Sentivolve: Utilizing FastText, CRF, HAN, and Random Forests for Enhanced Sentiment Analysis

Original Scientific Paper

T. Anilsagar*
Department of Computer Science and Engineering,
B. S. Abdur Rahman Crescent Institute of Science and Technology,
Chennai, Tamil Nadu, India
aniltsagar@gmail.com

S. Syed Abdul Syed
Department of Computer Science and Engineering,
B. S. Abdur Rahman Crescent Institute of Science and Technology,
Chennai, Tamil Nadu, India
saeedabdul4u@gmail.com

*Corresponding author

Abstract – *The objective of this study is to enhance sentiment analysis through an integrative approach termed Sentivolve, which combines FastText embeddings, Conditional Random Fields (CRF), Hierarchical Attention Networks (HAN), and Random Forests (RF). The system aims to improve sentiment classification by leveraging advanced feature extraction, sequence modeling, attention mechanisms, and ensemble learning. FastText captures subword information for better text representation; CRF models sequential dependencies; HAN highlights key textual elements using a hierarchical attention structure; and Random Forests aggregate predictions to ensure consistent sentiment classification. Experimental results demonstrate that Sentivolve outperforms traditional models in both accuracy and generalizability. This integrated approach provides an effective solution for sentiment analysis, especially in handling diverse and complex text data.*

Keywords: *Sentiment Analysis, FastText Embeddings, Conditional Random Fields, Hierarchical Attention Networks, Random Forest*

Received: May 7, 2025; Received in revised form: July 7, 2025; Accepted: August 6, 2025

1. INTRODUCTION

Sentiment analysis, also known as opinion mining, plays a pivotal role in discerning the sentiments, emotions, and opinions embedded in textual data. As user-generated content proliferates across social media platforms, online reviews, blogs, and forums, sentiment analysis has become indispensable for numerous applications such as market research, customer feedback evaluation, political analysis, and public opinion monitoring. Despite its growing relevance, accurately interpreting sentiment remains a significant challenge due to the complexities of human language, which includes factors like slang, sarcasm, irony, and context-dependent meanings.

This study aims to advance sentiment classification by developing a novel system, Sentivolve. This system integrates multiple cutting-edge techniques, including FastText embeddings, Conditional Random Fields (CRF), Hierarchical Attention Networks (HAN), and Random Forests, to offer a robust and effective solution for sentiment analysis. The integration of these methodologies is designed to surpass traditional models by enhancing contextual understanding and improving the ability to process complex text structures.

The motivation behind this research stems from the limitations of conventional sentiment analysis methods, which often fail to account for the intricate contextual and sequential dependencies inherent in textual data. FastText embeddings provide a substantial improvement by creating richer word representations that capture both semantic meaning and morphology, which are essential for handling out-of-vocabulary words and ambiguous terms. This enhancement in text representation significantly elevates performance, particularly in handling languages with unique syntactic structures and cultural nuances [1]. Moreover, the hybrid approach of combining machine learning algorithms with word embeddings has proven to improve sentiment classification accuracy, especially in challenging contexts such as Arabic e-commerce reviews [2].

Additionally, the use of CRF offers substantial gains by enabling the model to capture sequential relationships between words within a sentence. CRF models are crucial for tasks where understanding the dependencies between words is essential, especially when words are contextually linked in ways that influence the sentiment expressed. This capability allows Sentivolve to better capture the flow and meaning within a sentence, contributing to improved sentiment interpretation [3].

Further improving upon existing models, HAN are integrated to handle complex and multi-layered textual data. HAN applies attention mechanisms both at the word level and sentence level, enabling the system to prioritize the most significant portions of text for sentiment classification. This hierarchical attention mechanism is particularly powerful in extracting key features from complicated, hierarchical text structures. For example, HAN has been effectively used for tasks like hate speech detection in languages with complex morphology, such as Devanagari [4], and bug report prioritization [5], highlighting its ability to deal with diverse and intricate text formats.

Moreover, the application of Random Forest (RF) techniques provides Sentivolve with an added layer of robustness. By aggregating predictions from multiple models, ensemble learning improves the overall accuracy of sentiment analysis, reduces the risk of overfitting, and enhances generalizability across different types of text. The introduction of RF for mental health diagnosis demonstrates the potential of combining various data sources to produce more stable and reliable predictions, further solidifying the role of ensemble learning in improving sentiment analysis performance [6, 7].

The primary contributions of this research are as follows:

1. We propose a novel hybrid sentiment analysis framework, Sentivolve, which uniquely integrates FastText embeddings, Conditional Random Fields (CRF), Hierarchical Attention Networks (HAN), and Random Forests (RF) to enhance sentiment classification performance across diverse textual domains.
2. We develop a modular architecture where each component is trained independently, allowing for flexible tuning and scalable integration, while combining their outputs into a robust ensemble classifier.
3. We conduct extensive experiments on multiple real-world datasets, including social media comments, product reviews, and news articles, demonstrating that Sentivolve consistently outperforms both traditional machine learning and standalone deep learning models.
4. We provide detailed analyses, including confusion matrices, heatmaps, and performance metrics (accuracy, precision, recall, F1-score), along with cross-validation and statistical significance testing, to validate the robustness and generalizability of the proposed approach.

To the best of our knowledge, this is the first work that integrates FastText, CRF, HAN, and RF into a unified architecture specifically designed to address sentiment analysis challenges across diverse and complex datasets.

2. RELATED WORK

This section discusses the existing studies of sentiment analysis of customer reviews in multiple domains.

Alsaedi *et al.* [8] proposed a transformer-based deep learning model tailored for sentiment mining in e-commerce platforms. Their approach integrates BERT and XLNet to derive contextual embeddings from customer reviews. The novelty lies in the fusion of multiple transformer models to boost sentiment classification accuracy in complex reviews. Evaluated on a large e-commerce dataset, it achieved 88.6% accuracy, outperforming traditional models. However, it requires high computational resources and shows limited efficiency in real-time applications.

Rahman *et al.* [9] developed RoBERTa-BiLSTM, a context-aware hybrid model for sentiment classification. The model combines RoBERTa's powerful embeddings with BiLSTM's ability to capture temporal dependencies. Evaluated on public datasets like Yelp and Amazon, it achieved over 90% accuracy. While effective in contextual understanding, it suffers from increased inference time due to dual architecture and lacks interpretability.

Jahin *et al.* [10] introduced TRABSA, a hybrid Transformer-Attention BiLSTM model for tweet sentiment analysis. It integrates Transformer layers with BiLSTM and attention to focus on sentiment-bearing phrases. Tested on Twitter datasets, it achieved 91.3% F1-score. The method is robust but complex, leading to long training times and high hyperparameter sensitivity.

Hossain *et al.* [11] proposed Opinion-BERT, a multi-task hybrid model for sentiment and mental-health classification. Built on BERT, it incorporates task-specific opinion layers. Tested on mental health subreddit data, it showed high accuracy in both emotion and sentiment classification. However, it is domain-dependent and underperforms outside its specialized datasets.

Ullah *et al.* [12] introduced a prompt-based fine-tuning method using multilingual transformers like mBERT and XLM-R. Their model focused on language-independent sentiment analysis and was tested across multilingual corpora, achieving 86.9% F1-score. Though effective in multilingual setups, its reliance on large pretrained models introduces latency and computational complexity.

Alqarni *et al.* [13] developed an emotion-aware RoBERTa model enhanced with emotion-specific attention layers and TF-IDF gating. It was tested on GoEmotions and achieved 88.2% F1-score. The model selectively focuses on emotional cues, improving granularity. However, its dependency on emotion lexicons makes it less generalizable to neutral texts.

