International Journal of Computer Science Engineering Techniques – Volume 9 Issue 5, September - October - 2025

Domain-Specific Sentiment Analysis of Twitter Data: A Machine Learning Approach

1st Akshitha Katkeri
Department of Computer Science and Engineering
Assistant Professor
BNM Institute of Technology,
Affiliated to VTU
Bengaluru, India
akshithakatkeri@bnmit.in

2nd N G Shirisha
Department of Computer Science and Engineering
BNM Institute of Technology,
Affiliated to VTU
Bengaluru, India
ngshirisha01@gmail.com

3rd Manisha H S
Department of Computer Science and Engineering
BNM Institute of Technology,
Affiliated to VTU
Bengaluru, India
manishahangud@gmail.com

***Abstract*—This research investigates the application of machine learning techniques to analyze the sentiment expressed in tweets. Twitter, being a widely used platform, gives people a space to discuss a wide range of topics, including their experiences with products and services to opinions on social or political issues. Studying this type of content helps us better understand how the public feels about certain topics and what trends are emerging in conversations. For our analysis, we worked with the Sentiment140 dataset, which contains tweets already marked as either positive or negative. Before integrating this dataset into our models, we carried out several preparation steps. These included cleaning the text, removing elements like hashtags and stop-words that don't add much value, and converting the text into numbers using TF-IDF — a method that highlights important words. We then applied and compared three well-known machine learning models: Logistic Regression, Naive Bayes, and Support Vector Machine (SVM). Out of the three, SVM gave the best results in terms of accuracy. To enhance the practicality of our system, we have created a simple interface where users can input any sentence and immediately see whether the sentiment behind it is positive or negative. This feature could be useful for businesses to track public feedback, assess brand image, or support marketing decisions.**

***Index Terms*—Sentiment Analysis, Twitter, Machine Learning, Natural Language Processing (NLP), Sentiment140 Dataset, Text Classification, Social Media Analytics.**

I. Introduction

As social media has grown in popularity, it has become a place where people openly share thoughts, emotions, and feedback on nearly every subject. Among these platforms, Twitter is especially active, with users posting short messages—tweets—about daily events, opinions, and experiences. These tweets serve as a rich resource for public expression and provide valuable perspectives on people's sentiments regarding various subjects, including products, services, and social issues. Sentiment analysis focuses on identifying the emotional tone behind written text. Given the enormous volume of tweets shared each day is extremely high, manually reviewing them for sentiment is not realistic. Therefore, machine learning models are utilized to automatically determine whether a tweet conveys a positive, negative, or neutral sentiment. The effectiveness of the analysis often hinges on what extent the text is preprocessed beforehand, making preprocessing a crucial component of the overall process.

A. Problems and Opportunities

Social media platforms such as Twitter enable real-time sharing of feedback and opinions. Each day, users create content around a range of subjects, be it current affairs, news, goods, or others. All this user-generated content offers rich opportunities for businesses to use it in a variety of ways, from reputation management to tracking political attitudes or even studying mental health patterns. In addition, the users help in the creation of automated moderation tools meant to gauge the likely impact of the content. This will enable businesses to perform self-audits, improve marketing campaigns, and develop better services. Nevertheless, while the vast potential lies in wait, utilization of raw social media data is, as yet, challenging because of its native messiness, non-structure, and proclivity for colloquial usage. As the increasing demand to find core feelings and sentiments arises, sophisticated techniques in ML and NLP have become indispensable to achieve penetrating and profound insight extraction.

B. Project Objectives

Our project seeks to create a sentiment classification model utilizing machine learning methods. Objective is to categorize tweets into positive or negative sentiments. Follows:

- **Data Acquisition and Cleaning**: Utilizing the Sentiment140 dataset, which contains 1.6 million labeled tweets. The data undergoes cleaning steps like removing links, hashtags, and irrelevant symbols.
- **Feature Engineering**: Converting cleaned text into numerical vectors using the TF-IDF technique for model compatibility.
- **Model Development**: Training models such as Logistic Regression, Naive Bayes, and SVM using the prepared dataset.
- **Performance Evaluation**: Comparing the accuracy and effectiveness of each algorithm in correctly identifying sentiment.
- **Real-time Prediction Tool**: Building an interface that enables users to input text and instantly receive sentiment predictions.

