Sentiment Analysis of Twitter Data Using Machine Learning and Deep Learning

Md.Saiful Islam and Mahede Hasan

October 25, 2024

Sentiment Analysis of Twitter Data Using Machine Learning and Deep Learning

***Abstract*— In recent years, sentiment analysis has gained significant attention due to the surge in social media and e-commerce platforms. It involves analyzing people's opinions to determine the polarity, beneficial for assessing customer reviews and identifying social trends. Our thesis focuses on a dataset comprising over 29,530 tweets, aiming to discern whether they contain hateful content. Employing machine learning techniques such as Naïve Bayes, Support Vector Machine, Logistic Regression, and Random Forest, we conducted a classification task, evaluating performance through precision, recall, f1-score, and accuracy. Despite minor variations (1-2%) among the models, Random Forest yielded the highest accuracy at 96.24%. The study didn't conclude there; we extended our exploration to deep learning, specifically employing Bidirectional-Long Short-Term Memory. Surprisingly, the deep learning model's accuracy slightly lagged behind machine learning. Consequently, our final determination is that, for our dataset, machine learning outperforms deep learning. In the course of our research, we delved into the challenges and limitations, providing a comprehensive analysis of our work.**

Keywords— Models CNN, Detection, Naïve Bayes, Support Vector Machine, Logistic Regression, Random Forest.

I. INTRODUCTION

In the current era of the twenty-first century, characterized by technological advancement and information abundance, extensive data accessibility empowers informed decision-making. A significant portion of this data comprises opinions shared by millions on social media and blogging platforms like Facebook, Twitter, Snapchat, and Instagram. The rapid growth of these platforms facilitates instant expression of thoughts, beliefs, and emotions. Sentiment analysis, an automated process, categorizes textual content into positive, negative, or neutral sentiments. This analysis is crucial for businesses and customer services, allowing efficient processing of vast amounts of feedback to adapt services and monitor brand reputation. In the era of machine learning and deep learning, sentiment analysis has become more efficient, enabling the quick analysis of large datasets. The paper explores sentiment analysis using various machine learning and deep learning algorithms. The thesis focuses on determining tweet sentiment polarity using machine learning models (Random Forest, Naïve Bayes, Support Vector Machine, Logistic Regression) and deep learning models (Bidirectional-Long Short-Term Memory, Recurrent Neural Network with Bi-LSTM) [1]. Notably, the Random Forest algorithm excels in sentiment analysis, and the research compares the performance of machine learning and deep learning models on a Twitter dataset.

II. RELATED WORKS

The focus was on evaluating the efficacy of various supervised techniques in detecting spam reviews. In 2021, Zeeshan et al. evaluated supervised machine-learning algorithms for spam email detection. They used Naïve Bayes, Support Vector Machine, and Random Forest, achieving high accuracies: 98.8% for Multinomial Naïve Bayes, 97.6% for Bernoulli Naïve Bayes, 91.5% for Gaussian Naïve Bayes, 97.8% for Random Forest, and 98.5% for SVM[2].

In (2022), Naeem Ahmed and Rashid Amin presented a total of five supervised learning methods were scrutinized for their effectiveness in this task, namely Naïve Bayes, SVM, K-NN, Logistic Regression, and Decision Tree. The findings revealed that SVM exhibited superior performance with an accuracy of 83.19%, surpassing the other strategies. On the contrary, Decision Tree exhibited a notably lower accuracy of 51.00%[3].

In 2022, Arifuzzaman conducted Bangla Text Sentiment Analysis using various models. VDCNN achieved the highest accuracy (77.85%). RNN surpassed RCNN in precision (78.88%). Bi-LSTM excelled on a cricket dataset (78.14%). GRU models showed lower performance than LSTM and Bi-LSTM. BERT-LSTM achieved superior accuracy (84.18%) and precision (86.45%)[4].

Richa Dhanta et al. (2023) analyzed Twitter sentiment by classifying tweets into favorable, unfavorable, or neutral categories. They preprocessed data and used logistic regression and Naive Bayes, finding Naive Bayes to be the most effective based on F1 score, accuracy, recall, and precision[5].

