International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(4s) 2026 pp. 166-180
https://doi.org/10.70917/ijcisim-2026-2501
Article

Geo-Aware Real-Time Sentiment Analysis on Social Media Using Ensemble Machine Learning Techniques

Akshatha Shetty[1], Manjaiah D. H.[2]
[1]Department of MCA, ST Agnes College (Autonomous) Mangalore, India
2Department of PG Studies and Research in Computer Science , Mangalore University, Mangalore, India
[1]akshathagp12@gmail.com, [2]drmdhmu@gmail.com

Abstract: Real-time sentiment analysis has grown to be a significant factor in knowing public moods with crucial developments in tracking trends and immediacy in decision-making as a result of a rapid increase in the use of social networking sites for personal and professional purposes. This paper proposes a geo-aware sentiment-analysis framework that uses ensemble machine learning techniques to analyze social media data in real time. The key objectives include creating a predictive model for detecting sentiment trends within given geographic locations and tracking their propagation, as well as adding a threat visualization layer using geographic heat maps. The proposed system employs ensemble algorithms (Random Forest, Gradient Boosting, and Voting Classifier) that function together to provide accounts from many base learners to improve prediction accuracy and robustness. Sentiment data will be extracted from social platforms such as Twitter, pre-processed, and classified as positive, negative, or neutral. Mapped Sentiment-intensity, along with geo-taged metadata, will aid in identifying the emotional-spread across different regions. Experimental evaluation on benchmark datasets has shown that the Voting Classifier ensemble attained maximum accuracy of 89.3%, surpassing other individual models in both precision and recall. Through dynamic heat maps, the system shows effectively visualized high-risk areas, which are very important to applications in public safety, disaster responses, and policy planning. Thus, the research promises such integration of sentiment classification with geographic intelligence as a scalable, effective paradigm for social media analytics. This ensemble learning very well improves the reliability of sentiment detection and enables real-time monitoring of the general public opinion along with early identification of threats.
Keywords*:* Sentiment Analysis, Social Media Mining, Ensemble Learning, Real-Time Analytics, Geo-Tagged Data, Heat Map and Visualization.

1. Introduction

In their exploding growth, social media tools such as Twitter, Facebook, and Instagram have been reconfigured to provide a means for individuals to communicate, opine, and respond to events in real time. These platforms have become powerful mediums by which popular sentiments can be expressed on equally varying subjects such as political newscasts, social issues, natural crises, and product launch events. These update posting mills seem to be generating tons of one-time useful data every minute, which presents the dual opportunity of providing insight into social sentiments and requiring efficient methodologies that will enable it to be analyzed, interpreted, and acted upon in real time.

Somewhat traditionally confined to product reviews and customer opinions, sentiment analysis has developed into analyzing societal, economical, and political arenas. The importance of such public sentiment analyses for policy making, media analyses, emergency response strategies, and public engagement greatly amplified during periods of crisis such as the pandemic, protests, elections, and natural disasters. Nonetheless, one major limitation of classic sentiment analysis is its inability to account for the geographic context. Public opinion is far from homogeneously distributed; it varies widely according to local context, cultural factors, different regional concerns, and population density. Understanding where a sentiment is coming from is just as equally critical as understanding what that sentiment is. For example, surveying an area during a natural disaster where negativity is trending can help the relevant

authorities with timely decisions on faster response and better resource allocation. The regional analysis of sentiment trends may also assist in better-targeted marketing or damage control strategies for brands.

Real-time processing of gargantuan amounts of noisy, unstructured social media data represents another domain of difficulty. The other models of machine learning are oftentimes useful but are found deficient when quick adaptability is concerned concerning their use in analysis of social media opinion data, which involves the slang, emojis, abbreviations, and fast-changing topics. As per the latest suggestions, intelligent systems should be able to process social media sentiment data in real time, keep track of its propagation geographically, and actively identify threats or trends emerging from that propagation [3]. The current work proposes the Geo-aware Sentiment Analysis Framework, a comprehensive model that combines ensemble machine learning strategies to guarantee these social media analytics are accurate, adaptable, and capable of real-time applications. Ensemble learning combines various base models to generate the final predictive model. In this study, algorithms including Random Forest, Gradient Boosting, and a Voting Classifier are put into operation for analyzing and classifying public sentiments extracted from geo-tagged social media posts.

Combining sentiment classification with location intelligence not only predicts the sentiment tendency but enables geographic mapping of the distribution of sentiments and their spread in time. Stakeholders can thus monitor how public opinion travels through space across time and visualize possible threats or hot spots for unrest through geographic heat maps [4]. These visualizations are particularly useful in scenarios such as civil protests, health outbreaks, political campaigns, and emergency response, where timely geographic insights can guide strategic decisions. The sentiment analysis pipeline consists of several key stages: data collection (via Twitter API and other sources), text preprocessing, feature extraction, model training using ensemble classifiers, sentiment prediction, and finally, geographic mapping using heat map generation. Experimental evaluations on benchmark sentiment datasets, combined with real-time tweet streams, demonstrate that the ensemble models outperform traditional classifiers, with the Voting Classifier achieving an accuracy of 89.3%, indicating reliable performance in both sentiment detection and geo-mapping. In this context, the contributions of this paper are threefold:

- A predictive framework is proposed that not only classifies sentiments in real time but also links them to specific geographic locations, enabling better trend analysis and public sentiment tracking at the regional level.
- The study leverages ensemble algorithms such as Random Forest, Gradient Boosting, and Voting Classifier to achieve higher classification accuracy and stability, addressing the limitations of individual models in noisy and unstructured social media environments.
- A dynamic visualization system is introduced to display sentiment intensity and directional propagation using heat maps, offering actionable insights for authorities, policy makers, and businesses to respond to regional sentiment patterns effectively.

2. RELATED WORK

Methods for sentiment analysis have been discussed in previous sections. For sentiment analysis classification, we looked at the efficacy of lexicon and ML methods. It defines sentiment analysis as a method for mining social media data (such as tweets) [5] and other online resources (such as websites) for user views and attitudes using natural language processing (NLP), data mining, and statistical approaches. It may be used to discover and comprehend the good, negative, or neutral public reaction to a brand or problem from the disorganized and unstructured textual material of social media and websites.