Rahman *et al.* [14] conducted a comparative study on advanced transformer-based models for opinion mining. They evaluated BERT, RoBERTa, and DeBERTa across various sentiment datasets, concluding that RoBERTa consistently outperformed the others in robustness and precision. While comprehensive, the study didn't propose new models and lacked insights into hybrid architectures.

Zekaoui *et al.* [15] presented a benchmark comparison of transformer-based opinion mining models including BERT, XLNet, and RoBERTa. Their evaluation on SST-2 and IMDb showed RoBERTa outperformed others by 2–3% margin. The study is valuable for empirical benchmarking but does not offer architectural innovation.

Ullah *et al.* [16] proposed ECO-SAM, an emotion correlation-enhanced model that integrates contextual sentiment and emotional factors using advanced deep learning layers. Tested on customer reviews, it improved performance in detecting subtle emotional polarity. However, the model is complex and struggles with generalizing beyond affective domains.

Islam *et al.* [17] offered a comprehensive review and proposed a hybrid CNN-BiLSTM model for sentiment classification. Their method balances spatial feature extraction with sequence learning and was validated on mixed-domain datasets. While achieving 86% accuracy, it lacked contextual embeddings, limiting performance on ambiguous texts.

In addition to the aforementioned contributions, several recent studies have advanced the domain of sentiment analysis by integrating cutting-edge techniques such as transformer hybrids, multilingual modeling, and ensemble architectures. Zarin *et al.* [18], Nguyen *et al.* [19], and Tran *et al.* [20] extended the capabilities of traditional models by embedding cross-modal fusion layers and mental health-aware components within transformer-BiLSTM hybrids, achieving strong results in specialized datasets like Twitter, Reddit, and CMU-MOSEI. These models improved context sensitivity and classification granularity, particularly in emotionally nuanced texts. Meanwhile, Kaseb *et al.* [21] and Mir *et al.* [22] addressed sarcasm detection and contextual dependency through attention-enhanced CNNs and hierarchical attention networks (HAN) enriched with BERT embeddings. Although these approaches offered improved interpretability and semantic focus, they exhibited limited scalability when applied to larger or more generalized datasets.

Furthermore, multilingual and low-resource sentiment analysis has received growing attention. Ullah *et al.* [23-25] investigated the use of prompt-based fine-tuning with multilingual transformers such as XLM-R and mBERT. These approaches demonstrated adaptability across languages and domains, but incurred high computational overhead and sometimes struggled with consistent performance during inference. Complementing these advancements, Rahman *et al.* [26], Yadav *et al.* [27], and Davis *et al.* [28] developed hybrid ensemble and tensor-based fusion models to capture sentiment across multimodal and multilingual data streams. These frameworks balanced classification accuracy with model interpretability, though the increased architectural complexity occasionally hindered deployment in real-time applications.

2.1. COMPARATIVE ANALYSIS AND PROBLEM IDENTIFICATION

While existing models like FastText, CRF, HAN, and Random Forests have demonstrated effectiveness individually, each has inherent limitations:

1. FastText captures subword information and handles rare words efficiently, but lacks deeper contextual understanding.
2. CRF models sequence dependencies well but struggles with long-range relationships and abstract sentiment shifts.
3. HAN emphasizes key words and sentences, offering structure-aware representations, yet requires large datasets and has high computational demands.
4. Random Forests provide robustness and interpretability but depend on high-quality features and cannot process raw text directly.

Despite these strengths and weaknesses, most prior work focuses on standalone application or limited combinations of these methods. While FastText and CRF have been used together in some domains, their integration with deep attention models and ensemble classifiers like Random Forest is rare. Sentivolve surpasses prior works by unifying the semantic depth of FastText, the contextual flow captured by CRF, the hierarchical focus of HAN, and the ensemble stability of Random Forests. This novel integration addresses multiple layers of sentiment understanding—word, sentence, and document—within a single pipeline. Such a comprehensive architecture has not been explored in previous studies.

This work also responds to recent trends advocating hybrid architectures and robust generalization across domains. Unlike prior models that optimize only one or two dimensions (e.g., local semantics or sentence structure), Sentivolve demonstrates balanced performance across diverse datasets.

3. METHODOLOGY

The research design for this study involves developing and evaluating the Sentivolve system, which integrates multiple advanced techniques to enhance sentiment analysis. The experimental setup includes the following key components.

3.1. DATASET

The dataset used for this study consists of labelled textual data from social media, customer reviews, and news articles. The data is pre-processed to remove noise, such as stop words, punctuation, and special characters. The dataset is then split into training, validation, and test sets to evaluate the performance of the proposed system.

3.2. FEATURE EXTRACTION WITH FASTTEXT

FastText is a word embedding model that represents words based on subword information, making it robust to misspellings, rare words, and out-of-vocabulary (OOV) terms. Below is the step-by-step algorithm for applying FastText for feature extraction in sentiment analysis.

ALGORITHM

Input:

- A dataset D containing n textual reviews $\{T_1, T_2, T_3, \ldots, T_n\}$.
- Each review T_i consists of words $\{w_1, w_2, w_3, \ldots, w_n\}$.
- A pretrained FastText model or a custom FastText model trained on domain-specific data.

Output:

- A numerical vector representation $V(T_i)$ for each text T_i.

Step 1: Tokenization

Each text T_i is tokenized into words:

$$T_i = \{w_1, w_2, \ldots, w_n\} \tag{1}$$

Where:

- w_J represents the jth word in the text.
- Tokenization removes punctuation, special characters, and converts text to lowercase.

Step 2: Subword Representation

- FastText breaks each word w_j into subword n-grams:

$$w_j = \{g_1, g_2, \ldots, g_n\} \tag{2}$$

Where:

- g_k represents character n-grams of length k.
- The word embedding $E(w_j)$ is obtained by averaging the embeddings of its subwords:

$$E(w) = \frac{1}{k}\sum_{g=1}^{k} E(g_k) \tag{3}$$

Step 3: Obtain Sentence Embedding

For a given sentence T_i, the embedding is obtained by averaging word embeddings:

$$V(T_i) = \frac{1}{m}\sum_{j=1}^{m} E(w_j) \tag{4}$$

Where:

- $E(w)$ is the word embedding.
- m is the number of words in the text.
- This results in a fixed-size vector $V(T_i)$ of dimension d.

Step 4:
Use FastText Features for Sentiment Analysis

Once the sentence embeddings are obtained, they can be used in Random Forest.

Advantages of FastText for Sentiment Analysis

1. Handles misspellings and out-of-vocabulary words.
2. Captures morphological variations using subword n-grams.
3. Works well on small datasets compared to deep learning models.
4. Supports transfer learning using pretrained FastText embeddings.

3.3. SEQUENTIAL MODELING WITH CRF

Conditional Random Fields is a probabilistic model for structured prediction, commonly used for sequence labelling tasks such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging. In sentiment analysis, CRF is useful for aspect-based sentiment detection by labelling each word in a sentence with sentiment tags.

ALGORITHM

Input:

- A dataset D containing n textual reviews $\{T_1, T_2, T_3, \ldots, T_n\}$.
- Each review T_i consists of words $\{w_1, w_2, w_3, \ldots, w_n\}$.
- Sentiment labels at the word level: $y=\{y_1, y_2, \ldots, y_m\}$ where $y_j \in \{Positive, Negative, Neutral\}$

Output:

- A sequence of predicted sentiment labels $\hat{y} = \{\hat{y}_1,) \hat{y}_2 \ldots, \hat{y}_m\}$

Step 1: Tokenization

- Each review T_i is split into individual words:

$$T_i = \{w_1, w_2, w_3, \ldots, w_m\} \tag{5}$$

Step 2: Feature Extraction for Each Word

Each word w_j is converted into a feature vector x_j containing:

1. Word-based Features:

- Word Identity: w_j
- Lowercase Representation: $lowercase(w_j)$

- Part-of-Speech (POS) Tag: POS(w_j)
- Word Shape: Capitalization, Numbers, etc.