III. PROBLEM STATEMENT

In today's digital age, online social media allows millions to share their opinions, necessitating sentiment analysis for various purposes, such as improving business services or detecting harmful statements. This project aims to create an accurate sentiment analysis model using both machine learning and deep learning approaches. The project involves: Selecting a dataset of thousands of tweets.Preprocessing and filtering the dataset.Analyzing the dataset using supervised machine learning algorithms.Applying RNN with Bi-LSTM and comparing the results with machine learning algorithms.Identifying the best model based on the outcomes.

IV. INTRODUCTION TO SENTIMENT ANALYSIS

Sentiment Analysis (Opinion Mining) uses natural language processing (NLP) to assess the emotional tone of text, categorizing it as positive, negative, or neutral.

It leverages data mining, machine learning (ML), and artificial intelligence (AI) to extract subjective information and gauge public sentiment. Techniques like tokenization and part-of-speech tagging are essential, while models such as Bag of Words, Lexicon, and various classifiers (e.g., SVM, Naive Bayes) are used for analysis. Deep learning models like LSTM and CNN are also applied[6]. Key challenges include handling temporal shifts, context-dependent meanings, and creating comprehensive lexicons.

V. RESEARCH METHODOLOGY

Twitter, one of the world's most popular social media platforms, provides data for our sentiment analysis to determine whether tweets are hateful or not. This section of the thesis outlines the implementation of the selected methods. The dataset, obtained from Kaggle, is labeled for use in supervised learning models. The dataset structure includes: "id": The ID of the Tweet account,"label": Indicates sentiment, where 0 means non-hateful and 1 means hateful,"tweet": The tweet text[7]. In this Paper, we analyze the sentiment of tweets using both traditional machine learning algorithms—such as Naive Bayes, Logistic Regression, Linear Support Vector Machines (SVM), and Random Forest—and deep neural networks, specifically Bidirectional Long Short-Term Memory (Bi-LSTM).

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation
5	6	0	[2/2] huge fan fare and big talking before the...
6	7	0	@user camping tomorrow @user @user @user @use...
7	8	0	the next school year is the year for exams.ðŸ˜...
8	9	0	we won!!! love the land!!! #allin #cavs #champ...
9	10	0	@user @user welcome here ! i'm it's so #gr...
10	11	0	â† #ireland consumer price index (mom) climb...
11	12	0	we are so selfish. #orlando #standwithorlando ...
12	13	0	i get to see my daddy today!! #80days #getti...
13	14	1	@user #cnn calls #michigan middle school 'buil...
14	15	1	no comment! in #australia #opkillingbay #se...
15	16	0	ouch...junior is angryðŸ˜#got7 #junior #yugyo...

Fig. 1. Small sample screenshot of dataset of the Twitter Data[1]

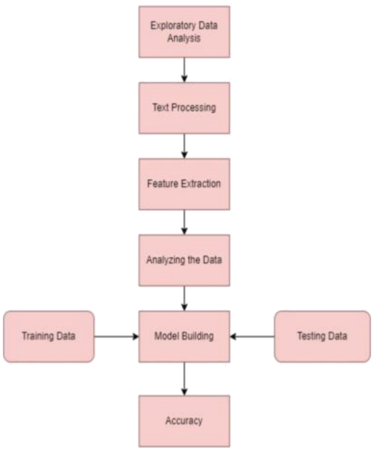


Fig. 2. Work Process

A. Data Pre-Processing:

Our sentiment analysis dataset consists of two parts: a training set and a testing set. After processing, we combined both datasets to analyze tweet sentiments. The graphs below illustrate the distribution of tweets: the blue graph represents the training set with approximately 5,800 tweets, while the brown graph represents the testing set with about 3,200 tweets. We also analyzed the distribution of hatred versus non-hatred tweets. The graph shows that non-hatred tweets (label 0) number around 30,000, while hatred tweets (label 1) number about 3,000[8].

