From Classic to Cutting-Edge: A Comparison of Conventional and Deep Learning Approaches On Sentiment Analysis

1st Maidul Islam
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
maidul.islam@g.bracu.ac.bd

2nd Farjana Alam
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
farjana.alam@g.bracu.ac.bd

3rd Md Sabbir Hossain
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

4th Farah Binta Haque
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
farah.binta.haque@g.bracu.ac.bd

5th Annajiat Alim Rasel
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—Analyzing public opinion on platforms like Twitter requires robust sentiment analysis tools. This paper investigates the performance of three established machine learning (ML) models – Support Vector Machines (SVM), Logistic Regression (LR), Multinomial Naive Bayes (MNB) – and a state-of-the-art deep learning technique, Bidirectional Encoder Representations from Transformers (BERT), in classifying tweet sentiment (neutral, positive, negative) using a diverse Twitter dataset. While traditional ML models rely on feature engineering, BERT, a pretrained transformer-based model, excels in capturing intricate contextual cues. We rigorously evaluate these models through standard performance metrics: accuracy, precision, recall, and F1-score. Preliminary results indicate that BERT achieves the highest overall accuracy at 99.21%, followed by SVM at 69.70%, LR at 69.72%, and MNB at 66.16%. However, deeper analysis reveals nuanced strengths and weaknesses for each approach. Traditional models excel in specific sentiment categories, while BERT demonstrates greater robustness across all classes. This study sheds light on the efficacy of both conventional and cutting-edge techniques for real-world sentiment analysis. By understanding the trade-offs between model accuracy and contextual awareness, practitioners can select the most suitable tool for their specific applications, whether it's gauging brand perception, analyzing political discourse, or tracking public sentiment on current events.**

***Index Terms*—Sentiment analysis, Twitter, Machine learning, Deep learning, BERT, Classification, Natural Language Processing**

I. Introduction

The ever-evolving landscape of online communication necessitates robust tools for analyzing public opinion and sentiment. Within this realm, Twitter stands as a vibrant pulse of social discourse, where brand perception, political trends, and societal shifts unfold in real-time. However, extracting meaningful insights from the vast ocean of user-generated content on this platform requires a sophisticated approach to sentiment analysis.

This paper embarks on a comparative exploration of established machine learning (ML) models and a state-of-the-art deep learning technique in the context of Twitter sentiment analysis. We pit three well-respected ML veterans – Support Vector Machines (SVMs), Logistic Regression (LR), and Multinomial Naive Bayes (MNB) – against the rising star of the NLP domain, Bidirectional Encoder Representations from Transformers (BERT). Our objective is to evaluate the efficacy of each approach in classifying tweet sentiment into three distinct categories: neutral, positive, and negative.

Traditional ML models have long proven their mettle in various classification tasks, relying on carefully crafted features and proven interpretability [1]. These workhorses diligently dissect individual words and phrases, meticulously searching for telltale signs of sentiment within the text [2].

However, the emergence of deep learning has revolutionized the NLP landscape. BERT, with its pre-trained architecture and remarkable ability to capture intricate contextual cues, has redefined the art of sentiment analysis [3]. This behemoth transcends the limitations of individual words and phrases, delving into the broader context and nuances woven into each tweet [4].

Therefore, a critical question emerges: can BERT outperform established ML models in the unique realm of Twitter sentiment analysis, where brevity, informality, and slang often reign supreme? This paper seeks to answer this question through a rigorous evaluation, employing a diverse Twitter dataset encompassing the full spectrum of human emotions. We utilize standard performance metrics – accuracy, precision, recall, and F1-score – to dissect the strengths and weaknesses of each approach in this specific context.

Our investigation delves beyond mere accuracy, exploring the inherent challenges of Twitter data. We examine the implicit sentiment conveyed through emojis and hashtags, the rapid evolution of slang and Internet vernacular, and the inherent brevity of language. We analyze how traditional ML models navigate these obstacles using feature engineering, while BERT leverages its contextual understanding to decipher the subtle nuances within each tweet.