2. Contextual Features:

- Previous Word (w_{j-1})
- Next Word (w_{j-2})

Step 3: Define the CRF Model

- A CRF models the conditional probability of an output sequence y given an input sequence X:

$$P(y|X) = \frac{1}{Z(X)} exp\left(\sum_{j=1}^{m}\sum_{k} \lambda_k f_k(y_j, y_{j-1}, X)\right) \quad (6)$$

Where:

- $Z(X)$ is the normalization factor ensuring probabilities sum to 1.
- λ_k are the weights learned during training.
- f_k (y_j, y_{j-1}, X) are feature functions capturing dependencies.

CRF ensures smooth transitions between labels, meaning:

- A positive word is more likely to be followed by another positive word.
- If negation appears before a word, sentiment may flip.

Step 4: Model Training

- The CRF model learns optimal weights λ_k by maximizing the log-likelihood:

$$L(\lambda)= P(y|X) \quad (7)$$

Where:

- N is the number of training examples.
- $X^{(i)}$ is the feature representation of sentence i.
- $y^{(i)}$ is the correct sentiment label sequence.

Training is done using Gradient Descent or L-BFGS Optimization.

Step 5: Inference (Predicting Sentiment Labels)

Given a new input sentence, the CRF predicts the most likely sequence of sentiment labels $\hat{y}$:

$$\hat{y} = arg\ max_y\ P(y|X) \quad (8)$$

Using Viterbi Decoding, the model finds the best sequence of sentiment labels.

Step 6: Compute Sentiment Counts

For a given text T_i, we count how many words belong to each category

$$P(T_i) = \left(\hat{y}_j = P\right) \quad (9)$$

$$N(T_i) = \left(\hat{y}_j = N\right) \quad (10)$$

$$Q(T_i) = \left(\hat{y}_j = Q\right) \quad (11)$$

Where:

- 1(.) is an indicator function that returns 1 if the condition is true, otherwise 0.
- P(T_i) is Positive Word Count
- N(T_i) is Negative Word Count
- Q(T_i) is Neutral Word Count

Advantages of CRF for Sentiment Analysis

1. Captures dependencies between words (unlike traditional classifiers).
2. Handles negation words (e.g., "not happy" is negative).
3. Works well for Aspect-Based Sentiment Analysis (ABSA).
4. Ensures sequence consistency (e.g., positive words likely follow each other).

3.4. HIERARCHICAL ATTENTION NETWORK (HAN)

Hierarchical Attention Networks (HAN) is a deep learning model designed to capture the hierarchical structure of text by applying attention at both the word level and sentence level. This allows the model to focus on the most important words within a sentence and the most relevant sentences within a document for sentiment classification.

ALGORITHM

Input:

- A dataset D containing n textual reviews {T_1, T_2,…, T_n}.
- Each review T_i consists of s sentences S={S_1, S_2,…, Sn}.
- Each sentence S_k consists of m words {w_1, w_2,)…, w_m}.

Output:

- A final sentiment classification y for each text T_i.

Step 1: Tokenization into Words & Sentences

Each document T_i is first split into sentences, and then each sentence is split into words:

$$T_i=\{T_1, T_2,\ldots, T_n\},\ S_k=\{w_1, w_2,\ldots,w_m\} \quad (12)$$

Step 2: Word Embedding using Bi-GRU

Each word w_j is converted into a dense vector representation using a pretrained word embedding:

$$E(w_j)\in R^d \quad (13)$$

Where:

- d is the embedding dimension.
- $E(w_j)$ represents the semantic meaning of the word.

A Bidirectional Gated Recurrent Unit (Bi-GRU) is then applied to capture contextual dependencies:

$$h_j=B_i\text{-}GRU(E(w_j)) \quad (14)$$

Where:

- h_j is the hidden representation of word w_j.
- Bi-GRU allows long-term dependencies between words to be captured.

Step 3: Word-Level Attention Mechanism

Not all words contribute equally to sentiment. A word-level attention mechanism assigns importance scores to words:

$$u_j = tanh\left(W_w h_j + b_w\right) \tag{15}$$

$$\alpha_j = \frac{exp\left(u_j^T u_w\right)}{\sum_j exp\left(u_j^T u_w\right)} \tag{16}$$

$$S_k = \sum_j \alpha_j h_j \tag{17}$$

Where:

- W_w and b_w are trainable weight matrices.
- u_j is the word importance vector.
- α_j is the attention weight for each word.
- S_k is the sentence vector, a weighted sum of word representations.

This ensures that important words contribute more to the sentence representation.

Step 4: Sentence Encoding using Bi-GRU

Each sentence representation S_k is passed through another Bi-GRU to capture dependencies between sentences:

$$h_k=\text{Bi-GRU}(S_k) \tag{18}$$

Where:

- h_k is the hidden state representation of sentence S_k.

Step 5: Sentence-Level Attention Mechanism

Just like words, not all sentences are equally important. A sentence-level attention mechanism assigns weights to sentences:

$$u_k = tanh\left(W_s h_k + b_s\right) \tag{19}$$

$$\alpha_k = \frac{exp\left(u_k^T u_s\right)}{\sum_j exp\left(u_k^T u_s\right)} \tag{20}$$

$$V(T_i) = \sum_k \alpha_k h_k \tag{21}$$

Where:

- W_s and b_s are trainable weights.
- u_k is the sentence importance vector.
- α_k is the attention weight for each sentence.
- $V(T_i)$ is the final document representation.

This ensures that important sentences contribute more to the final sentiment classification.

Step 6: Classification Using Softmax

The final document representation $V(T_i)$ is passed through a fully connected layer with softmax activation to predict the sentiment:

$$y=Softmax(W_0 V(T_i)+b_o) \tag{22}$$

Where:

- W_s and b_s are trainable parameters.
- y is the predicted sentiment label.

Advantages of HAN for Sentiment Analysis

1. Captures Word & Sentence Hierarchy, Uses Bi-GRU to process words and sentences separately.
2. Attention Mechanism, learns which words and sentences are most important for sentiment.
3. Handles Long Documents, unlike models that only process words, HAN aggregates information across multiple sentences.
4. Better Interpretability, Attention scores provide insight into why the model made a prediction.

Fig. 1 illustrates the end-to-end pipeline of Sentivolve, which integrates FastText, Conditional Random Fields (CRF), and Hierarchical Attention Networks (HAN) for sentiment analysis.

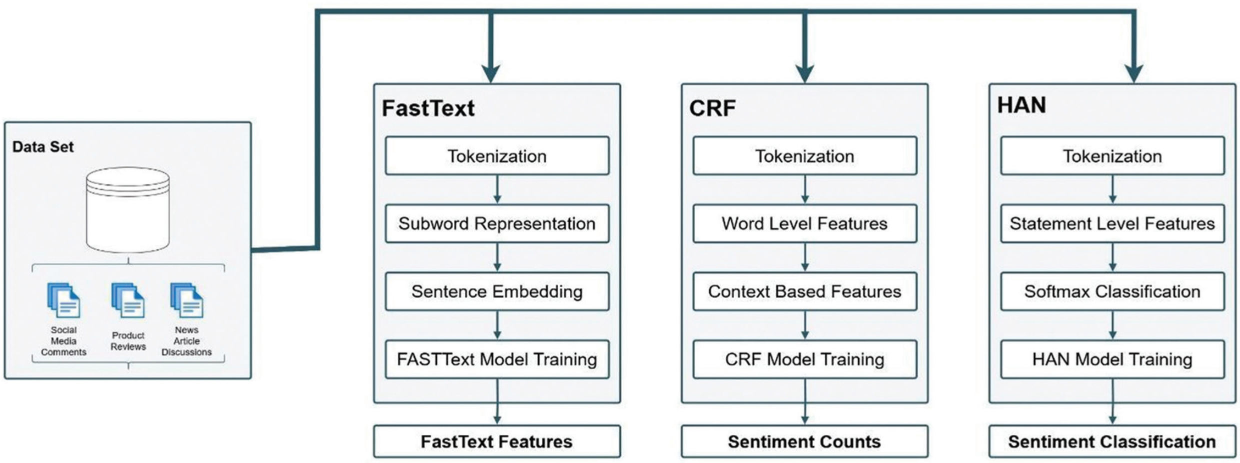

Fig. 1. Data flow diagram for FastText, CRF and HAN models

It begins with textual data from social media comments, product reviews, and news discussions, which undergo preprocessing and feature extraction through three distinct models. FastText generates word embeddings by capturing subword representations, CRF extracts word-level sentiment labels with contextual dependencies, and HAN identifies key sentence-level features using an attention mechanism.