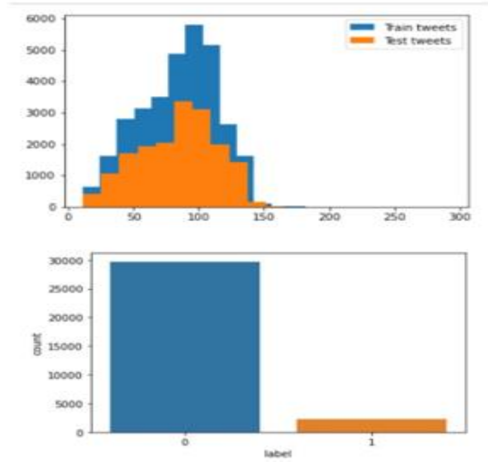


Fig.3. Distribution of train, test, hatred and non-hatred

We analyzed the distribution of tweet lengths for both hatred and non-hatred tweets. The graphs below illustrate this distribution: Hatred Tweets: The x-axis represents tweet length (in words), and the y-axis shows the number of tweets for class 1 (hatred).Non-Hatred Tweets: Similarly, the x-axis represents tweet length for class 0 (non-hatred), with the y-axis showing the number of tweets[8]. Additionally, we used a word cloud to identify the most common words in the tweets. This helped in distinguishing between positive and negative words, though some words appeared nonsensical due to a lack of preprocessing.

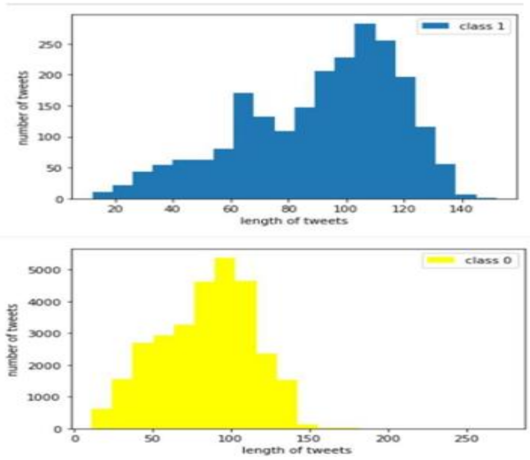


Fig. 4. Distribution of length of the hatred and non-hatred tweet

Most used words: These are the most common words found across all tweets, which are irrelevant to our sentiment analysis. To focus on the sentiment, we differentiate and identify the positive and negative words specifically. We then separately compute the frequencies of these positive and negative words for a more precise sentiment analysis[9].

Fig.5. World Cloud view for most used word in all tweets [10]

This view shows all the neutral words that are irrelevant to sentiment analysis and do not indicate positive or negative sentiment. After identifying these neutral words, we then focus on extracting the positive and negative words needed for sentiment analysis. The positive words used in tweets signify non-hatred tweets, as positive words typically indicate positive sentiments. Thus, all positive words help define non-hatred tweets.

Fig. 6. World Cloud view for positive and negative words all tweets

This section analyzes the negative words used in hatred tweets, which help identify a tweet's negative sentiment.

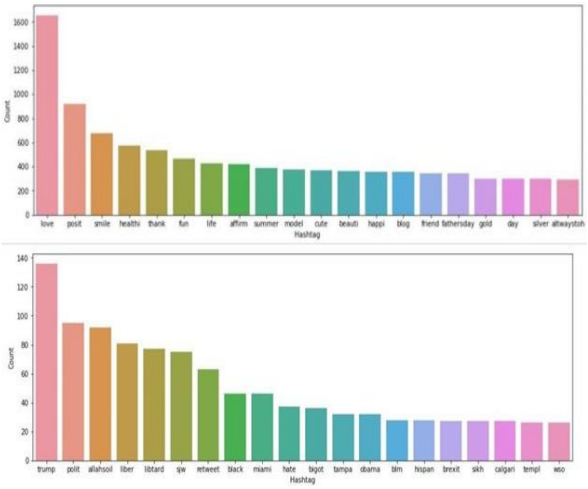


Fig.7. Quantity of top 20 positive and negative words

Negative words are inherently associated with negative contexts, making them indicative of hatred tweets. To better understand the prevalence of these words, we extracted the top 20 positive and negative words. The results are illustrated in the following figures.

B. *Text Processing:*

Clean Text: To prepare the dataset, we cleaned the text by removing usernames, hashtags, patterns, short words, articles, and irrelevant symbols. We also removed punctuations, numbers, and special characters. Tokenization split text into words, stemming reduced words to their root forms, and lemmatization converted words to their base forms using NLTK. Stop-words were eliminated to focus on meaningful content[11].

C. *Analyzing the Data:*

Splitting the Dataset: The data was divided into: Training Data: 80% for training the model. Test Data: 20% for testing the model.