II. LITERATURE REVIEW

The analysis of Twitter has become a focal point, over time, for its usefulness in capturing public opinions from brief and informal messages [1]. [2] Antonakaki and colleagues further highlighted its role in detecting harmful online behavior, such as cyberbullying, through machine learning approaches. [3] Kolchyna and team explored both lexicon-based and machine-learning techniques, concluding that blending the two leads to better classification accuracy. This insight enabled the integration of traditional and modern techniques in sentiment classification on Twitter. [4] Cliche developed a hybrid deep learning framework with CNN and LSTM layers to detect emotions in tweets. Their research demonstrated that neural networks can outperform older models by understanding subtle and complex language patterns. [5] Sahni's approach leveraged distant supervision by tagging tweets with noisy labels based on their emotional tone, enabling efficient sentiment detection. Its approach has helped reduce the need for manually marked data while maintaining the effectiveness of the classification. [6] Chen and colleagues proposed a novel model incorporating emojis and an attention-based LSTM to enhance focus on key tweet elements, boosting overall accuracy. These studies emphasize the growing sophistication of approaches to analyzing emotions and the shift from traditional rule-based methods to automatic learning and deep learning. Building on earlier techniques, this project applies classic machine learning models—like logistic regression, SVM, and naive Bayes—trained on the Sentiment140 dataset.

III. SYSTEM OVERVIEW

The overall architecture of the sentiment analysis system is structured in stages, from data handling to model deployment. Below is a breakdown of each component:

a) Dataset Description: The dataset was split into 80 percent for training and 20 percent for testing. To evaluate how well each model performed, metrics including precision, recall, F1-score, and accuracy were used, based on the classification reports generated.

b) Preprocessing Pipeline: To prepare the tweets for model input, a sequence of pre-processing steps are performed:

- **Text Cleaning**: This involves removing URLs, mentions, hashtags, and special characters from the text.
- **Normalization**: All text is converted to lowercase to achieve uniformity.
- **Tokenization**: The text is split into individual words.
- **Stopword Removal**: Common words that have little semantic value are filtered out.
- **Lemmatization**: Words are reduced to their base or dictionary form using tools like the WordNet Lemmatizer.
- **Feature Transformation**: the given text is converted into numerical vectors using methods such as TF-IDF and **TF-IDF** and optionally, Count Vectorizer.

- *Model Training:* Four classification models are developed and trained:

- **Logistic Regression**: A simple yet effective linear classifier.
- **Naive Bayes**: A probabilistic model assuming independent between-natures.
- **Support Vector Machine (SVM)**: Model that finds the optimal decision boundary in a high-dimensional space.
- **LSTM**: A Deep learning model developed with Keras is especially suitable for handling sequential text data.

c) Model Optimization:

- **Grid Search** is used with Logistic Regression to fine-tune parameters such as regularization.
- **Synthetic Minority Over-sampling Technique** is employed to balance the dataset if sentiment classes are unevenly distributed.

- *Model Evaluation:* The models are tested on unseen data using various metrics:

- **Precision**: indicates how many of the tweets predicted as positive are actually correct, helping measure the model's exactness.
- **Recall**: It assesses the ability to identify all positive cases.
- **F1-Score**: The harmonic mean of precision and recall.
- **Accuracy**: Overall percentage of correct predictions.
- **Confusion Matrix**: Summarizes the model's performance by displaying correct and incorrect predictions for all classes.

d) Visualization:

- **Word Clouds**: Generated for each sentiment class to visualize commonly occurring words.
- **Training Graphs**: Accuracy and loss graphs are plotted for the LSTM model to track the learning process across epochs.

e) Model Saving and Deployment:

- Trained models are saved using Model.save() for future use.
- Tokenizer and label encoder are also saved.
- The model can optionally be deployed via a web application (e.g., Flask) for real-time sentiment predictions.

f) Deliverables:

- Preprocessed dataset
- Trained models: Trained models: A collection of trained classification models, including a linear model, a probabilistic approach, a margin-based classifier, and a deep neural network (Logistic Regression, Naive Bayes, SVM, and LSTM, respectively).
- Evaluation reports and confusion matrices
- Visualizations: Word clouds, accuracy/loss plots
- Exported models for deployment