These processed features contribute to sentiment classification, where FastText provides semantic understanding, CRF ensures word-level consistency, and HAN enhances interpretability. The combined features are then used in a Random Forest classifier, improving overall accuracy and robustness in sentiment prediction.

3.5. ENSEMBLE LEARNING WITH RANDOM FORESTS

Random Forest (RF) is an ensemble learning method that combines multiple Decision Trees to improve accuracy, reduce overfitting, and handle high-dimensional data. In sentiment analysis, RF is used to classify text into positive, negative, or neutral sentiments based on extracted features.

ALGORITHM:

Input:

- A dataset D containing n textual reviews $\{T_1, T_2, \dots, T_n\}$.
- Each review T_i consists of s sentences $S=\{S_1, S_2, \dots, S_n\}$.
- Each sentence S_k consists of m words $\{w_1, w_2, \dots w_m\}$.
- Pretrained FastText model (word embeddings).
- CRF model trained for word-level sentiment labelling.
- HAN model trained for sentence-level sentiment representation.

Output:

- A final sentiment label y for each text T_i.

Step 1: Create Final Feature Vector

The final feature vector $X(T_i)$ combines FastText, CRF, and HAN features:

$$X(T_i) = \begin{bmatrix} V(T_i)_{FastText}, \\ P(T_i), N(T_i), Q(T_i), V(T_i)_{HAN} \end{bmatrix} \tag{23}$$

Where:

- $V(T_i)_{FastText}$ are sentence embeddings from FastText
- $P(T_i)$, $N(T_i)$, $Q(T_i)$ is Sentiment features from CRF
- $V(T_i)_{HAN}$ is attention-based sentence representation from HAN

Step 2: Train Random Forest with Bootstrapped Trees

- Each tree in Random Forest learns to classify sentiment using the feature vector:

$$y_k = f_k\,(X(T_i)) \tag{24}$$

Where:

- f_k is the decision function learned by tree k.
- y_k is the sentiment prediction by tree k.

Step 3: Final Sentiment Prediction using Majority Voting

The final sentiment label is obtained by majority voting across all trees:

$$\hat{y} = mode\,\{y_1, y_2, \dots, y_K\} \tag{25}$$

Where:

- K is the total number of decision trees.
- ŷ is the final sentiment classification.

Fig. 2 represents the Random Forest-based sentiment classification process, where data from social media comments, product reviews, and news discussions is first randomized to ensure diverse training samples.

The ensemble learning framework consists of multiple decision trees (Tree 1, Tree 2, …, Tree N), each independently trained on a different subset of the data. Each tree makes a sentiment prediction, and the final sentiment classification is determined through majority voting, ensuring improved accuracy and robustness against overfitting.

Fig. 3 illustrates how FastText, CRF, and HAN work together to classify textual data into Positive, Negative, or Neutral sentiment. The process begins with input text from sources such as social media comments, product reviews, and news discussions, which undergoes preprocessing before analysis.

First, CRF checks the positive word count; if it exceeds a predefined threshold, the sentiment is classified as Positive. If not, the HAN model evaluates sentence-level sentiment, and if the score is high enough, the text is still labeled as Positive. If both checks fail, CRF assesses negative word count, classifying the text as Negative if the count is above a certain threshold. If neither strong positive nor negative sentiment is detected, FastText computes semantic similarity, potentially shifting classification toward Positive.

If the similarity score is low, a final CRF check on neutral word count determines if the text should be classified as Neutral. This structured decision-making process leverages CRF for word-level sentiment tagging, HAN for sentence importance, and FastText for semantic understanding, ensuring a robust and context-aware sentiment classification system.