D. *Model Building Algorithm:*

Input: Labeled data Output: Load Data, Exploratory Data Analysis, Data Preprocessing, Tokenization, Stemming, Lemmatization using NLTK, Feature Extraction, Split Dataset into Train and Test Sets, Combine Datasets, Implement Models, Create Confusion Matrix, Measure Accuracy. Accuracy Measures: Train Accuracy, Test Accuracy, Precision, Recall, F1-Score.

E. *Metrics and Evaluation:*

Measuring the percentage of properly anticipated events to all observations, this metric is the most basic way to assess performance[12].The accuracy of the model is calculated using the categorization data collected during each test phase, and it is stated as follows: Accuracy (%) = (nc)×100%.

• Precision: The positive predicted value is another name for precision. It is the percentage of truly positive predicted positives: Precision $= \frac{TP}{TP+FP}$

• Recall: The percentage of real positives that are anticipated to be positive is known as recall: Recall $= \frac{TP}{TP+FN}$

• F1 Score: The F1 score, the harmonic mean of precision and recall, provides a comprehensive measure of classification accuracy. It ranges from 0 to 1 and, along with the ROC curve, helps evaluate the effectiveness of classification models: F1 Score $= \frac{(R+P)*2}{(R+P)}$

• Accuracy: Accuracy is the number of correctly classified instances (true positives and true negatives):

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

VI. Applying Machine Learning Algorithm

Following these preprocessing and feature selection stages, a variety of machine learning models were examined, leading to the identification of seven models to be functionally compared. These models were selected based on study findings from multiple writers as well as factors like popularity, usability, and back-end functionality. The many classifiers employed in the study are as follows:

1. Support Vector Machines (SVM): Powerful for text classification due to their ability to find a hyperplane in n-dimensional space, effectively classifying data points. Linear SVMs are often applied to text classification problems with many features. The decision boundary is defined by the equation: f(x) = wT + b, where w is the weight vector, X is the data dataset, and b is the linear coefficient.

2. Naive Bayes: Naive Bayes classifier is a classification method rooted in Bayes' Theorem, assuming that features are independent of each other within each class. It operates on the principle that the presence of one feature does not influence the presence of another within the same class.

$$P(A|B) = \frac{P(A)\times P(B|A)}{P(B)}$$

3. Random Forest Classifier: Leverages the collective decision-making of numerous decision trees, with the majority vote determining the model's prediction. This approach ensures robustness, scalability, and resistance to overfitting. While fast and easy to interpret, its real-time prediction capability may diminish with a higher number of trees.

4. Logistic Regression: Logistic regression is a classification model employing a logistic function to represent a binary outcome. In mathematical terms, it utilizes the logistic function to compress the result of a linear equation, constraining it to a range between 0 and 1: $P(x)$= 1/1+e−(β°+β1x).

VII. Applying DEEP Learning Algorithm

Neural networks, or artificial neural networks (ANNs), are a core part of deep learning, inspired by the human brain. They consist of an input layer, multiple hidden layers, and an output layer. Each node in these layers processes input data, activating if the output surpasses a set threshold. Types of Neural Networks: ANNs: Mimic brain neuron networks for learning and decision-making. CNN: Focus on image and pattern reconditioners: Utilize feedback loops for predicting future events based on previous data. (LSTM) Structure an RNN variant with gates to manage data flow[13]. Long Short-Term Memory (LSTM): Networks are a type of Recurrent Neural Network (RNN) designed to handle long-term dependencies and mitigate the vanishing gradient problem. They use a complex architecture involving multiple gates to control the flow of information. Here is a summary of the key equations involved in LSTM cells:

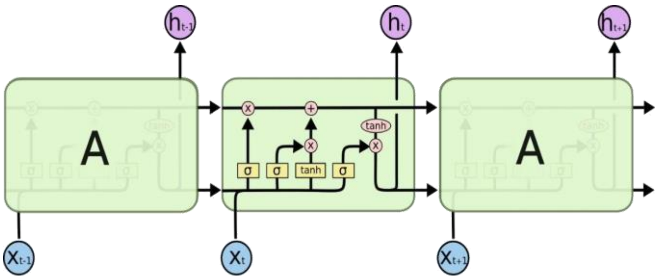


Fig.8. The repeating module in an LSTM contains four interacting layers.

1.Forget Gate: ft=σ(Wf·[ht−1,xt]+bf) Decides which information to discard from the cell state.