IV. SYSTEM IMPLEMENTATION

This research work evaluates the effectiveness of different sentiment analysis models using a Twitter dataset. It examines both classical machine learning approaches and model deep learning architectures, all of which aim to classify the tone of each tweet. '(SVM): A model embedding layer, LSTM layers, and a softmax output layer. The efficacy of each method

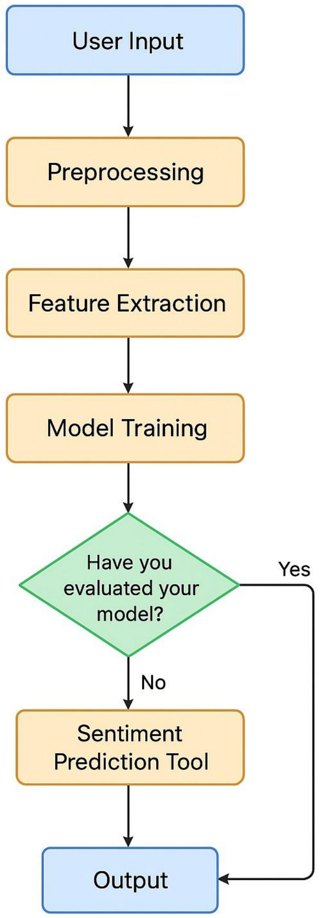

Fig 1. Flowchart Representation of Sentiment Analysis Model Development Process.

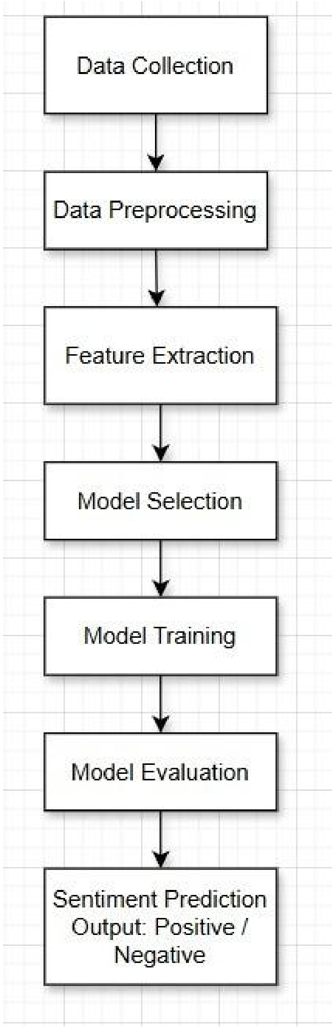

Fig 2. Stepwise Framework for Sentiment Analysis Model Development.

in determining tweets to be categorized as either positive or negative. test dataset involved is a mixture of the Sentiment140 and Twitter The training and Airline Sentiment datasets with evenly balanced positive and negative tweets.

A. Implementation Details The experiment was carried out in Python with standard libraries like scikit-learn, Tensor-Flow, and Keras. We used scikit-learn to implement Logistic Regression, Naive Bayes, and SVM, while the LSTM model was developed using the Keras framework. The dataset was split into training and testing sets in an 80% for training and 20% for testing.To evaluate the performance of all the models, we used precision, recall, F1-score, and accuracy based on the classification reports generated for each model.

B. Model Evaluation

Equations (1), (2), and (3) provide the formulas needed to calculate precision, recall, and F1-score.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Where:

- **TP** = True Positives
- **FP** = False Positives
- **FN** = False Negatives

1. **Logistic Regression:** Logistic Regression was accurate at 76 Figure 3 presents key performance metrics—precision, recall, and F1-score—for two sentiment classes: negative and positive. • Negative Sentiment: Precision of 0.77, Recall of 0.74, F1-score of 0.76 (Support: 9995 samples) • Positive Sentiment: Precision of 0.75, Recall of 0.78, F1-score of 0.76 (Support: 10005 samples) • The overall accuracy is 0.76, with a total support of 20000 samples. The model exhibited balanced effectiveness across all sentiment classes with macro