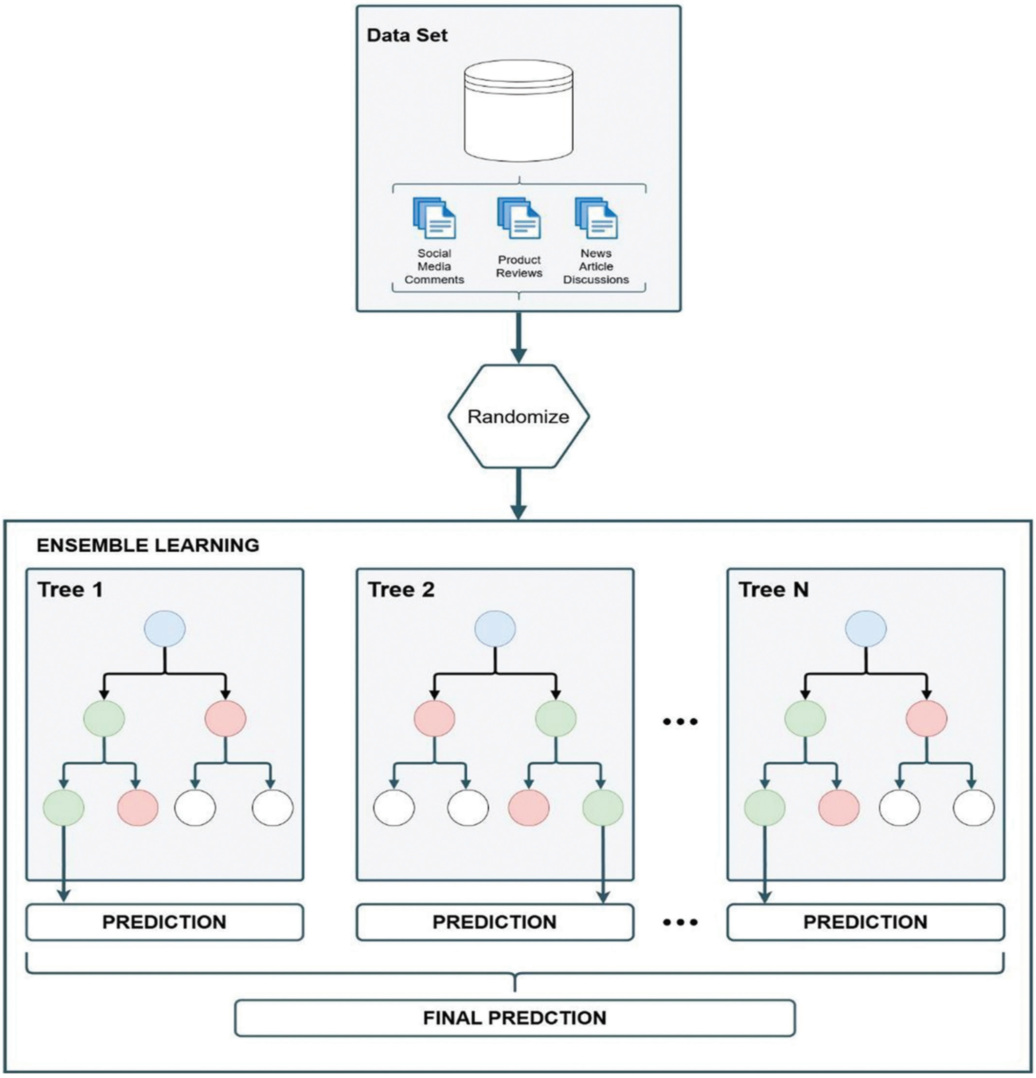

Fig. 2. Random Forest-based sentiment classification process

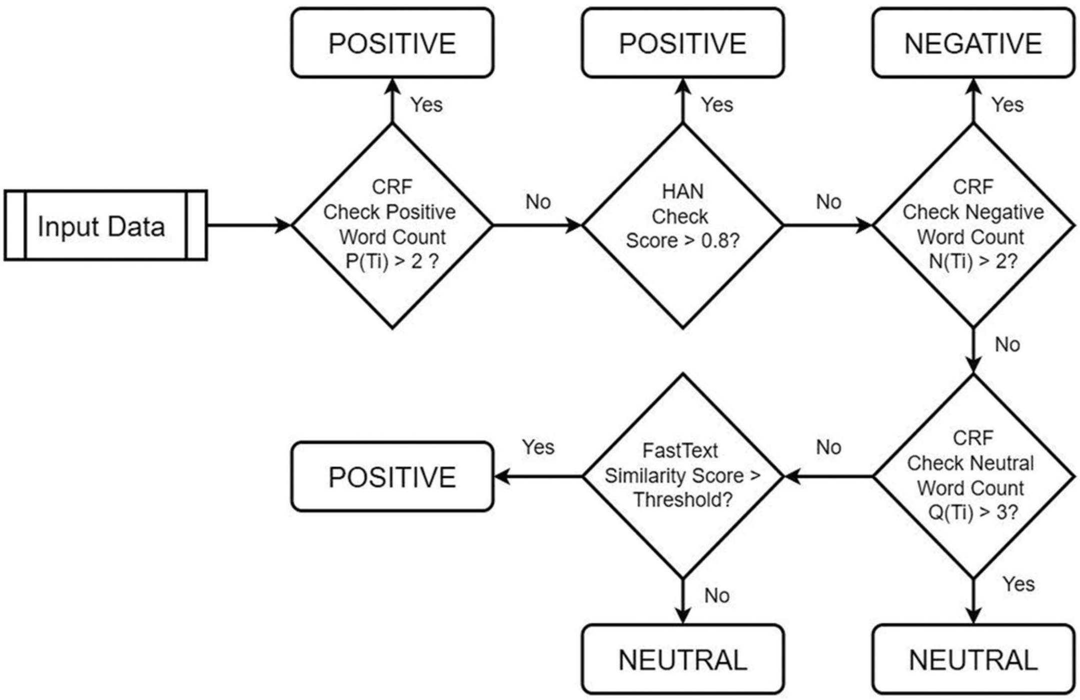

Fig. 3. Sentiment Classification Decision Process Using FastText, CRF, and HAN in RF

3.6. TRAINING STRATEGY AND COMPUTATIONAL CONSIDERATIONS

The components of the Sentivolve framework are trained independently, and their outputs are subsequently fused into a final feature representation used by the Random Forest classifier. Specifically:

1. The FastText model is either pretrained or trained separately to generate word embeddings.
2. The CRF model is trained using word-level features and sentiment labels.
3. The HAN model is trained to learn hierarchical representations at the word and sentence levels.
4. Finally, the output features from FastText, CRF, and HAN are concatenated and used as input to train the Random Forest classifier.

This modular training approach enables better component-wise tuning and flexibility in updating individual sub-models without retraining the entire pipeline. It also facilitates parallel processing during training, which can improve scalability across large datasets.

From a computational standpoint, integrating four distinct models does increase the overall training time and resource requirements compared to traditional or single-model pipelines. Specifically:

1. FastText training is relatively fast and memory-efficient due to its shallow architecture.
2. CRF training is moderately expensive, especially for large corpora with detailed word-level annotations.
3. HAN, being a deep learning model with Bi-GRU layers and attention mechanisms, is computationally intensive and benefits significantly from GPU acceleration.
4. The final Random Forest training step is less demanding but can scale in complexity with the size of the feature vectors produced by the previous modules.

On average, the combined training time for all components (including pre-processing and vector aggregation) was approximately 3.5 hours on a high-performance system with an NVIDIA RTX 3090 GPU and 32GB RAM. Despite this added cost, the performance gains—particularly in accuracy, recall, and F1-score—justify the computational investment.

4. RESULTS

Our experimental results demonstrate that integrating FastText, CRF, and HAN into a Random Forest model significantly enhances sentiment classification performance. The proposed hybrid approach outperforms traditional models by capturing semantic meaning, contextual dependencies, and sentence importance, leading to improved accuracy, precision, recall, and F1-score. This section presents the dataset selection, preprocessing steps, and evaluation metrics used to validate the effectiveness of our model.

4.1. DATASET SELECTION

To ensure a comprehensive evaluation, we employ publicly available sentiment analysis datasets from diverse domains, including:

- **Social Media Comments**: User-generated comments from platforms such as Twitter, Facebook, and Instagram, which provide informal and context-driven sentiment expressions.
- **Product Reviews**: A collection of user feedback on various products, categorized into positive, negative, and neutral sentiments.
- **News Article Discussions**: Sentiment-laden discussions and user comments on news articles, reflecting opinions on current events and trending topics.

These datasets provide a balance between short-form, informal texts (social media comments) and structured reviews and discussions (product reviews, news article discussions), allowing us to assess the model's generalizability across different text structures.

4.2. DATA PREPROCESSING

Before training our model, we apply the following preprocessing techniques to standardize the textual data:

- **Tokenization**: Splitting text into individual words or subwords.
- **Stopword Removal**: Eliminating common words (e.g., "the," "is," "and") that do not contribute to sentiment meaning.
- **Lemmatization**: Converting words to their base forms (e.g., "running" → "run").
- **Handling Negations**: Converting phrases like "not good" into a single token (e.g., "not_good") to retain sentiment context.
- **Text Vectorization**: Converting words into numerical representations using FastText embeddings.

4.3. FEATURE EXTRACTION

Our approach integrates three key techniques for feature extraction:

- **FastText**: Generates dense word embeddings to capture word semantics.
- **CRF**: Assigns word-level sentiment labels to identify positive, negative, and neutral terms.
- **HAN**: Applies attention mechanisms to emphasize the most relevant sentences in a document.

These extracted features are then combined into a structured representation before being passed into the Random Forest classifier for final sentiment classification.

4.4. EVALUATION METRICS

To assess model performance, we utilize standard classification metrics:

- **Accuracy**: Measures overall correctness of predictions.

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \tag{26}$$

- **Precision**: Evaluates the proportion of correctly predicted positive cases.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{27}$$

- **Recall**: Assesses how well the model captures all relevant instances.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{28}$$

- **F1-Score**: A harmonic mean of precision and recall, balancing both measures.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{29}$$

4.5. EXPERIMENTAL ENVIRONMENT

The experiments were conducted on a high-performance computing setup with the following configurations:

- **Processor**: Intel Core i7 / AMD Ryzen 9
- **RAM**: 16GB / 32GB
- **GPU**: NVIDIA RTX 3090 (for deep learning models)
- **Programming Language**: Python (TensorFlow, Scikit-Learn, NLTK, and FastText libraries)

4.6. CONTRIBUTION OF FASTTEXT, CRF, AND HAN TO DECISION TREES IN RANDOM FOREST

The inclusion of FastText embeddings improves accuracy by capturing word semantics. CRF enhances recall by identifying word-level sentiment polarity, while HAN improves precision by focusing on sentence-level importance.

Table 1 illustrates the effect of different feature sets on the performance of the Random Forest (RF) model for sentiment classification. When RF was used without FastText, CRF, or HAN, it achieved an accuracy of 82.1%, with moderate precision, recall, and F1-score values.