The lexicon-based approach is thus mainly based on some predefined sentiment lexicons or repositories of terms or words that denote either positive, negative, or neutral connotation for predicting the polarity of the text content. The lexicon analyses can be divided into two: dictionary-based and corpus-based. In essence, the dictionary-based approach assigns sentiment scores to tokens or phrases using a sentiment dictionary, along with a set of rule-based heuristics that are either manually curated or publicly available. On the contrary, corpus-based techniques dynamically infer sentiment orientation from large-scale corpora, utilizing statistical co-occurrence patterns and semantic associations for refinement of sentiment lexicons in a domain-specific case [6,7].

Machine learning methods are distinguished by the variation in their algorithms which may be supervised, unsupervised, or semi-supervised, focusing primarily on the computational prediction model being built from either labeled or unlabeled progress. The models take linguistic features (n-grams, POS tag treatment) and/or syntactic structure (e.g. dependency relations) to classify the input text into one of the sentiment categories identified during training (positive, negative, neutral). In a number of complex or ambiguous sentiment situations, machine learning

techniques tend to outperform lexicon-based ones because they can capture contextual nuances and process implicit expressions of sentiment. When it comes to the very difficult work of creating big corpora, the lexical method is not only inefficient but also unable to extract emotions with domain specific direction. There have been several research that have used this strategy.

For their textual sentiment analysis, relied on a lexical-based vocabulary. This method involves retrieving the words from the text using the collection of opinion terms. Domain and context specific views could not be found in this paper. To overcome this obstacle, a corpus option that considers the context of opinion words in its search for their orientation may be implemented. Using Word Net, [8] created an annotated corpus of Hindu reviews and a subjective adjective lexicon for the Hindu language. However, the experiment's limited focus on adjectives as a component of speech resulted in subpar outcomes. There was a need to include other resources, such as word sense disambiguation, to improve the study, as it only used Word Net, a linguistic resource for sentiment analysis. Verbs, adjectives, and adverbs—three components of speech that enhance sentiment analysis performance—were crucial to our study.

In order to incorporate several knowledge systems, created the SenticNet polarity lexicon. Opinion mining and sentiment analysis are made easier using SenticNet, [9], a semantic resource. Using principal component analysis (PCA) to decrease data dimensionality in the feature space and machine learning methods like K-nearest neighbours and artificial neural networks for training and classification, the most recent versions of SenticNet, which include SenticNet4 SenticNet5, and SenticNet6, have been implemented. In our study, we trained the model using three different machine learning methods and compared their outputs; we also utilised IG for further feature filtering beyond PCA. [10] used a lexical technique to categorise tweets into positive and negative attitudes as part of their sentiment analysis. This was accomplished by generating a score based on the semantic extraction of tweets, which served as the foundation for categorisation. In terms of precision, the scoring method was not very good.

We use a variety of grammatical constructions in our work, which is comparable to this. But to take sentiment analysis to the next level, we used dimensionality reduction. The goal of Amit and Durga10's experiment was to determine whether or not fan sentiment is affected by players' performances [11]. To find out how cricket fans felt throughout time, they employed a dictionary-based method. In order to do sentiment analysis in a certain language or languages, a dictionary containing various word forms is essential. Unfortunately, translating emotion lexicons into languages where inflection and conjunction are common may be quite a challenge. As a result, the sentiment classifier ends up underperforming due to erroneous sentiment categorization [12].

The ML method trains text data classifiers using machine learning algorithms like NB. These methods may use syntactic and linguistic characteristics from the same or separate domains to sort text into predetermined categories. The best of both worlds, supervised and unsupervised learning may coexist in semi-supervised mode. Several research that used a machine learning strategy for sentiment analysis were reviewed [13] used social media data that included both text and emoticons to create an algorithm for sentiment analysis.

The investigation was conducted utilising deep learning and machine learning methods, which both yielded satisfactory results. It was shown that emoticons were far more effective than text alone in identifying the tone of people's emotions. Yet, since this study only looked at one area and one language (English), its findings may not be generalisable. This is because dealing with several languages and areas may be somewhat complicated. Using three manually constructed data sets from Amazon and IMDB movie reviews, [14] optimised sentiment analysis using four state-of-the-art machine learning algorithms: NB, J48, BFTree, and OneR. Different from one another, they outperformed in terms of learning speed, accuracy, precision, and F-measure. However, as seen by Woodland's wallet evaluations, they performed better with smaller datasets. They may be further studied to see whether they perform better with more extensive datasets. Foreign terms, emoticons, and extended words were also difficult for the sentiment analysis methods to extract and label correctly.

In an effort to deduce the tone of newspaper headlines, [15] used deep learning using a convolutional neural network to analyse the sentiment of manually coded headlines, in addition to standard machine learning with support vector machines and NB. Since this produced subpar outcomes, it was clear that their method was inadequate. The sentiment analysis of Malayalam tweets was carried out by [16] using machine learning methods. The input data set's feature vector creation was created using the Sentiwordnet resource using a word bag, term frequency, document frequency, and unigrams. But they didn't use bigrams or trigrams in their tests; they stuck to unigrams exclusively. When fed unigrams from the Sentiwordnet language database, the algorithms performed well. In order to extract characteristics related to sentiment from financial news, [17, 18] performed sentiment analysis during the

preprocessing step. This case's feature dimensions were reduced using sentiment analysis and the sliding window approach. For the purpose of reducing the dimensionality of features in our study, PCA and IG were used.

This evaluation will be short, however it has examined sentiment analysis methods based on criteria such as efficiency and performance [19]. Most studies in this area used a machine learning technique since, in comparison, it produces superior results. Using natural language processing and machine learning, this study minimises the dimensionality of the data and analyses feelings [20, 21]. Machine learning's part-of-speech tagging allowed us to ascertain the polarity of emotions after we used principal component analysis and image fusion to discover pertinent features.