Sentiment	Precision	Recall	F1-Score	Support
Negative	0.75	0.74	0.75	9995
Positive	0.75	0.75	0.75	10005
Accuracy	--	--	0.75	20000
Macro Avg	0.75	0.75	0.75	20000
Weighted Avg	0.75	0.75	0.75	20000

Fig 3. Classification Report for Logistic Regression in Sentiment Analysis

Sentiment	Precision	Recall	F1-Score	Support
Negative	0.76	0.74	0.75	9995
Positive	0.75	0.77	0.76	10005
Accuracy			0.76	20000
Macro Avg	0.76	0.75	0.75	20000
Weighted Avg	0.76	0.76	0.75	20000

Fig 4. Classification Report for Naive Bayes Model in Sentiment Analysis

and weighted averages of 0.76 for precision, recall, and F1-score.

2. **Naive Bayes:** achieved a solid accuracy of 75 Figure 4 displays key performance metrics—precision, recall, and F1-score—for two sentiment classes: negative and positive.
• Negative Sentiment: Precision of 0.75, Recall of 0.74, F1-score of 0.75 (Support: 9995 samples) Figure 4 displays key performance metrics—precision, recall, and F1-score—for two sentiment classes: negative and positive.

• Positive Sentiment: Precision of 0.75, Recall of 0.75, F1-score of 0.75 (Support: 10005 samples) • The overall accuracy is 0.75, with a total support of 20,000 samples. A macro and weighted average of 0.75 across precision, recall, and F1-score reflects the model's balanced effectiveness across all evaluation parameters.

3. **SVM:** had an accuracy of 75 Figure 5 presents key performance metrics—precision, recall, and F1-score—for two sentiment classes: negative and positive. • Negative Sentiment: Precision of 0.76, Recall of 0.74, F1-score of 0.75 (Support: 9995 samples) • Positive Sentiment: Precision of 0.75, Recall of 0.77, F1-score of 0.76 (Support: 10005 samples) • The overall accuracy is 0.76, with a total support of 20,000 samples. The F1-score averaged around 0.75, showing that the model performed evenly across all sentiment categories.

4. **LSTM (Long Short-Term Memory):** Performed badly, with a mere 50 Classification Report for Support Vector Machine in Sentiment Analysis This classification report provides precision, recall, and f1-score metrics for the LSTM model's performance in predicting negative and positive sentiments. The model exhibits poor performance in identifying negative sentiment, with an f1-score of 0.00 for that class. C. The results indicated that traditional machine learning models, such as Logistic Regression, outperformed the LSTM model. The LSTM's weaker performance may be due to issues like

Sentiment	Precision	Recall	F1-Score	Support
Negative	0.00	0.00	0.00	9995
Positive	0.50	1.00	0.67	10005
Accuracy	--	--	0.50	20000
Macro Avg	0.25	0.50	0.33	20000
Weighted Avg	0.25	0.50	0.33	20000

Fig 5. Evaluation of the Performance of Support Vector Machines for Sentiment Analysis.

Sentiment	Precision	Recall	F1-Score	Support
Negative	0.00	0.00	0.00	9995
Positive	0.50	1.00	0.67	10005
Accuracy	--	--	0.50	20000
Macro Avg	0.25	0.50	0.33	20000
Weighted Avg	0.25	0.50	0.33	20000

Fig 6. Classification Report of LSTM Model Displaying Performance Metrics for Negative and Positive Classes.

class imbalance or insufficient training. Naive Bayes and SVM demonstrated stable performance maintaining a good balance between precision and recall. Further refinement is required by exploring different model designs and training strategies to improve its effectiveness in sentiment analysis tasks. C. The results show that the established machine learning models—logistic regression, Naive Bayes, and SVM—yielded consistent and reliable outcomes across different sentiment categories. In contrast, the LSTM model demonstrated lower performance, possibly due to class imbalance, limited training cycles, or inadequate hyperparameter optimization. To improve results, future research could investigate various deep learning configurations, increase the amount of training data, and incorporate techniques such as word embeddings and attention mechanisms.