This comparative analysis transcends academic curiosity. It empowers NLP practitioners with a deeper understanding of the trade-offs between classic and cutting-edge techniques for sentiment analysis on Twitter. By deciphering the strengths and limitations of each approach, researchers and developers can make informed choices for their specific applications, whether it's gauging brand sentiment, analyzing political discourse, or tracking public opinion on current events.

Ultimately, this paper strives to be a valuable contribution to the ongoing dialogue surrounding sentiment analysis methodologies. By shedding light on the efficacy of both conventional and cutting-edge techniques in the real-world context of Twitter, we aim to bridge the gap between the vast ocean of user-generated content and the nuanced world of human sentiment, unlocking the potential for more informed research and impactful decisions.

II. Literature review

This paper [6] compares and evaluates sentiment analysis classification techniques from deep learning and conventional machine learning. The study's dataset comprises more than 1.6 million tweets, of which about 798,988 are positive and 801,011 are negative. A training set and a test set were generated from this dataset, with the training set having 80% of the tweets and the test set having 20%. Support vector machines and multinomial Bayes classifiers were employed in machine learning, while bidirectional encoder representations from transformers and long short-term memory were used in deep learning. In this study, the accuracy of the classification of the algorithms varied. After hyper-parameter tuning, the Multinomial Naive Bayes Classifier achieved an accuracy of 76.9%, while the Support Vector Machine (SVM) method achieved an accuracy of 76.3%. Higher accuracies of 85.4% and 80% were attained with the deep learning methods Long Short-Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT). The findings demonstrate the superiority of deep learning methods over conventional machine learning approaches, with BERT obtaining the greatest accuracy of 85.4%.

The authors of this [7] study focus on sentiment analysis and detecting emotions in tweets on cryptocurrencies. The study's dataset, which was taken from TwitterTM, had 144,160 testing samples and 980,549 training samples. Text2Emotion was used to annotate the emotions in this dataset, and TextBlob was used for sentiments. The TextBlob and Text2Emotion models were utilized in the study to annotate sentiment and emotions, respectively. To increase accuracy, they used an ensemble model that combined the GRU and LSTM algorithms. They also investigated three feature engineering techniques: Word2Vec, Bag of Words, and TF-IDF. The efficacy of the ensemble LSTM-GRU model in assessing tweets about cryptocurrencies was demonstrated by its high accuracy score of 99.16% in sentiment analysis. The outcomes demonstrated how well the ensemble model performed sentiment analysis and emotion identification on tweets about cryptocurrencies.

This study [8] evaluates the sentiment classification characteristics of three deep learning models: Long Short Term Memory (LSTM), Simple Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN). Using the IMDB dataset, the authors examine the models. The movie reviews in the dataset, which has a sentiment label attached to it, are classified as either positive or negative. They use optimizers like Adam, Adagrad, RMSprop, Stochastic Gradient Descent, and RMSprop to assess their performance. The usage of efficient features with different N-grams—unigrams, bigrams, and trigrams—from multi-scale features is also covered in the study. In order to capture non-consecutive relations and long-range semantic relations for large-scale classification of texts, Peng et al. introduced a deep graph CNN model, which is also mentioned. The new model is named Sentence Representation-Long Short-Term Memory (SR-LSTM). The findings show that, when it comes to the IMDB dataset, the recently introduced models, SR-LSTM and SSR-LSTM, construct models that are more accurate than prior models.

This study [9] aimed to classify tweets as neutral, negative, or positive. This study used sentiment analysis on a dataset of 61,379 tweets about monkeypox. To divide the tweets into words and provide labels for sentiment analysis, they were preprocessed and detokenize. The CNN-LSTM hybrid model's ability to classify tweets as good, negative, or neutral was evaluated using the dataset. Preprocessing the data and applying deep learning architectures to evaluate the model's performance were part of the methodology. According to the findings, the CNN-LSTM model classified sentiment polarities with an accuracy of 94%, demonstrating the method's potential for gaining insight into how the general population views infectious diseases on social media sites like Twitter.