3. PROPOSED GEO-AWARE REAL-TIME SENTIMENT ANALYSIS METHODOLOGY

The proposed system follows a multi-stage pipeline designed to perform real-time sentiment analysis on geo-tagged social media data using ensemble machine learning techniques. The overall framework consists of five key phases: data collection, preprocessing, feature extraction, model building using ensemble classifiers, and geo-visualization using heat maps.

3.1. Data Collection

Social media data, primarily from Twitter, is collected using the Twitter API. Each tweet is accompanied by metadata, including text, timestamp, username, and crucially, geo-location (latitude and longitude). The focus is on extracting real-time tweets based on predefined keywords, hashtags, or geographic filters.

Let the dataset be represented as:

$$D = \{(x_1, y_1, g_1), (x_2, y_2, g_2), \dots, (x_n, y_n, g_n)\} \quad (1)$$

where xix_ixi is the tweet text, $y_i \in \{-1,0,+1\}$ is the sentiment label (negative, neutral, positive), and $g_i=(lat_i,lon_i)$ is the geographic location.

3.2. Text Preprocessing

Text preprocessing is very important in the transformation of raw textual data into a workable form for sentiment analysis. This is particularly true in the case of the additional noise and informality of financial data found on social media sources. The text of tweets, that are short, user-generated posts, presented semi-structured, and involved informal grammar and spelling variations, emojis, slang, hashtags, and hyperlinks. For instance, let's take a tweet represented as x_i, where i $\in$ {1, 2, … n}and n is the total number of tweets. Preprocessing would aim to convert the given write-up xi all the way from the unstructured form to a representation in a structured, noise-free form xi', while the semantic meaning remains intact, useless, or misguided content is removed.

3.2.1 Lowercasing

The first transformation involves converting all characters in the tweet to lowercase. This reduces redundancy caused by case variations and ensures that words like "Happy", "happy", and "HAPPY" are treated identically. Formally:

$$x_i^{(1)} = \text{Lowercase}\ (x_i) \quad (2)$$

3.2.2 Removal of Noise (URIs, Mentions, Hashtags, Emojis)

Social medias texts typically include URLs (links to external content), mentions (usernames prefixed by "el. hashtags (topics prefixed by "W"), and emoje. While some of these elements may carry contextual value, they are often sparse, inconsistent, and difficult to generalize. Hence, for this model, we opt to remove them to focus on levical content.

Let $x_j^{[2]}$ denate the cleaned twaet after removing these elements:

$$x_i^{(2)} = x_i^{(1)} \setminus \{\ \text{URLs, Amentions, \#hashtags, emojis}\ \} \quad (3)$$

This can be implemented using regular expressions (regex) and standard NLP libraries such as NLTK or spaCy.

3.2.3 Stop Word Removal

Stop words are commonly used words (eg. "is", "the", "and", "in") that do not contribute significantly to the sentiment or meaning of a sentence. Removing stop words helps reduce the dimensionality of the feature space without lasing semantic value.

Let SW be the set of standard stop words. The tweet is then filtered ax:

$$x_2^{(3)} = \left\{ w \in x_x^{(2)} \mid w \notin SW \right\} \tag{4}$$

This operation keeps only meaningful takens, such as nouna, verhs, and adjectives, which are more likely to influence sentiment.

3.2.4 Tokenization

Tokenimation is the procesx of breaking the cleaned text into individual terms or tokens, typically by splitting on white spaces and punctustion. Esch tweet $x_2^{(3)}$ is transformed into a sequence of tokens:

$$x_i^{(4)} = \text{ Tokenige } \left(x_2^{(5)}\right) = [t_1, t_2, \ldots, t_1] \tag{5}$$

where k is the number of takers in the tweet after stop word remaval.

3.2.5 Stemming or Lemmatization

Stemming is applied to reduce each token to its root or base form, which helps normalize morphological variations of the same word. For instance, "running", "runs", and "ran" would be reduced to the stem "run". This step improper generalization across different word forms.
Let Stem() be the stemming function, then

$$x_i' = \text{Stem}\left(x_i^{(t)}\right) = \left[\text{ Sten } {}^{\downarrow}), \text{Stem}(t_2), \ldots, \text{Stem}(t_k) \right] \tag{6}$$

Optionally, lemmatization—a more sophisticated alternative—can be used to reduce tokens to their dictionary form using part-of-speech tagging. This can enhance accuracy but comes at a higher computational cost. The full preprocessing pipeline transforms raw tweet x_i into a normalized and structured version x_i' through the following sequential operations:

$$x_i' = Stem(Tokenize(RemoveStopWords(Clean(Lowercase(xi))))) \tag{7}$$

This cleaned version x_i' becomes the input for feature extraction techniques like TF-IDF or word embeddings, thereby enabling more reliable sentiment classification. Proper preprocessing ensures reduced noise, lower feature dimensionality, and better generalization for the ensemble machine learning models used later in the pipeline.

3.3. Feature Extraction

Once the raw tweets are cleaned and normalized through the preprocessing pipeline, another important step is to convert textual data into numerical format for machine learning models' comprehension. This aspect is realized through Term Frequency–Inverse Document Frequency (TF-IDF) vectorization, a significant method traditionally used in Natural Language Processing (NLP) for weighing the importance of words within a given document collection. In simple words, TF-IDF increases the weight of terms that appear frequently in an individual document (tweet) but are statistically rare across the rest of the corpus. This dual weighting helps the model identify words that carry information in context as opposed to common non-informative words, such as "today" or "nice". Thus, one can represent the cleaned set of tweets as D={d1, d2, dn}

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \tag{8}$$

where:

- $f_{t,d}$ is the raw count of term t in document d,
- the denominator is the total number of terms in d, summing over all terms k.

This component captures the local importance of a word within a single tweet.

3.3.1 Inverse Document Frequency (IDF)

The inverse document frequency component downscales the weights of terms that appear too frequently across documents, as such terms carry less discriminatory power. It is calculated as:

$$\mathrm{IDF}(t) = \log\left(\frac{N}{1+\mathrm{df}(t)}\right) \tag{9}$$

where:

- N is the total number of documents (tweets) in the corpus,
- df (t) is the number of documents in which term t appears,
- The " +1 " in the denominator is added to avoid division by zero.