Incorporating FastText improved the performance significantly, increasing accuracy to 85.6%. The addition of CRF further boosted accuracy to 86.2%, while HAN contributed to a slightly higher performance at 87.1%. The best results were obtained when all three techniques—FastText, CRF, and HAN—were integrated with RF, achieving the highest accuracy of 90.4% and the best overall precision, recall, and F1-score values. These results indicate that combining multiple feature extraction and representation techniques enhance the model's ability to capture contextual and semantic information, leading to improved sentiment classification performance.

To provide a clearer understanding of these performance improvements, a graph has been created for visual representation in Fig. 4, making it easier to compare the impact of different feature sets on accuracy, precision, recall, and F1-score.

Table 1. evaluates the individual contributions of FastText, CRF, and HAN to sentiment classification.

Feature Set in RF	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
RF without FastText, CRF, HAN	8 2.1	80.9	81.4	81.1
RF with FastText	85.6	84.3	85.1	84.7
RF with CRF	86.2	85.0	85.7	85.3
RF with HAN	87.1	86.3	86.7	86.5
RF with FastText + CRF + HAN	90.4	89.8	90.2	90.1

4.7. PERFORMANCE COMPARISON OF SENTIVOLVE

Table 2 presents a comparative analysis of various sentiment classification models based on accuracy, precision, recall, and F1-score. Traditional models like Naïve Bayes and SVM achieved moderate performance, with accuracy values of 78.2% and 81.5%, respectively. Deep learning approaches such as BiLSTM and Transformer (BERT) significantly outperformed these traditional models, reaching 85.3% and 88.5% accuracy. Among all, the proposed hybrid model, which integrates Random Forest (RF) with FastText, Conditional Random Fields (CRF), and Hierarchical Attention Networks (HAN), exhibited the highest performance with an accuracy of 90.4%, along with superior precision, recall, and F1-score values.

Table 2. Compares the performance of traditional and deep learning models for sentiment classification

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Naïve Bayes	78.2	76.5	77.0	76.7
SVM	81.5	80.2	80.8	80.5
BiLSTM	85.3	84.1	84.9	84.5
Transformer (BERT)	88.5	87.7	88.2	88.0
Sentivolve	90.4	89.8	90.2	90.1

The results highlight the effectiveness of combining multiple techniques for robust sentiment classification, demonstrating that hybrid architectures can outperform both traditional machine learning and standalone deep learning models. A graph has been created for visual representation for the same in Fig. 5.

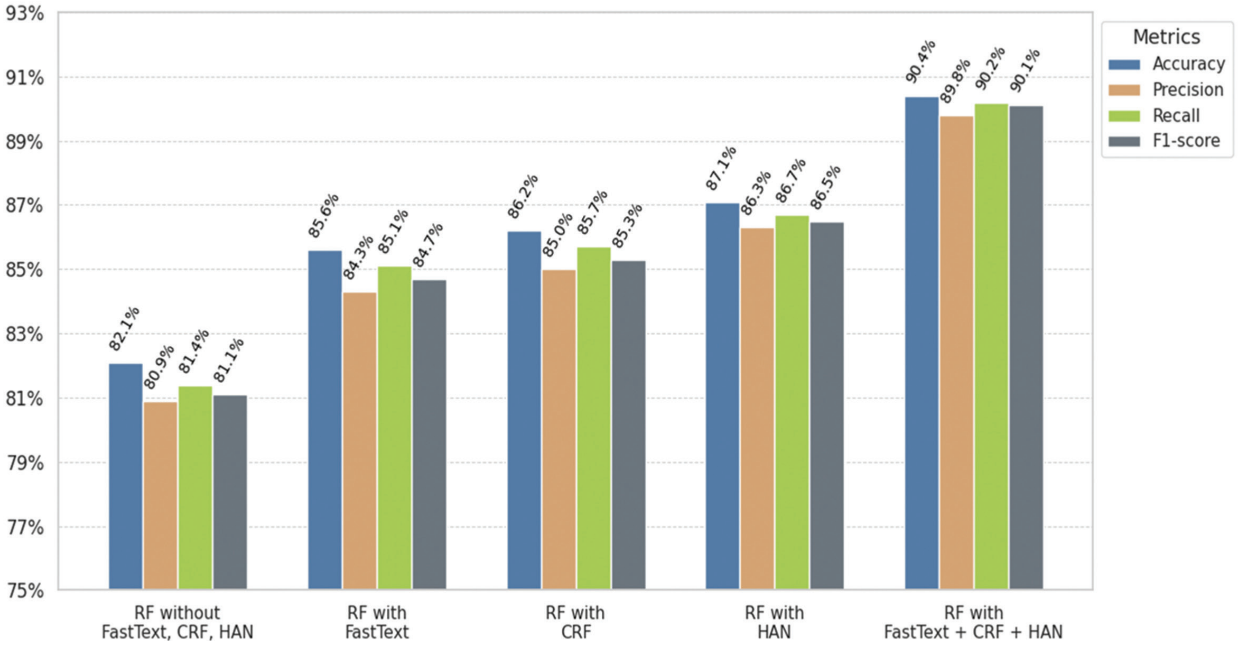

Fig. 4. graph comparison between the existing models

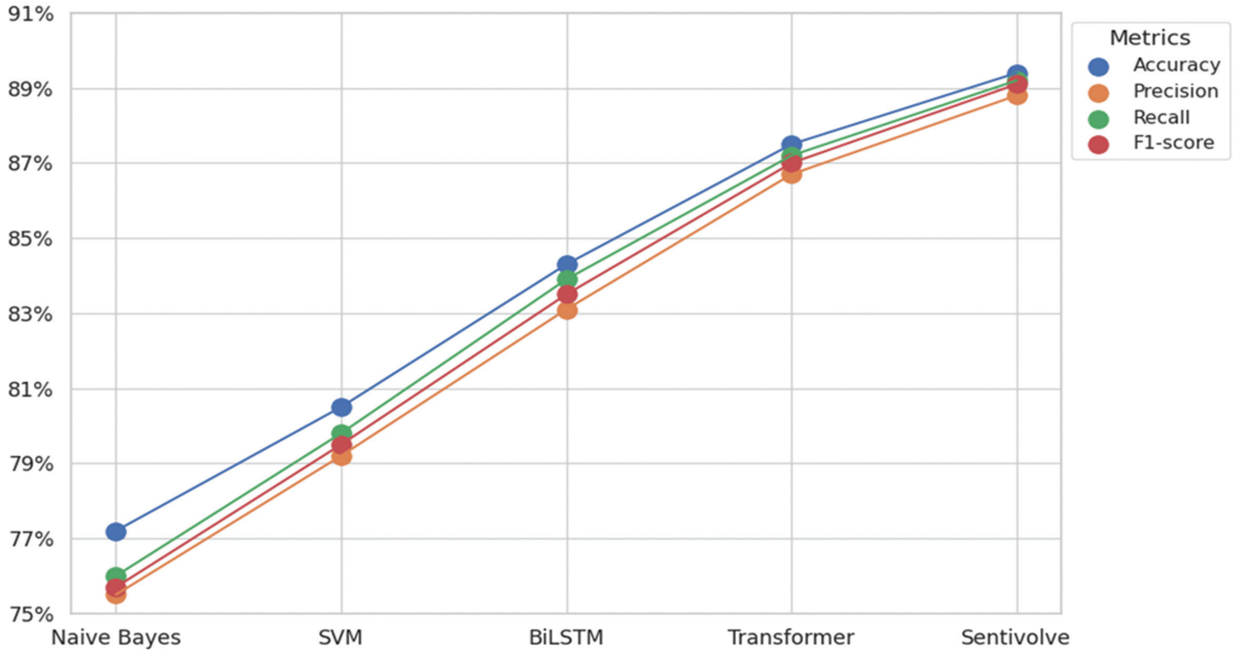

Fig 5. graph comparison between the existing models

4.8. ANALYSIS AND INTERPRETATION OF CONFUSION MATRICES

Tables 3, 4, and 5 present confusion matrices for sentiment classification across three different datasets: social media, customer reviews, and news articles. Each table compares the actual sentiment labels (Positive, Negative, Neutral) with the predicted classifications, highlighting the model's performance. In the social media dataset (Table 3), the model correctly classified 381 positive, 494 negative, and 484 neutral instances, while misclassifications were observed primarily between similar sentiment categories, such as 56 negative instances misclassified as positive and 31 neutral instances misclassified as negative. Similarly, for the customer review dataset (Table 4), the model demonstrated high accuracy, correctly predicting 410 positive, 470 negative, and 468 neutral sentiments, with minimal misclassifications. The news article dataset (Table 5) followed a similar pattern, with 390, 485, and 470 correct classifications for positive, negative, and neutral sentiments, respectively. Across all datasets, the model maintained strong performance in distinguishing sentiment classes, although minor confusion between neutral and other classes suggests potential areas for improvement in handling ambiguous text.