V. CONCLUSION

This study provides a detailed comparison of classical machine learning models and deep learning architectures for sentiment analysis using Twitter datasets (Ëmotion Recognition through Text, Speech, and Image") [7]. The experiments were performed with Python using scikit-learn and Keras libraries, and the models were trained on 80 Among the models that were tested, Logistic Regression had the best accuracy of 76 The experimental findings indicate that, in the current setup and data conditions, conventional machine learning models outperform better than the Long Short-Term Memory model in sentiment analysis of Twitter data. Future work will concentrate on overcoming the limitations of the LSTM model. This will involve modifying its architecture and parameters, increasing the number of training epochs, and potentially

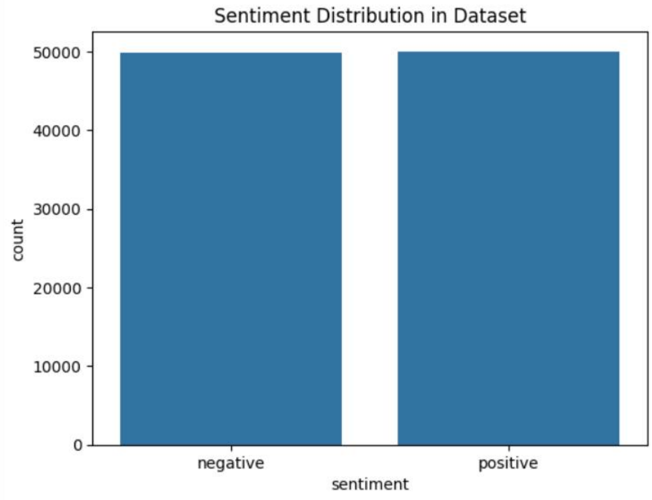

Fig 7. Sentiment Distribution in Dataset.

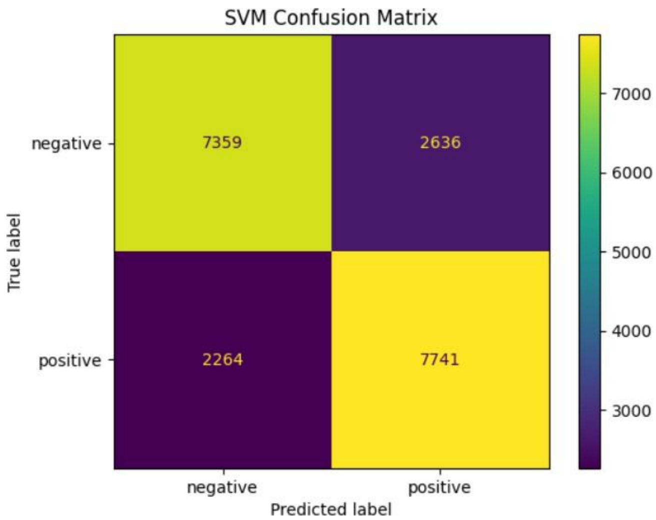

Fig 8. SVM Confusion Matrix Illustrating Model Predictions

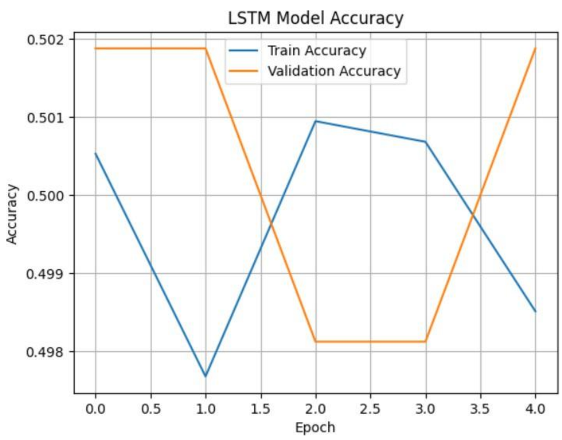

Fig 9. Training and Validation Accuracy Trends Across Epochs in LSTM Model Development.