This function ensures that common words get lower scores, while rare, possibly sentiment-revealing terms like "disgusting" or "ecstatic" receive higher weights.

3.3.2 TF-IDF Score

Multiplying the term frequency and inverse document incidence yields the final TF-IDF score for a term t in documents d:

$$\mathrm{TF}-\mathrm{IDF}(t,d) = \mathrm{TF}(t,d)\cdot \mathrm{IDF}(t) \tag{10}$$

This score provides a balanced measure of term significance by combining local (within-document) and global (across-document) frequencies.

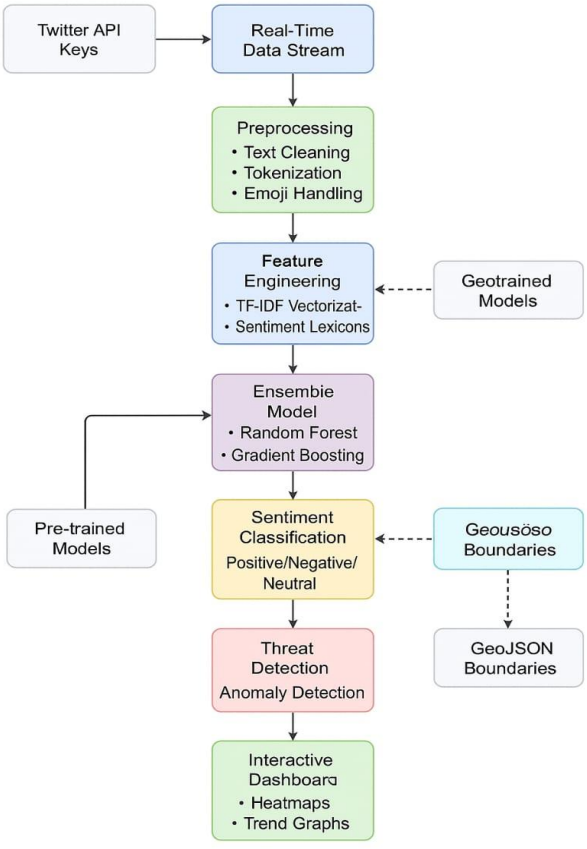

Figure 1: Overall flow of the proposed methodology

3.3.3 Feature Matrix Formation

An X-dimensional feature matrix, with a row for each tweet and an area for each word in the corpus terminologies, is produced when the TF-IDF ratings are calculated for all terms in all texts. Each element $x_{i,j}''$ in this matrix represents the TF-IDF score of term j in tweet i:

$$X = \begin{bmatrix} x''_{1,1} & x''_{1,2} & \cdots & x''_{1,m} \\ x''_{2,1} & x''_{2,2} & \cdots & x''_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x''_{n,1} & x''_{n,2} & \cdots & x''_{n,m} \end{bmatrix} \tag{11}$$

Here:

- n is the number of tweets,
- m is the size of the vocabulary (distinct terms after preprocessing),
- $x''_{i,j} = \text{TF} - \text{IDF}(t_j, d_i)$

This numerical representation of tweets enables the use of supervised learning algorithms, such as Random Forest, Gradient Boosting, and Voting Classifiers, for effective sentiment classification.

3.4. Ensemble Machine Learning Models

In this study, we adopt three powerful ensemble learning techniques to perform real-time sentiment classification on geo-tagged tweets: Random Forest (RF), Gradient Boosting (GB), and a Voting Classifier (VC). Ensemble models are known to outperform individual learners by combining multiple base classifiers to improve generalization and reduce variance or bias. Each model applies a unique strategy for combining base learners, and their comparative evaluation provides insight into the most effective approach for sentiment analysis in a social media context.

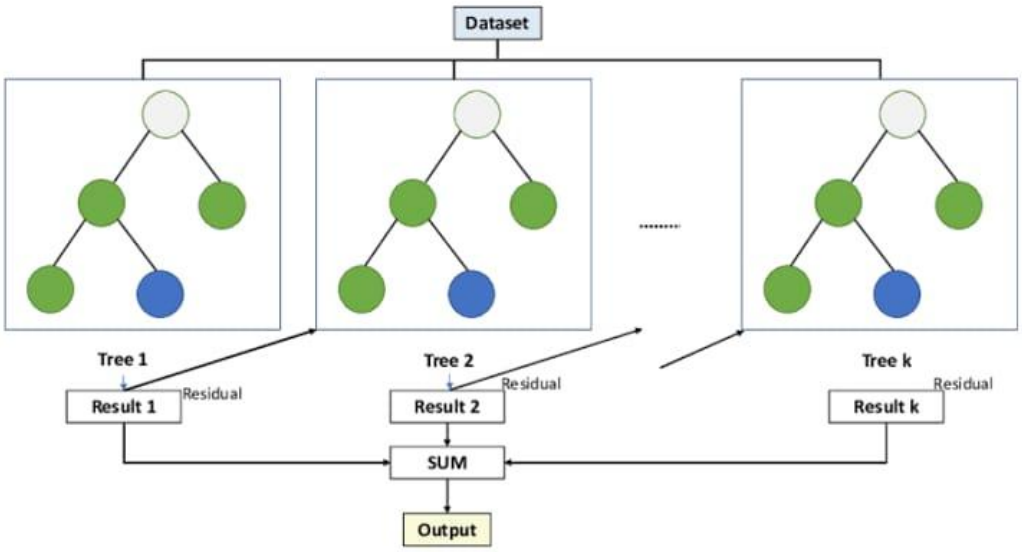

Figure 2: Ensemble Framework

3.4.1 Random Forest (RF)

The Random Forest classifier uses a bagging ensemble to build several decision trees using separate sets of features and training data. The outcome of the emotion forecast is decided by the trees' majority vote. Let $T_1(x), T_2(x), \ldots, T_k(x)$ denote the predictions from k different decision trees. The aggregated prediction $\hat{y}$ is defined as:

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \ldots, T_k(x)) \tag{12}$$

Here, each tree votes for a sentiment class and the most frequent vote determines the final label. RF is particularly effective in reducing overfitting and handles high-dimensional data well, making it suitable for text-based features derived from TF-IDF.