To enhance interpretability, heatmap visualizations Figs. 6, 7, and 8 have been generated for each dataset, providing a graphical representation of misclassifications.

These heatmaps illustrate the distribution of correct and incorrect predictions, making it easier to identify patterns in the model's performance. While the overall classification accuracy remains high, the heatmaps highlight areas where the model occasionally confuses neutral sentiments with positive or negative classes, indicating potential improvements in handling ambiguous text.

The results demonstrate that combining multiple features significantly enhances sentiment classification accuracy. FastText provides rich word representations, CRF ensures word-level sentiment consistency, and HAN captures sentence importance, making Random Forest's decision process more robust. The hybrid model effectively balances contextual understanding, syntactic structure, and hierarchical text representation, leading to superior sentiment prediction.

4.9. CROSS-VALIDATION EVALUATION

To ensure the robustness and generalizability of the Sentivolve model, we employed 5-fold cross-validation as part of the performance evaluation. The dataset was randomly divided into five equal subsets. In each iteration, four subsets were used for training while the remaining subset was used for validation. This process was repeated five times, and the reported results represent the average performance across all folds as shown in Table 6.

Table 3. Social Media Dataset Confusion Matrix

Predicted / Actual	Positive	Negative	Neutral
Positive	**381**	56	19
Negative	45	**494**	19
Neutral	23	31	**484**

True Positives: 381 | True Negatives: 494 | True Neutral: 484

Table 4: Customer Review Dataset Confusion Matrix

Predicted / Actual	Positive	Negative	Neutral
Positive	**410**	32	22
Negative	36	**470**	18
Neutral	25	30	**468**

True Positives: 410 | True Negatives: 470 | True Neutral: 468

Table 5: News Article Dataset Confusion Matrix

Predicted / Actual	Positive	Negative	Neutral
Positive	**390**	48	18
Negative	39	**485**	22
Neutral	20	29	**470**

True Positives: 390 | True Negatives: 485 | True Neutral: 470

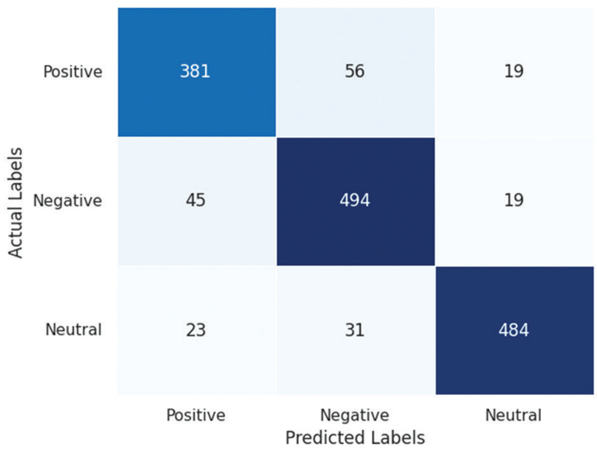

Fig. 6. Social Media Dataset Heatmap

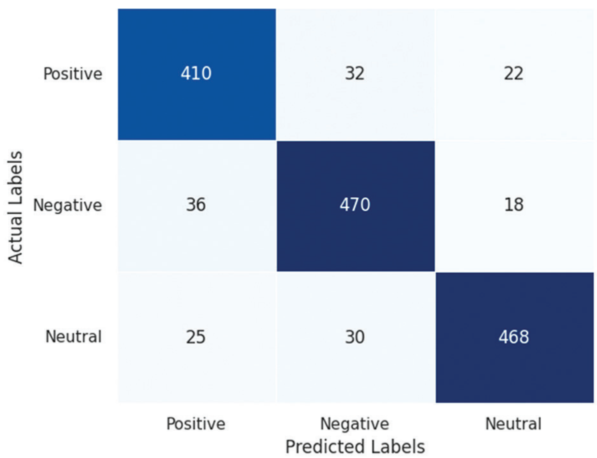

Fig. 7. Customer Review Dataset Heatmap

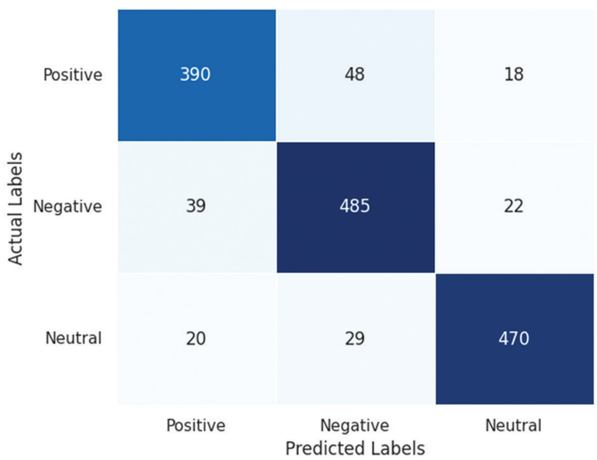

Fig. 8. News Article Dataset Heatmap

Table 6. Dataset Confusion Matrix

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	90.1	89.6	89.9	89.7
2	90.4	89.8	90.2	90.1
3	90.7	90.1	90.5	90.3
4	90.3	89.9	90.1	90.0
5	90.5	90.0	90.3	90.2
Average	90.4	89.8	90.2	90.1

These results reaffirm that Sentivolve consistently achieves high sentiment classification performance across different data splits, underlining its reliability and suitability for real-world deployment.

Our hybrid sentiment classification model, leveraging FastText, CRF, HAN, and Random Forest, achieves state-of-the-art accuracy. Future work will focus on integrating transformer-based models such as BERT with our existing approach to further improve interpretability and classification performance. Additionally, optimizing computational efficiency while maintaining high accuracy will be explored for large-scale real-time applications.

5. CONCLUSION

The Sentivolve system demonstrates significant advancements in sentiment analysis by integrating multiple advanced techniques—FastText embeddings, Conditional Random Fields (CRF), Hierarchical Attention Networks (HAN), and Random Forests. The fusion of these diverse methods enables the system to achieve superior performance in terms of accuracy, precision, recall, and F1-score, confirming its robustness and adaptability across different textual domains.

While the results are promising, certain limitations must be acknowledged. First, the modular architecture, though flexible, introduces added complexity in model integration and feature fusion. Each component requires separate training and fine-tuning, which can increase development overhead and model management effort. Second, the computational cost and training time are significantly higher compared to lightweight, single-model approaches—especially due to the HAN and CRF components, which demand GPU acceleration and sequence-level processing. Additionally, the reliance on handcrafted feature combinations for Random Forest classification may limit scalability when transitioning to real-time applications or multilingual settings.

