web of Conferences*, 2022. doi: 10.1051/e3sconf/202235905001.

[7] X. Yu Li, L. B. Han, and Z. Feng Jiang, "Deep Learning-Based Algorithm for Classification of News Text," *IEEE Access*, vol. 12, pp. 159086–159098, 2024, doi: 10.1109/ACCESS.2024.3487311.

[8] L. Trisnawati, N. A. B. Samsudin, S. K. B. A. Khalid, E. F. B. A. Shaubari, S. -, and Z. Indra, "An Ensemble Semantic Text Representation with Ontology and Query Expansion for Enhanced Indonesian Quranic Information Retrieval," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, 2025, doi: 10.14569/IJACSA.2025.0160148.

[9] K. Karnan and Dr. L. R. A. Babu, "Text Mining and Natural Language Processing Frameworks for Enhanced Fake News Detection, Sentiment Analysis, and Automated Summarization in Social Media," *International Journal of Basic and Applied Sciences*, vol. 14, no. 2, pp. 107–112, Jun. 2025, doi: 10.14419/hgj17c14.

[10] G. Jianan, R. Kehao, and G. Binwei, "Deep learning-based text knowledge classification for whole-process engineering consulting standards," *Journal of Engineering Research*, vol. 12, no. 2, pp. 61–71, Jun. 2024, doi: 10.1016/j.jer.2023.07.011.

[11] P. Rakshit, P. Sarkar, and S. Roy, "Hybrid Deep Learning Approach for Sentiment Analysis on Twitter Data," *Multimed Tools Appl*, vol. 84, no. 15, pp. 15271–15292, Jun. 2024, doi: 10.1007/s11042-024-19555-4.

[12] M. M. Rahman, A. I. Shiplu, Y. Watanobe, and M. A. Alam, "RoBERTa-BiLSTM: A Context-Aware Hybrid Model for Sentiment Analysis," *IEEE Trans Emerg Top Comput Intell*, pp. 1–18, 2025, doi: 10.1109/TETCI.2025.3572150.

[13] Z. Zhang, Y. Meng, and D. Xiao, "Prediction techniques of movie box office using neural networks and emotional mining," *Sci Rep*, vol. 14, no. 1, p. 21209, Sep. 2024, doi: 10.1038/s41598-024-72340-z.
[14] S. Gupta and B. Kishan, "A performance-driven hybrid text-image classification model for multimodal data," *Sci Rep*, vol. 15, no. 1, p. 11598, Apr. 2025, doi: 10.1038/s41598-025-95674-8.
[15] Y. Liu, H. Yu, T. Guan, P. Chen, B. Ren, and Z. Guo, "Intelligent prediction of compressive strength of concrete based on CNN-BiLSTM-MA," *Case Studies in Construction Materials*, vol. 22, p. e04486, Jul. 2025, doi: 10.1016/j.cscm.2025.e04486.
[16] Q. Pu, F. Huang, F. Li, J. Wei, and S. Jiang, "Integrating Emotional Features for Stance Detection Aimed at Social Network Security: A Multi-Task Learning Approach," *Electronics (Basel)*, vol. 14, no. 1, p. 186, Jan. 2025, doi: 10.3390/electronics14010186.
[17] J. Soni and K. Mathur, "Enhancing sentiment analysis via fusion of multiple embeddings using attention encoder with LSTM," *Knowl Inf Syst*, vol. 66, no. 8, pp. 4667–4683, Aug. 2024, doi: 10.1007/s10115-024-02102-w.
[18] B. G. Bokolo and Q. Liu, "Advanced Comparative Analysis of Machine Learning and Transformer Models for Depression and Suicide Detection in Social Media Texts," *Electronics (Basel)*, vol. 13, no. 20, p. 3980, Oct. 2024, doi: 10.3390/electronics13203980.
[19] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *Processing*, vol., 2009.
[20] S. M. Orebi, "Opinion Mining in Text Short by Using Word Embedding and Deep Learning," *Journal of Applied Data Sciences*, vol. 6, no. 1, pp. 526–636, Jan. 2024, doi: 10.47738/jads.v6i1.438.
[21] J. Song, X. Zu, and F. Xie, "A Contrastive Learning Framework for Keyphrase Extraction," *Data Intell*, vol. 6, no. 4, pp. 1032–1056, Dec. 2024, doi: 10.3724/2096-7004.di.2024.0018.
[22] R. Malhotra and J. Patidar, "Enhanced software change proneness prediction in android applications using balanced data techniques and advanced language models," *Discover Computing*, vol. 28, no. 1, p. 51, Apr. 2025, doi: 10.1007/s10791-025-09552-y.
[23] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," Jul. 2019.
[24] R. W. Pratiwi, S. F. Handayani, D. Dairoh, and D. I. Af'idah, "Sentiment analysis on Twitter reviews data using the bidirectional long short-term memory (BiLTM)," 2024, p. 030021. doi: 10.1063/5.0199238.
[25] M. Xia, "A Text Sentiment Analysis Model Based on BiLSTM-Conv1D Deep Neural Network," in *2025 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS)*, IEEE, Feb. 2025, pp. 1–5. doi: 10.1109/ICICACS65178.2025.10968782.
[26] A. Purwinarko, K. Budiman, A. Widiyatmoko, F. A. Sasi, and W. Hardyanto, "Integrating C4.5 and K-Nearest Neighbor Imputation with Relief Feature Selection for Enhancing Breast Cancer Diagnosis," *Scientific Journal of Informatics*, vol. 12, no. 1, pp. 107–118, May 2025, doi: 10.15294/sji.v12i1.21673.
[27] I. Barranco-Chamorro and R. M. Carrillo-García, "Techniques to Deal with Off-Diagonal Elements in Confusion Matrices," *Mathematics*, vol. 9, no. 24, p. 3233, Dec. 2021, doi: 10.3390/math9243233.
[28] C. S. Hong and T. G. Oh, "TPR-TNR plot for confusion matrix," *Commun Stat Appl Methods*, vol. 28, no. 2, pp. 161–169, Mar. 2021, doi: 10.29220/CSAM.2021.28.2.161.
[29] T.-H. Kim *et al.*, "A simulation-based framework for scaling factor modeling and GMM-based zonal stratification in activated decommissioning waste," *Nuclear Engineering and Technology*, vol. 58, no. 2, p. 103981, Feb. 2026, doi: 10.1016/j.net.2025.103981.
[30] F. C. Oettl, A. I. Weinblatt, J. Gutierrez Naranjo, M. Parks, F. Cushner, and A. Gonzalez Della Valle, "Developing and validating machine learning models to predict length of hospitalization before obese patients undergo elective arthroplasty," *J Orthop*, vol. 71, pp. 291–298, Jan. 2026, doi: 10.1016/j.jor.2025.10.008.