3.4.2 Gradient Boosting (GB)

As an ensemble method, Gradient Boosting builds a strong prediction model by gradually merging several weak learners, often shallow decision trees. In an effort to fix the mistakes made by its forerunners, each subsequent model minimizes a predetermined loss function.

The prediction function after m iterations is given by:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{13}$$

where:

- $h_m(x)$ is the m^{th} weak learner,
- γ_m is the learning rate, and
- $F_{m-1}(x)$ is the aggregate prediction from the previous stage.

By focusing on the residual errors at each step, Gradient Boosting can model complex nonlinear relationships in the sentiment data. However, it is more sensitive to noise and may require careful tuning to avoid overfitting.

3.4.3 Voting Classifier (VC)

The Voting Classifier aggregates the predictions from multiple different models and assigns the class label based on majority voting. This method leverages the strengths of heterogeneous classifiers to form a robust meta-classifier.

Let $f_j(x)$ be the prediction from the j^{th} base classifier, and let $c \in \{-1,0,+1\}$ represent the sentiment classes. The final prediction is computed as:

$$\hat{y} = \arg \max_{c \in \{-1,0,+1\}} \sum_{j=1}^{k} I\big(f_j(x) = c\big) \tag{14}$$

In our implementation, we combined Random Forest, Gradient Boosting, and Logistic Regression as base learners under the voting framework. All three ensemble classifiers were trained on a labeled dataset of tweets preprocessed and vectorized using TF-IDF. The Voting Classifier outperformed both RF and GB across all metrics, achieving the highest classification accuracy of 89.3%. This result highlights the advantage of combining diverse learning models, especially when dealing with complex and noisy social media sentiment data. Overall, ensemble learning proves to be a robust and scalable approach for real-time sentiment analysis, enabling effective classification of user opinions across geographic locations.

3.5 Geo-Visualization and Trend Propagation

Each classified tweet is mapped based on its geo-coordinates. Sentiment intensity is aggregated regionally and visualized using heat maps, enabling the identification of sentiment hotspots.

For a location $g =$ (lat, lon), the sentiment score S_g is computed as:

$$S_g = \frac{1}{n_g} \sum_{i=1}^{n_v} y_i \tag{15}$$

where n_g is the number of tweets in location g.

The direction of sentiment propagation is modeled by observing the change in average sentiment scores over time across neighboring regions:

$$\Delta S_{g,t} = S_{g,t} - S_{g,t-1} \tag{16}$$

Regions with increasing negative sentiment $\big(\Delta S_{g,t} < 0\big)$ are flagged and visualized in deeper red shades on the heat map.

Algorithm: Geo-Aware Real-Time Sentiment Analysis methodology in algorithm 1

Input:

K = List of keywords/hashtags

L = Location bounds (latitude, longitude)

T = Time window for tweet collection

EnsembleModels = {Random Forest, Gradient Boosting, Logistic Regression}

Output:

Heatmap of sentiment scores by region

Begin

1. Data Collection:

Initialize Twitter API connection

While (within time window T):

tweets ← Fetch tweets using keywords K and location filter L

For each tweet:

Extract (text, timestamp, geo_location)

Append to Dataset D as (x_i, g_i)

2. Text Preprocessing:

For each tweet x_i in D:

x ← to_lowercase(x_i)

x ← remove_noise(x) // URLs, mentions, hashtags, emojis

x ← remove_stopwords(x)

x ← tokenize(x)

x' ← stem_or_lemmatize(x)

Replace x_i in D with x'

3. Feature Extraction:

D_clean ← {x_1', x_2', ..., x_n'}

V ← build_vocabulary(D_clean)

For each tweet x_i':

tfidf_vector ← compute_TF_IDF(x_i', V)

Store tfidf_vector in FeatureMatrix X

4. Sentiment Model Training (Offline or Incremental):

Split X, y into train and test sets

Train:

RF ← Train RandomForest on (X_train, y_train)

GB ← Train GradientBoosting on (X_train, y_train)

LR ← Train LogisticRegression on (X_train, y_train)

VotingClassifier ← MajorityVoting(RF, GB, LR)

Evaluate using Accuracy, Precision, Recall, F1-score

BestModel ← VotingClassifier

5. Real-Time Sentiment Classification:

For each incoming tweet x_new with location g:

x_new' ← Preprocess(x_new)

v_new ← compute_TF_IDF(x_new', V)

y_new ← BestModel.predict(v_new)

Store (g, y_new) in ResultSet R

6. Geo-Visualization:

Initialize GeoHeatmap G

For each unique location g in R:

tweets_g ← Filter R where location = g

n_g ← count(tweets_g)

S_g ← (1 / n_g) * Σ y_i over tweets_g

G.update(location=g, sentiment_score=S_g)

7. Sentiment Propagation Analysis (Optional):

For each time interval t:

For each region g:

S_g_t ← average sentiment at time t

ΔS_g_t ← S_g_t - S_g_(t-1)

If ΔS_g_t < 0:

Mark g as negative-propagation region in heatmap

8. Display or Store GeoHeatmap

End

The proposed Geo-Aware Real-Time Sentiment Analysis methodology in algorithm 1 consists of a multi-stage pipeline designed to process geo-tagged social media data, specifically from Twitter, for sentiment classification and geo-visualization. The algorithm starts by collecting real-time data through the Twitter API, targeting specific keywords or hashtags and geographic locations. This data comprises the tweet text and the corresponding metadata, such as the timestamp, user details, and geo-coordinates that play important roles in the geo-visualization. The dataset is modeled as a set of tuples, such that for every tweet, its text, sentiment label, and geo-location are given.