2. Input Gate: it=σ(Wi·[ht−1,xt]+bi) Decides which values to update in the cell state.

Candidate Cell State: C ~t=tanh(WC·[ht−1,xt]+bC) Proposes new values to add to the cell state.

3. Update Cell State: Ct=ft⊙Ct−1+it⊙C~t Combines the old cell state with the new candidate values.

4. Output Gate: ot=σ(Wo·[ht−1,xt]+bo) Determines what the next hidden state should be.

Hidden State: ht=ot⊙tanh(Ct) Outputs the hidden state for the current time step.

VIII. RESULT AND ANALYSIS

In this analysis, we examined sentiment analysis on Twitter data to classify tweets as either hateful or non-hateful. Our dataset comprises 29,530 tweets, each labeled as either 1 (hateful) or 0 (non-hateful). We tested four machine learning algorithms—Support Vector Machine, Naïve Bayes Classifier, Random Forest, and Logistic Regression—alongside one deep learning algorithm, Bidirectional Long Short-Term Memory (Bi-LSTM). We evaluated the performance of these algorithms using a variety of metrics, including confusion matrices, precision, recall, F1-score, and overall accuracy. All algorithms performed well, and their results were assessed through our evaluation system to determine the best classification outcomes.

A. *Applying Machine Learning Algorithms:*

TABLE 1. Experimental Result of Machine Learning Algorithms

Algorithm	Train Accuracy (%)	Test Accuracy (%)	Precision	Recall	F1-Score
Support Vector Machine	82.36	94.86	0.95	0.89	0.91
Naïve Bayes Classifier	80.25	94.08	0.92	0.87	0.89
Random Forest	97.74	96.24	0.98	0.90	0.93
Logistic Regression	77.32	95.01	0.96	0.88	0.92

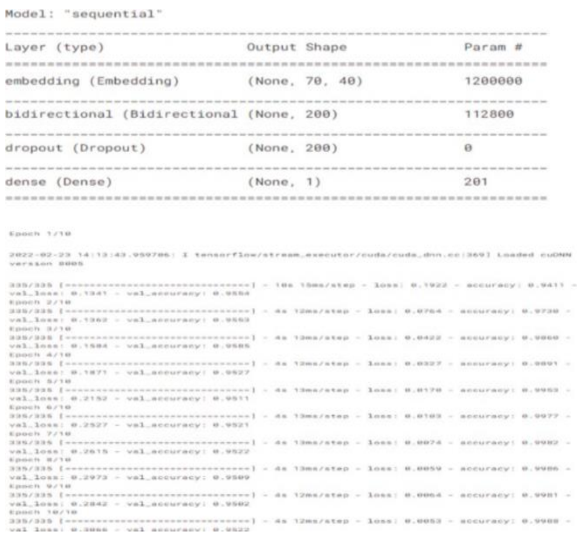
From the experimental results, it's evident that the Random Forest model delivers the highest accuracy for both training and testing datasets, along with the best F1-score[14]. This model outperforms others in our experiment, demonstrating its superior performance across accuracy, precision, recall, and F1-score metrics.Here's a summary of the performance of each model:Random Forest: Achieved the best results with an accuracy of 96.24% and an F1-score of 0.93. It excels due to its bagging technique, which enhances accuracy and manages missing data effectively, making it the top performer.Support Vector Machine (SVM): Recorded a training accuracy of 82.86%, a testing accuracy of 94.86%, and an F1-score of 0.91. SVM performs well by creating a hyperplane to maximize the margin between classes, contributing to its strong results.Logistic Regression: Delivered a training accuracy of 77% and a testing accuracy of 95.01%, with an F1-score of 0.92. It performs well in binary classification tasks, which aligns with the binary nature of our dataset.Naïve Bayes Classifier: Had the lowest accuracy in our experiment, with a training accuracy of 80.25%, a testing accuracy of 94.08%, and an F1-score of 0.89. This model's performance is limited by its reliance on word probabilities, which sometimes leads to inaccuracies. Overall, while all models showed strong results, Random Forest stands out as the most effective for our dataset, demonstrating exceptional performance and accuracy.

B. Applying Deep Learning Algorithms:

we applied the Bidirectional Long Short-Term Memory (Bi-LSTM) model from the deep learning category[15]. This model achieved an impressive accuracy of 95.22%, demonstrating excellent performance in our analysis.