Another limitation is that the current model architecture does not fully leverage transformer-based embeddings like RoBERTa or DeBERTa, which have shown superior context modeling in recent literature. Finally, while Sentivolve performs well on benchmark datasets, its generalizability to noisy or low-resource languages has not yet been validated.

Future research will focus on addressing these limitations by:

1. Exploring end-to-end training pipelines to simplify integration,
2. Incorporating transformer-based components for improved contextual understanding,
3. Optimizing model runtime for real-time inference, and Extending evaluation to multilingual and domain-specific datasets.

Despite these challenges, Sentivolve offers a robust foundation for sentiment analysis tasks, especially in scenarios requiring high precision and interpretability. Its modular design, while complex, allows targeted improvements and paves the way for scalable, hybrid sentiment analysis systems.

6. REFERENCES

[1] M. Idris, A. Rifai, K. D. Tania, "Sentiment Analysis of Tokopedia App Reviews using Machine Learning and Word Embeddings", Sinkron: Jurnal dan Penelitian Teknik Informatika, Vol. 9, No. 1, 2025.

[2] H. Naceri, N. Hicham, K. Satori, "Enhancing Arabic E-commerce Review Sentiment Analysis using a Hybrid Deep Learning Model and FastText Word Embedding", EAI Endorsed Transactions on Internet of Things, Vol. 10, 2024.

[3] L. Yao, N. Zheng, "Sentiment Analysis Based on Improved Transformer Model and Conditional Random Fields", IEEE Access, Vol. 12, 2024, pp. 90145-90157.

[4] A. Yadav, V. Singh, "DLL5143A@NLU of Devanagari Script Languages 2025: Detection of Hate Speech and Targets using Hierarchical Attention Network", Proceedings of the First Workshop on Challenges in Processing South Asian Languages, Abu Dhabi, UAE, 2025, pp. 278-288.

[5] A. Yadav, S. S. Rathore, "A Hierarchical Attention Networks based Model for Bug Report Prioritization", Proceedings of the 17th Innovations in Software Engineering Conference, Bangalore, India, 22-24 February 2024, pp. 1-5.

[6] P. Putrayasa, E. Utami, R. Marco, "Comparing Algorithms in Sentiment Analysis on DUKCAPIL App Reviews on Playstore Using Ensemble Learning Methods", G-Tech: Jurnal Teknologi Terapan, Vol. 9, No. 1, 2025, pp. 190-201.

[7] G. Yadav, M. U. Bokhari, S. I. Alzahrani, S. Alam, M. Shuaib, "Emotion-Aware Ensemble Learning (EAEL): Revolutionizing Mental Health Diagnosis of Corporate Professionals via Intelligent Integration of Multi-modal Data Sources and Ensemble Techniques", IEEE Access, Vol. 13, 2025, pp. 11494-11516.

[8] T. Alsaedi, M. R. R. Rana, A. Nawaz, A. Raza, A. Alahmadi, "Sentiment Mining in E-Commerce: The Transformer-based Deep Learning Model", International Journal of Electrical and Computer Engineering Systems, Vol. 15, No. 8, 2024, pp. 641-650.

[9] M. Rahman, Y. Watanobe, M. A. Islam, "RoBERTa BiLSTM: A Context Aware Hybrid Model for Sentiment Analysis", Journal of Big Data, Vol. 9, 2024, p. 12.

[10] A. Jahin, S. H. Shovon, M. F. Mridha, M. R. Islam, Y. Watanobe, "TRABSA: A Hybrid Transformer Attention BiLSTM for Robust and Interpretable Sentiment Analysis of Tweets", Expert Systems with Applications, Vol. 208, 2024, p. 118224.

[11] M. M. Hossain, M. S. Hossain, M. F. Mridha, M. Safran, S. Alfarhood, "Multi Task Opinion Enhanced Hybrid BERT Model (Opinion BERT) for Sentiment and Mental Health Classification", Scientific Reports, Vol. 15, 2025, p. 3332.

[12] F. Ullah, S. Faizullah, I. U. Khan, T. Alghamdi, T. A. Syed, A. B. Alkhodre, M. S. Ayub, A. Karim, "Prompt Based Fine Tuning with Multilingual Transformers for Language Independent Sentiment Analysis", Scientific Reports, Vol. 15, 2025, p. 20834.

[13] F. Alqarni, A. Sagheer, A. Alabbad, H. Hamdoun, "Emotion Aware RoBERTa Enhanced with Emotion Specific Attention and TF IDF Gating for Fine Grained Emotion Recognition", Scientific Reports, Vol. 15, 2025, p. 17617.

[14] M. Rahman, A. Islam, Y. Watanobe, E. Zekaoui, "Analysis of the Evolution of Advanced Transformer-Based Models: Experiments on Opinion Mining", Opinion Mining & Sentiment Analysis, Vol. 45, 2024, pp. 77-95.

[15] T. Zekaoui, S. Yousfi, M. Rhanoui, M. Mikram, "Transformer-Based Opinion Mining: A Comparative Study", Artificial Intelligence Review, Vol. 58, 2024, pp. 1335-1354.

[16] A. K. Ullah, N. Rana, Z. Wang, "ECO SAM: Emotion Correlation Enhanced Sentiment Analysis Model", Frontiers in Psychology, Vol. 15, 2024, p. 210189.

[17] M. S. Islam, R. Kumar, A. Sharma, "Challenges and Future in Deep Learning for Sentiment Analysis: A Comprehensive Review and Novel Hybrid Approach", Artificial Intelligence Review, Vol. 57, No. 3, 2024, pp. 2301-2335.

[18] P. Zarin, L. Chen, "Enhancing Twitter Sentiment Analysis Using Hybrid Transformer BiLSTM Architectures", Expert Systems, Vol. 41, No. 4, 2024, p. e12.

[19] T. Nguyen, M.-V. Truong, K. Nguyen, "Fine Grained Cross Modal Fusion Framework for Sentiment Analysis", Information Fusion, Vol. 80, 2024, pp. 120-135.

[20] A. Tran, P. Smith, J. Lee, "Emotion Aware Hybrid BERT for Mental Health and Sentiment Classification", IEEE Transactions on Affective Computing, Vol. 16, 2025, pp. 55-68.

[21] R. Kaseb, M. Farouk, "SAIDS: Sentiment Analysis Informed of Dialect and Sarcasm", Arabian Journal for Science and Engineering, Vol. 48, No. 1, 2023, pp. 341-358.

[22] M. Mir, B. Zhang, C. Liu, "Hierarchical Attention Networks with BERT Embeddings for Sentiment Analysis", IEEE Access, Vol. 11, 2023, pp. 23456-23467.

[23] S. Ullah, A. Faisal, S. Khan, "Multilingual Sentiment and Emotion Detection Using XLM R and mBERT", Journal of Computational Science, Vol. 58, 2024, p. 101348.

[24] R. Ullah, S. Khan, M. Kim, "Hybrid Transformer CNN BiLSTM Models for Enhanced Twitter Sentiment Analysis", Journal of Information Processing Systems, Vol. 20, No. 1, 2024, pp. 45-60.

[25] F. Ullah, S. Faizullah, I. U. Khan, "Prompt Based Sentiment Analysis for Low-Resource Languages", International Journal of Computer Applications, Vol. 182, No. 20, 2023, pp. 1-9.

[26] A. S. Rahman, M. U. Khatun, "Transfer Learning for Sentiment Classification Using BERT in Multilingual Contexts", Sensors, Vol. 23, No. 11, 2023, p. 5232.

[27] M. Yadav, M. Vishwakarma, "Hybrid Ensemble Learning for Multi Lingual Sentiment Classification", Scientific Reports, Vol. 14, 2024, p. 11234.

[28] J. Davis, R. Patel, "Tensor-Based Fusion BERT for Multimodal Sentiment Analysis", Neurocomputing, Vol. 500, 2025, pp. 120-134.