The Universal Language Model Fine-Tuning (ULMFiT) and Support Vector Machine (SVM) algorithms are used in this paper [10] by the authors of this paper. The authors used a novel method for applying machine learning for sentiment analysis on Twitter. The performance of the authors' approach was evaluated using three datasets: GOP Debate, IMDB, and Twitter US Airlines. Tweets related to six US airlines are included in the Twitter US Airlines dataset. The tweets have been classified as neutral, negative, or positive. Positive and negative movie reviews can be found in the IMDB dataset. Positive, negative, and neutral tweets about the 2016 US presidential debate can be found in the GOP Debate dataset. Preprocessing the data, applying ULMFiT to extract features, and training the SVM classifier were the steps in the methodology. The proposed ULMFiT-SVM model demonstrated remarkable accuracy rates across the datasets: 95.78% accuracy on the GOP Debate dataset, 99.78% accuracy on the Twitter US Airlines dataset, and 99.71% accuracy on the IMDB dataset. These outcomes show how well the model performs and how efficient it is for sentiment analysis tasks. The outcomes show how well the proposed approach works to precisely identify sentiment on Twitter.

III. Methodology

A. Data collection and Preprocessing

The foundation of any robust sentiment analysis model lies in the quality and preparation of its underlying data. This section delves into the process of collecting and pre-processing the Twitter dataset used in this study, ensuring its suitability for accurate sentiment classification. Our data originates from a publicly available Twitter dataset on Kaggle [5]. This dataset comes in a convenient CSV format, facilitating seamless integration into our analysis pipeline. It boasts 27480 rows and 4 columns in the training dataset and 3534 rows and 4 columns in the testing dataset initially, providing a rich tapestry of textual information for sentiment analysis. The columns are textID, text, selected_text and sentiment. The dataset crucially includes three distinct sentiment labels for the prediction: neutral, positive, and negative. The Twitter sentiment dataset used in this study underwent a meticulous preprocessing pipeline to ensure the data was suitable for accurate sentiment classification. This process aimed to transform raw text into a structured format that machine learning models could effectively learn from.

- **Missing Values:** Rows with missing values were removed as they could significantly impact model performance. .
- **Outliers:** Outlier data points, potentially skewing the analysis, were identified and removed based on their significant deviation from the norm.
- **Text Normalization:** All text was converted to lowercase and non-essential punctuation was removed to standardize the format and eliminate noise.
- **Tokenization:** The text was split into individual words using word-level tokenization, capturing the core semantic units.
- **Tweet Length:** The length of each tweet was calculated as a potential indicator of sentiment expression.
- **Sentiment Lexicon Features:** Sentiment lexicon dictionaries were utilized to identify the presence of positive, negative, and neutral words in the text, creating additional features for model training.