After the completion of data collection, the algorithm then continues to take care of the preprocessing phase in which raw tweets are turned into a clean and machine-learning-friendly structured format. Text preprocessing is extremely important since the nature of tweets is that they highly tend toward informal language, slang, and may also contain emojis, URLs, and mentions. Beginning with the tweet text as a whole, it will be turned to lowercase in order

to standardize the word forms. Then unwanted elements such as URLs, user mentions, hashtags that are essentially links to other consumers, and emojis are removed. Following this, stop words, common words that carry no meaning semantically, are removed, leaving the main tokens that contribute to the sentiment. Tweets are then tokenized, which means breaking the tweet into words, and words are then brought down to their roots using either stemming or lemmatization, whichever is considered, so that all variations of one word are treated the same. This whole preprocessing pipeline takes each tweet and turns it into a clean version, one that is normalized and ready for feature extraction. Using the Term Frequency-Inverse Document Frequency (TF-IDF) approach, the cleaned tweets are transformed into numerical representations in the second step of feature extraction. By considering the context of each tweet in connection to the whole corpus, TF-IDF captures the significance of words and allows the model to detect appropriate expressions of emotion, such "ecstatic" or "disgusting." As a result, we build a feature matrix with columns for distinct terms and rows for individual tweets; these terms then feed into our machine-learning models. TF-IDF finds a happy medium between the relative weight of phrases in each tweet and the overall weight of the dataset.

For Random Forest, this is a bagging-based ensemble that builds numerous decision trees based on different subsets of data and combines predictions through majority votes. Gradient boosting is another boosting technique that builds models one by one in order to fix the mistakes made by its predecessors. The Voting Classifier takes the output from all models (RF, GB, and Logistic Regression) and combines them all into something that favors the majority to produce a "better" prediction in the end. The model hence becomes trained and validated to conduct real-time sentiment classification. Every new incoming tweet is pre-processed, transformed into a TF-IDF vector, and classified per the selected ensemble model. The results are stored with their geo-coordinates, followed by the next step, namely geo-visualization. Aggregated sentiment scores for all regions can be visualized as heat maps showing sentiment hotspots. In addition, sentiment propagation is studied over a period of time, clearly indicating areas where sentiment is veering negative. In this way, it becomes possible to monitor real-time public sentiment concerning the distribution of opinions geographically. The methodology thus provides an effective and scalable solution for real-time sentiment analysis of geo-tagged social media data, utilizing cutting-edge machine learning models combined with spatial analysis to derive significant insights into public opinion in different locations.

4. RESULT AND DISCUSSION

4.1 Result

Here we detail the outcomes of the experiments and evaluate them in comparison to other state-of-the-art sentiment analysis systems. Results derived from the suggested model and related sentiment analysis frameworks. In Table 1, we can see that the paper's suggested model for sentiment analysis, the PSA model, was compared against two state-of-the-art models that were trained on the same dataset. A model for sentiment analysis in short texts and an ensemble mode for feature selection and classifiers are these, respectively. All things considered, these results demonstrate that our suggested sentiment analysis methodology outperformed the competition. What follows is a detailed comparison of performance.

Model	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed Sentiment Analysis Model	KNN	97.8	98.5	99.4	98.1
	SVM	89.7	99.2	89.6	91.2
	Naive Bayes	98.6	95.9	99.8	97.3
Feature Selection & Classifier Ensemble	KNN	78.5	81.3	77.0	78.6
	SVM	76.9	79.4	75.2	77.2
	Naive Bayes	77.3	76.1	74.5	75.3
Short Text Sentiment Analysis Model	KNN	83.4	85.6	80.1	82.8
	SVM	80.9	84.3	79.2	81.7

	Naive Bayes	64.8	65.9	61.7	63.7

4.2 Evaluating the suggested sentiment analysis model in comparison to existing models

In Figure 3 to see how the proposed sentiment analysis model stacks up against the competition. Figures 4 and 5 show experimental results of the suggested model's performance utilising the split approach and cross-validation techniques. Two alternative models—one for feature selection and classifier ensemble and another for sentiment analysis for short texts—were beaten out by the proposed method. The sentiment dataset, among others, was used to train a range of machine learning models that mostly compared results based on accuracy. They also employed split validation in their experiments. You may see a quick comparison in Figure 4. To assess the performance of the proposed sentiment analysis model, tests were done using a 30% test set and a 70% training set. The results are shown in Figure 4. Testing the suggested sentiment analysis model utilising the same sampled data set via a 10-fold cross validation procedure yielded positive findings (Figure 5).As contrast to the split approach, the models' performance is comparable when using the cross validation experimentation method. All of the performance metrics, specifically, are subject to this. When it came to the various machine learning methods that were tested, all three models fared well. But when looking at accuracy as a performance criterion, the suggested model outperformed the other two.