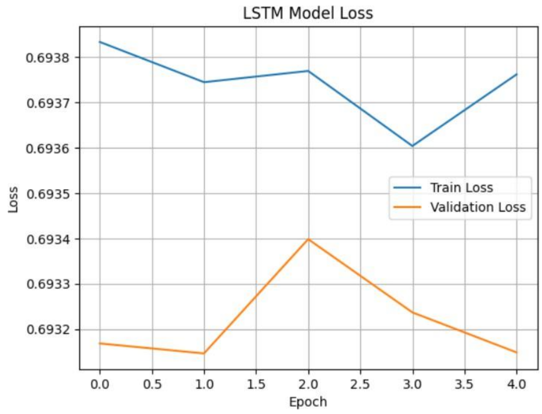

Fig 10. Training and Validation Loss Evolution Across Epochs in LSTM Model Optimization.

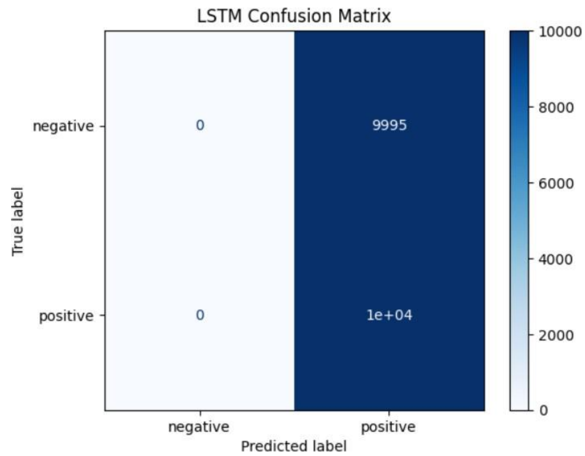

Fig 11. Confusion Matrix of LSTM Model Showing Classification Performance

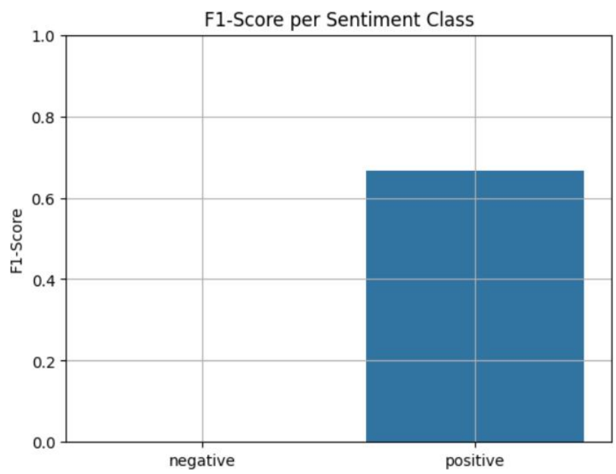

Fig 12. F1-Score Distribution Across Sentiment Classes in LSTM Model Evaluation.

incorporating techniques such as word embeddings and attention mechanisms to enhance deep learning performance.

REFERENCES

[1] N. Yadav, O. Kudale, S. Gupta, A. Rao, and A. Shitole, "Twitter sentiment analysis using machine learning for product evaluation," in *Proc. 2020 Int. Conf. Inventive Comput. Technol. (ICICT)*, Coimbatore, India, 2020, pp. 181–185.

[2] A. Antonakaki, S. Spiliotopoulos, Fragopoulou, and P. Ioannidis, "Detecting cyberbullying on Twitter: A machine learning approach," *Procedia Comput. Sci.*, vol. 194, pp. 353–362, 2021.

[3] O. Kolchyna, T. S. Treleaven, D. Luong, and T. Aste, "Twitter sentiment analysis: Lexicon method, machine method, and their combination," *arXiv preprint* arXiv:1507.00955, 2015.

[4] D. Cliche, "BB twtr at SemEval-2017 Task: Twitter sentiment analysis with CNNs and LSTMs," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval-2017)*, Vancouver, Canada, 2017, pp. 573–580.

[5] A. Sahni, A. Goyal, and N. Bali, "Efficient Twitter sentiment classification using distant supervision," in *Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Noida, India, 2017, pp. 527–531.

[6] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using Bi-sense emoji embedding and attention-based LSTM," *Knowl.-Based Syst.*, vol. 170, pp. 39–51, 2019.

[7] "Emotion recognition through text, speech and image," *Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET)*. [Online]. Available: www.ijraset.com/research-paper/emotion-recognition-through-text-speech-and-image