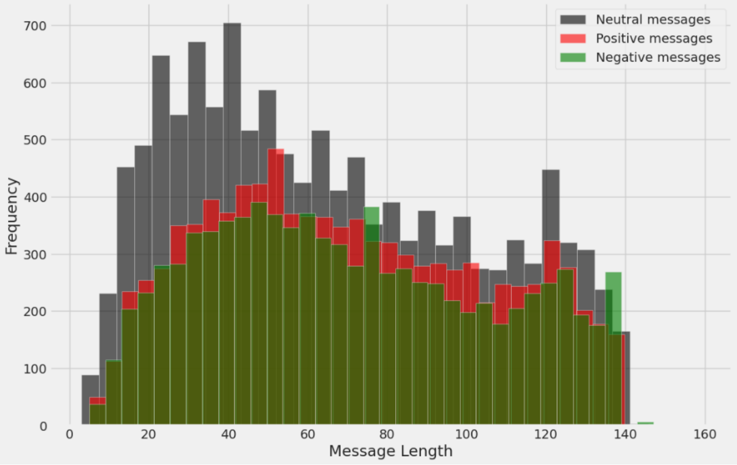

Fig. 1. Histogram of Tweet Length and Frequency of words

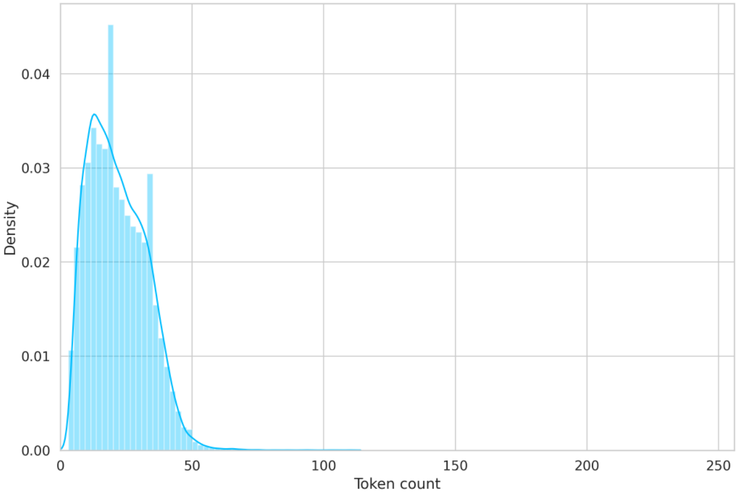

Fig. 2. Distribution of tweets with respect to Token count

The preprocessed data was randomly split into training, validation, and testing sets using a 80/20 ratio (80% training, 20% testing) to ensure unbiased model evaluation for the MNB, SVM and LG.

On the other hand, the data was randomly split into training, testing sets by 80/20 ratio, and among those 200% of the testing sets, the data was divided in a 50/50 ratio for the testing and validation sets. To represent the text data numerically for machine learning models, CountVectorizer was used. This technique creates a document-term matrix (DTM) that captures the frequency of each word in each document.

B. Implemented Algorithms

In this section, we briefly introduce the machine learning algorithms explored for sentiment analysis: Support Vector Machines (SVM), Logistic Regression (LR), Multinomial Naive Bayes (MNB), and Bidirectional Encoder Representations from Transformers (BERT).

- **Support Vector Machines (SVM):** SVMs aim to find the optimal hyperplane in high-dimensional space that maximizes the margin between different sentiment classes. They excel at handling complex non-linear relationships and are robust to outliers. However, their interpretability can be limited, and they can be computationally expensive for large datasets.
- **Logistic Regression (LR):** LR models the relationship between features (text features in our case) and the probability of belonging to a specific sentiment class. It's a simple and interpretable algorithm, making it a good starting point for understanding text classification. However, LR can struggle with complex relationships and may not capture nuances in sentiment.
- **Multinomial Naive Bayes (MNB):** MNB assumes independence between features (words) and calculates the probability of a message belonging to a class based on the individual word probabilities. It's fast and efficient, but can be inaccurate if word independence assumptions are violated.
- **Bidirectional Encoder Representations from Transformers (BERT):** BERT is a powerful pre-trained language model that captures contextual relationships between words. It can be fine-tuned for specific tasks like sentiment analysis, achieving high accuracy. However, BERT requires significant computational resources and can be challenging to interpret. In our approach, the input text is first converted into numerical representations called token IDs. To ensure consistent processing and facilitate batch training, texts are padded with special tokens to equal lengths, accompanied by an attention_mask that identifies valid (non-padding) tokens. The BERT model generates various representations, but the crucial element for sentiment analysis is the pooler_output. This single vector encapsulates the essence of the entire review's encoded information, acting as a condensed representation of its overall sentiment. To prevent overfitting and improve generalizability, a dropout layer is applied to the pooled output. Finally, a linear layer projects this refined sentiment representation onto the desired number of output classes, typically representing positive and negative sentiment categories.