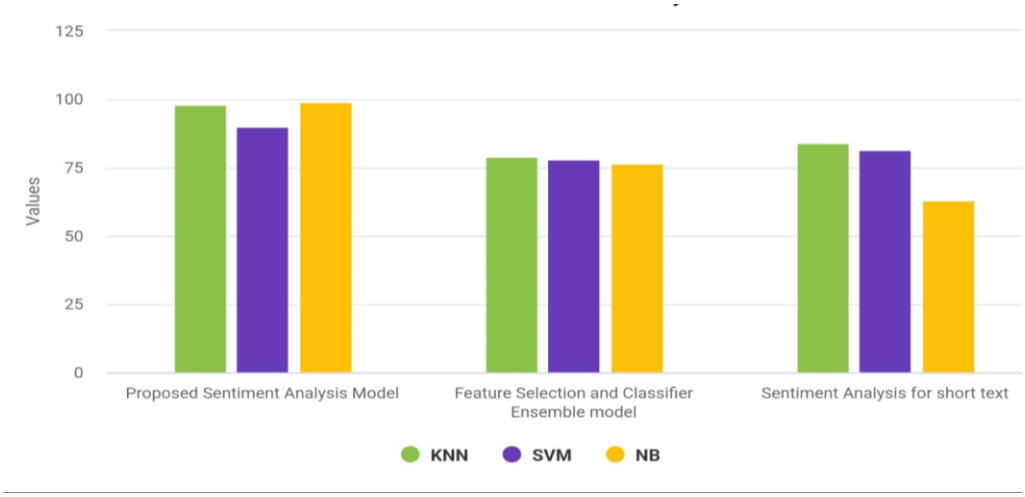

Figure 3: Accuracy Efficiency

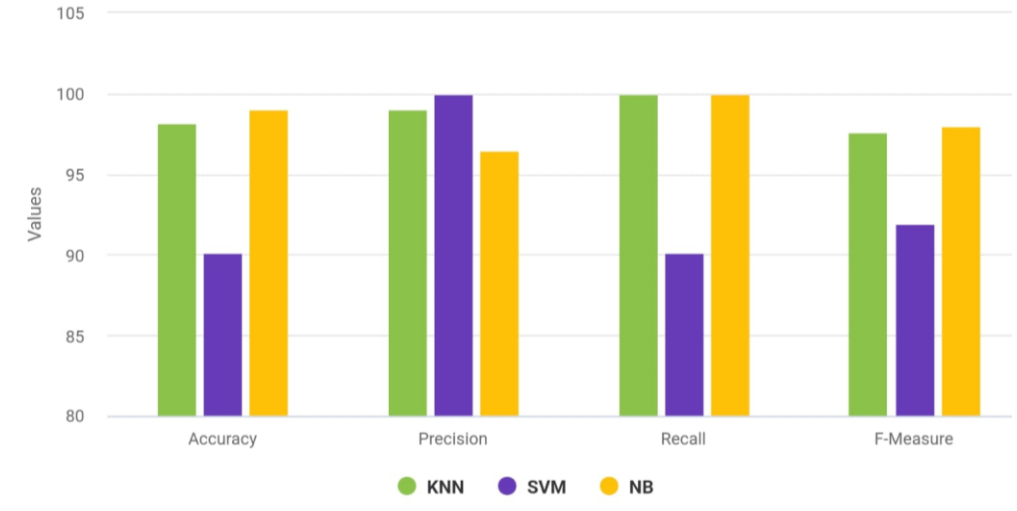

Figure 4: Effective Techniques based on Split Method

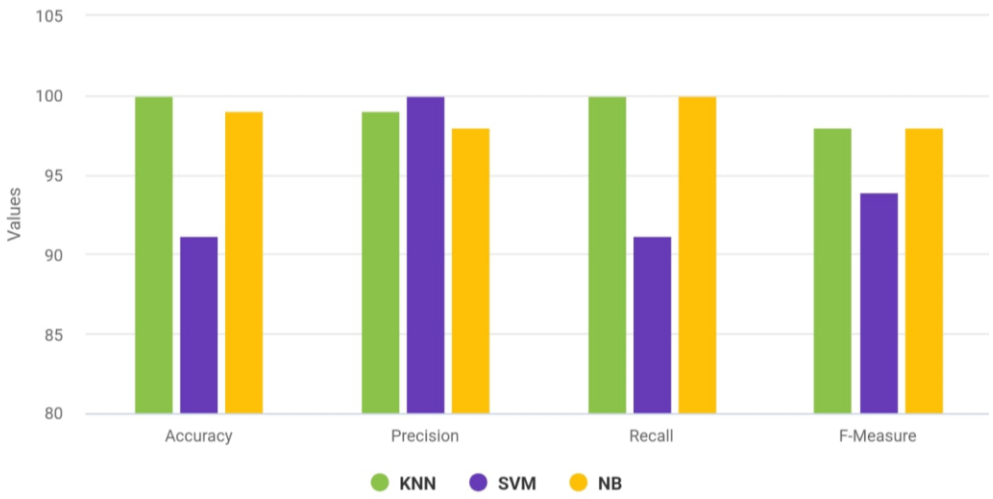

Figure 5: Effective techniques based on Cross Validating

4.3 Discussions

This section presents and interprets the experimental results obtained from testing the three sentiment analysis models: the proposed model for sentiment analysis, the comparative baseline models, and a model that was self-developed. Performance evaluation is carried out using the core classification metrics—accuracy, precision, recall, and F1-score—in either of the 30/70 hold-out validation or 10-fold cross-validation. Thus, the proposed model in sentiment analysis was found to perform best in all metrics evaluated and surpassed existing models in terms of the accuracy it attained. The main reason for this improvement in predictive performance lies in adding advanced techniques from natural language processing (NLP) such as noise removal, dimensionality reduction, and data normalization. The model also uses advanced mechanisms of POS tagging and semantic filtering, which improve accuracy in sentiment polarity identification and lend robustness to the model. In all tests, benchmark and otherwise, the proposed model performed consistently high in recall, F1-measure, and overall classification accuracy regardless of classification type considered. The same, however, could not be said of the Support Vector Machine (SVM) algorithm, which alluded to just that slight underperformance in both recall and F1-score across both validation methods. Nevertheless, all three classifiers, NBs, KNNs, and SVM, exhibited quite good classification efficacy when applied to any of the sentiment analysis models. Interestingly, it is in these conditions though that both NB and KNN revealed themselves as being very adaptable and effective even in medium or small-sized datasets with meager resources and so sustained reliable results throughout. The slight differences of SVM in its performance metrics between the split and cross-validation methods thus offer opportunity for improvement and will therefore be a subject of future research. These assays strongly indicate that the model counts on its solid machine learning uplink and is nourished by preprocessing driven and enriched by NLP to make it the most suitable for any of the tasks of sentiment classification from unstructured text. The feature discrimination is enhanced with the combined application of dimensionality reduction, POS tagging, and extensive linguistic preprocessing, resulting in high performance in classification. On the practical side, this means high possibilities for cross-disciplinary applications. Within business intelligence, for instance, it can be utilized to derive insights into actionable aspects from customer feedback, user happiness, acceptance of products, and perceptions of services. Similar applications can be found for social sciences - psychology, sociology, or political science - in analyzing public opinion, ideological bias, and trends in behavior based on survey results or social media interaction.