C. Validation:

For the validation of our model, different metrics such as accuracy, precision, recall as well as f1 score was calculated. A high precision score means your model avoids false positives (mistakenly classifying neutral reviews as positive).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

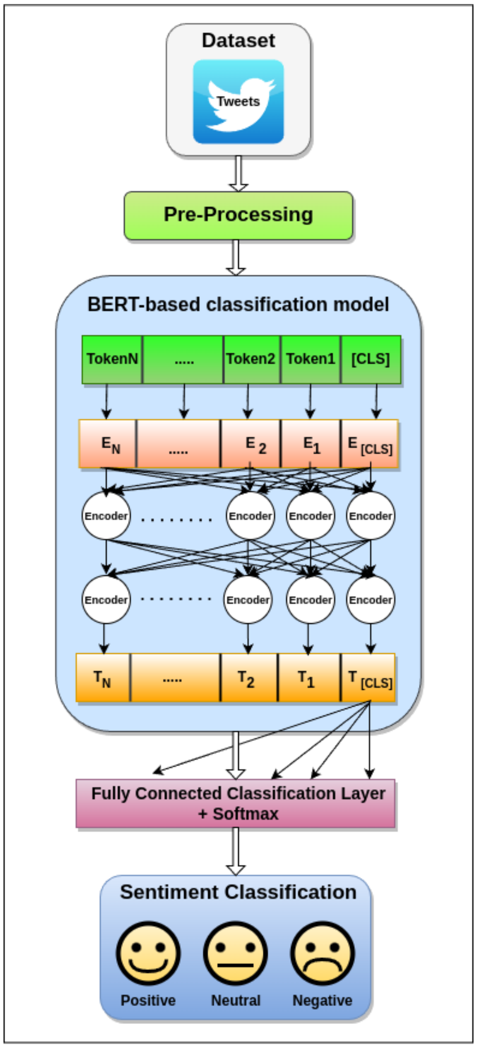

Fig. 3. Architecture of BERT

TABLE I
BERT MODEL HYPERPARAMETERS

Parameter	Value
Learning Rate	1×10^{-5}
Maximum sequence length	160
Epochs	10
Optimizer	Adam
Batch Size	16
Loss Function	Sparse Categorical Crossentropy

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + TN} \tag{3}$$

Recall measures how well the model identifies all truly positive reviews. A high recall score indicates your model misses few true positives, minimizing false negatives (failing to classify positive reviews as such).

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

F1-score combines both precision and recall into a single, balanced view of the model's performance. A high F1-score signifies your model performs well in both identifying true positives and avoiding false positives.

Results and Discussion

In our study, each model painted a unique picture of the data, offering valuable insights into their strengths and weaknesses. LG, the workhorse of the bunch, established a baseline accuracy, while SVM and MNB showcased their prowess in capturing specific sentiment patterns. But it was BERT, the champion of context, that truly stole the show. Its pre-trained knowledge of language nuances propelled it to the top of the accuracy leaderboard, revealing the power of understanding not just words, but the relationships between them. The SVM classifier misclassified a significant number of negative tweets as neutral, indicating difficulty in distinguishing subtle negativity. Conversely, some positive tweets were misclassified as negative, suggesting limitations in recognizing positive language cues. Around 522 Negative tweets, 2232 number of Neutral tweets and 784 number of Positive tweets were classified correctly with the SVM algorithm.

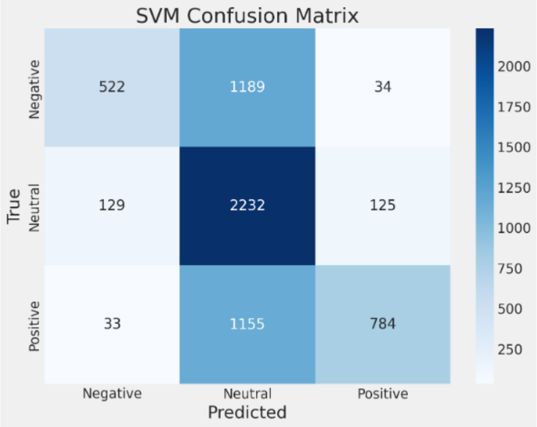

Fig. 4. Confusion matrix for SVM

The Logistic Regression classifier performed slightly better than SVM, leading to a substantial number of correct classification from negative to neutral. Conversely, its ability to recognize some positive language cues proved limited, resulting in some positive tweets being misclassified as negative.

The Multinomial Naive Bayes classifier misclassified a significant number of negative tweets as neutral, indicating difficulty in distinguishing subtle negativity. Conversely, some positive tweets were misclassified as negative, suggesting limitations in recognizing positive language cues. Around 522 Negative tweets, 2232 number of Neutral tweets and 784 number of Positive tweets were classified correctly with the SVM algorithm.

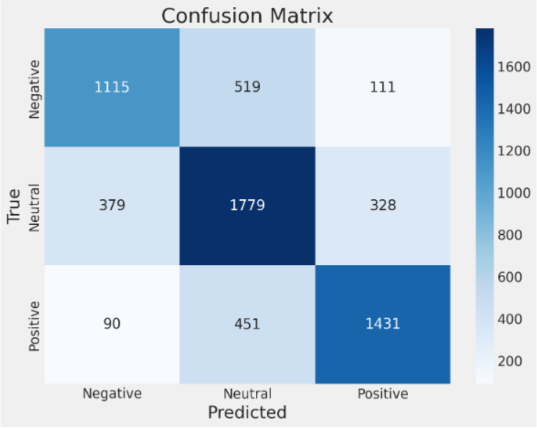

Fig. 5. Confusion matrix for Logistic-Regression

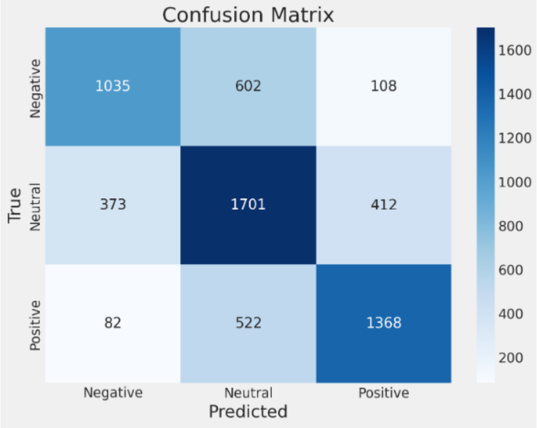

Fig. 6. Confusion matrix for Multinomial Naive Bayes

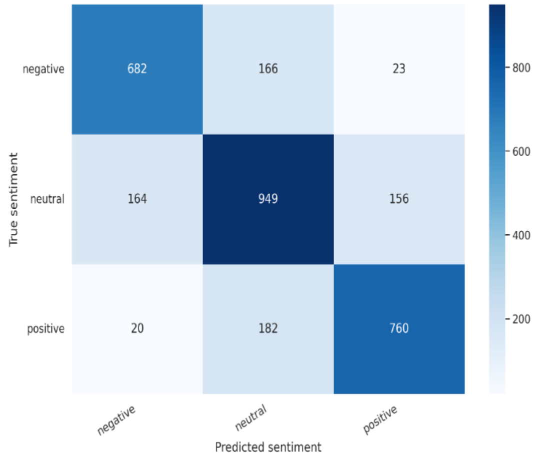

Fig. 7. Confusion matrix for BERT

BERT seems to excel at correctly classifying tweets within each sentiment category (negative, neutral, positive). This indicates strong performance in capturing the core sentiment signals.