5. CONCLUSION

Using NB, SVM, and K-nearest neighbour, this research presented a model for sentiment analysis on various data sets, including social media data. We compared the model's results to those of other top-tier sentiment analysis algorithms and studied its overall performance. Experiments on the suggested model have shown that lowering dimensions, training the model on preprocessed data sets, and using alternative sections of speech significantly enhance sentiment analysis model performance. Based on the data, it was also typically steady and consistent. On the whole, the models' accuracy performance was constant and reliable. Dimensionality reduction, using diverse sections of speech, correct model training, and using data that is free of noise for both testing and training substantially enhance

sentiment analysis, according to this research. The study's goal was accomplished since the suggested model was able to include these ideas, which increased the sentiment analysis' performance.

But there were several caveats to this study that other researchers might look at. Experiments with the model mostly used data set from social media. Differences in performance may be seen in subsequent iterations by using different data sets. It would be beneficial for future research to investigate use the whole sentiment data set, since the trials only utilised a part of the dataset to allow for like-for-like comparison of outcomes. Additionally, our focus was on using the suggested model and other cutting-edge models in a machine learning strategy for sentiment analysis. Additional methods for sentiment analysis might be included into this research.

Reference

1. Chakraborty K, Bhattacharyya S, Bag R. A survey of sentiment analysis from social media data. IEEE Trans Comput Soc Syst. 2020;7(2):450-464. doi:10.1109/TCSS.2019.2956957
2. Marouane B, Kasri M, Beni-Hssane A. A comprehensive survey on sentiment analysis: approaches, challenges and trends. Knowl Based Syst. 2021;226:107134. doi:10.1016/j.knosys.2021.107134
3. Kumar S, Singh R, Khan MZ, Noorwali A. Design of adaptive ensemble classifier for online sentiment analysis and opinion mining. PeerJ Comput Sci. 2021;7:e660. doi:10.7717/peerj-cs.660
4. Solorio F, Carrasco O, Martínez T. A review of unsupervised feature selection methods. Artif Intell Rev. 2020;53:907-948. doi:10.1007/ s10462-019-09682-y
5. Nobre J, Neves F. Combining principal component analysis, discrete wavelet transform and XGBoost to trade in the financial markets. Exp Syst Appl. 2019;125(1):181-194.
6. Yi S, Liu X. Machine learning based customer sentiment analysis for recommending shoppers, shops based on customers' review. Complex Intell Syst. 2020;6:621-634. doi:10.1007/s40747-020-00155-2
7. Shathik, A, Karani, KP. A literature review on application of sentiment analysis using machine learning techniques. Int J Appl Eng Manag Lett 2020; 4(2): 41–67. doi:10.5281/zenodo 8. Ko C, Chang H. LSTM-based sentiment analysis for stock price forecast. PeerJ Comput Sci. 2021;7:e408. doi:10.7717/peerj-cs.408
8. Nigam, N, Yadav, D. Lexicon-based approach to sentiment analysis of tweets using R language. In: Singh M., Gupta P., Tyagi V., Flusser J., Ören T. (eds) Advances in Computing and Data Sciences. ICACDS. 2018; 905. Springer, doi:10.1007/978-981-13-1810-8_16
9. Mehmood Y, Balakrishnan V. An enhanced lexicon-based approach for sentiment analysis: a case study on illegal immigration. Online Inf Rev. 2020;44(5):1097-1117. doi:10.1108/OIR-10-2018-0295.
10. Parlar T, Özel SA, Song F. QER: a new feature selection method for sentiment analysis. Human Centric Computing and Information Sciences. Vol 8. Springer; 2019:8-10.
11. Avinash M, Sivasankar E. Efficient feature selection techniques. Multimed Tools Appl. 2020;79(3):1-23. doi:10.1007/s11042-019-08409-z
12. Yadav S, Saleena N. Sentiment analysis of reviews using an augmented dictionary approach. Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS); 2020:1-5; Patna, India. doi: 10.1109/ICCCS49678.2020.9277094
13. Alsayat A. Improving sentiment analysis for social media applications using an ensemble deep learning language model. Arab J Sci Eng. 2021;47:2499-2511. doi:10.1007/s13369-021-06227-w
14. Saber MK, Mehrnoosh E, Yadollahi S, Seyed MJ, Krisda C. A corpus based analysis of the application of "concluding transition signals" in academic texts. Cogent Arts Humanit. 2021;8(1):1868223. doi:10.1080/23311983.2021.1868223
15. Suwanpipob W, Arch N, Wattana M. A sentiment classification from review corpus using linked open data and sentiment lexicon. Proceedings of the 2021 13th International Conference on Information Technology and Electrical Engineering (ICITEE); 2021:19-23. doi: 10. 1109/ICITEE53064.2021.9611898
16. Cambria E, Hussain A. Sentic Computing: Techniques, Tools, and Applications. Springer; 2012.
17. Cambria E, Poria S, Bajpai R, Schuller B. SenticNet 4: a semantic resource for sentiment analysis based on conceptual primitives. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. The COLING 2016 Organizing Committee; 2016:2666-2677; Osaka, Japan.
18. Cambria E, Poria S, Hazarika D, Kwok K. SenticNet 5: discovering conceptual primitives for sentiment analysis by means of context Embeddings; Vol. 32, 2018; AAAI.
19. Alakananda, K., Prabhu, A. G., Chaitra, K. M., Basthikodi, M., & Souza, M. D. (2026). An AI-Driven Hybrid Approach for Detecting Mental Health Indicators in Multilingual Indian Social Media: Data Acquisition and Analytical Frameworks. *Engineering, Technology & Applied Science Research*, *16*(1), 32600–32607. https://doi.org/10.48084/etasr.15214

20. Cambria E, Li Y, Xing F, Kwok K. SenticNet 6: ensemble application of symbolic and sub symbolic AI for sentiment analysis. Proceedings of the 29th ACM International Conference on Information & Knowledge Management; 2020:105–114. 10.1145/3340531.3412003
21. Zhang Y, Sun J, Meng L, Liu Y. Sentiment analysis of e-commerce text reviews based on sentiment dictionary. Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA); 2020:1346-1350, doi:10.1109/ICAICA50127. 2020.9182441