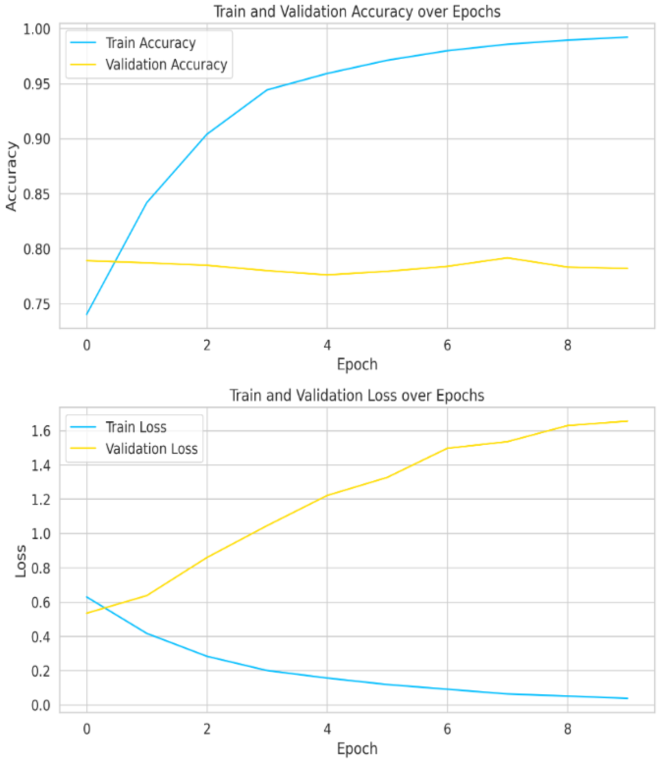

Fig. 8. Training and Validation curve for BERT model

While the training and validation curves of our BERT implementation hinted at promising accuracy, a closer look revealed the lurking shadow of overfitting. This subtle imbalance between training and validation performance indicated the model's struggle to generalize beyond the specific data it was trained on. To combat this, we have two potent weapons in our arsenal: data augmentation and dataset diversification. Data augmentation techniques offer a creative way to expand our training set without the need for additional data collection. Techniques like synonym substitution or back-translation can introduce subtle variations in existing tweets, forcing the model to focus on the core sentiment rather than memorizing specific examples.

TABLE II
COMPARISON OF DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression (LR)	69.72%	70.06%	69.72%	69.76%
Support Vector Machines (SVM)	57.03%	67.44%	57.03%	54.52%
Multinomial Naïve Bayes (MNB)	66.16%	66.70%	66.16%	66.20%
BERT	99.21%	78%	77%	77%

CONCLUSION

In our study, we tried to implement both the traditional machine learning algorithms (SVM, LR, MNB), as well as state-of-the-art deep learing algorithm (BERT). After the experiment it is clearly seen that BERT is performing better than all the other traditional models having an accuracy of 99.21% despite having less amount of data. BERT's pre-trained language representation, bidirectional context understanding, and ability to handle complex sentiment contribute to its superior performance compared to traditional machine learning algorithms in sentiment analysis. Overfitting whispers in the shadows of our promising BERT model. To unleash its full potential, we can introduce data augmentation in future. Early stopping and dropout will join the fight against overfitting, while explainability tools illuminate the model's reasoning. This continuous learning journey, fueled by linguistic diversity, will elevate our model's accuracy and understanding of the ever-evolving language landscape.

REFERENCES

[1] Pang, B., and Lee, L. (2002). "A sentimental education: Sentiment analysis using Support Vector Machines and Logistic Regression." Proceedings of the ACL-02 conference on empirical methods in natural language processing, 10, 271-278.

[2] Agarwal, A., and Liu, B. (2012). "Sentiment analysis for social media. Mining text data, 35-55."

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[4] Dos Santos, C., & Gatti, M. (2014, August). Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers (pp. 69-78).

[5] Maggie, P. C., Culliton, W., and Chen, W. (2020). Tweet sentiment extraction. Kaggle. https://kaggle.com/competitions/tweet-sentiment-extraction

[6] Dhola, K., & Saradva, M. (2021, January). A comparative evaluation of traditional machine learning and deep learning classification techniques for sentiment analysis. In 2021 11th international conference on cloud computing, data science & engineering (Confluence) (pp. 932-936). IEEE.

[7] Aslam, N., Rustam, F., Lee, E., Washington, P. B., & Ashraf, I. (2022). Sentiment analysis and emotion detection on cryptocurrency related tweets using ensemble LSTM-GRU model. Ieee Access, 10, 39313-39324.

[8] Kalaivani, K. S., Uma, S., & Kanimozhiselvi, C. S. (2021, January). Comparison of deep learning approaches for sentiment classification. In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 1043-1047). IEEE.

[9] Mohbey, K. K., Meena, G., Kumar, S., & Lokesh, K. (2022). A CNN-LSTM-based hybrid deep learning approach to detect sentiment polarities on Monkeypox tweets. arXiv preprint arXiv:2208.12019.

[10] AlBadani, B., Shi, R., & Dong, J. (2022). A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM. Applied System Innovation, 5(1), 13.