TABLE 2. Experimental Result of Deep Learning Algorithm

Algorithm	Accuracy (%)	Precisio n	Recal l	F1-Score
Bi-LSTM	95.22	0.83	0.79	0.81



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 70, 40)            1200000
_________________________________________________________________
bidirectional (Bidirectional (None, 200)               112800
_________________________________________________________________
dropout (Dropout)            (None, 200)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 201
=================================================================
Epoch 1/10
```

The embedding layer produces a 2D vector for each word in the input sequence[16]. This output is then converted into binary vectors using one-hot encoding, where each integer is represented as a binary vector with a single 1 at the integer's index and 0s elsewhere. The model is then trained for 10 epochs.

C. Applying confusion matrix:

In our experiment, we utilized the confusion matrix to gain deeper insights into the performance of all the models. The confusion matrix helps us understand how each model predicts values correctly or incorrectly, providing a clearer picture of model performance. It reveals the number of true positives, true negatives, false positives, and false negatives, which allows us to identify errors and assess the accuracy of predictions. We applied the confusion matrix to our classifier models—Support Vector Machine, Naïve Bayes Classifier, Random Forest, and Logistic Regression—to evaluate their effectiveness and detect any discrepancies in their predictions[17].

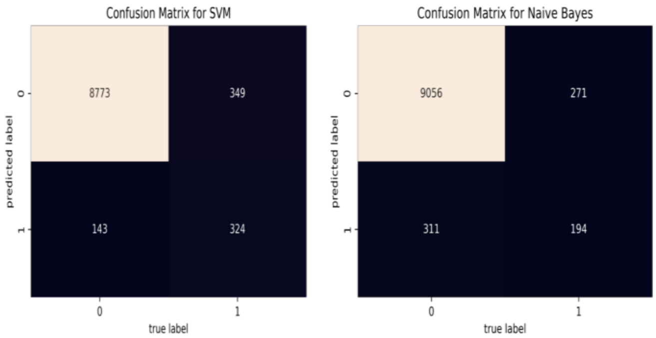


Fig. 9. Confusion Matrix of SVM,Naïve Bayes

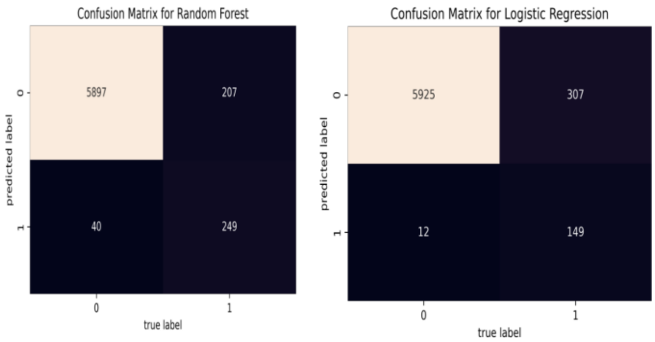


Fig.10. Confusion Matrix of Random Forest, Logistic

D. Comparison Between Machine Learning and Deep Learning:

In our experiment, we utilized both machine learning and deep learning models for sentiment analysis. We tested several machine learning algorithms, achieving good accuracy rates. Surprisingly, when we applied a deep learning algorithm, the machine learning models still provided the best accuracy[18]. By using both approaches, we aimed to offer a comprehensive view of sentiment analysis.

TABLE 3. Comparison between the accuracy rates of ML & DL

Model	Applied Algorithms	Accuracy Rate (%)
Machine Learning	Support Vector Machine	94.86
	Naïve Bayes Classifier	94.08
	Random Forest	96.24
	Logistic Regression	95.01
Deep Learning	Bi-LSTM	95.22

In our experiment, we observed only a 1% accuracy difference between machine learning and deep learning models, with machine learning models performing slightly better. This can be attributed to several factors:

Machine learning models can be effectively trained with smaller datasets, like our 29,530 tweets, whereas deep learning models typically require larger datasets.Deep learning models, although generally more accurate, did not perform better in our case, possibly due to the use of 10 epochs, which might have led to overfitting[19].Deep learning models require complex architectures and high-performance hardware, whereas machine learning models. can achieve good results with simpler setups.

Machine learning models handle missing data well by breaking tasks into smaller components, contributing to their accuracy[20].Despite these observations, both approaches provided high accuracy, indicating that each has its strengths and applicability depending on the context.

XIV. CONCLUSION

In our study, we have underscored the importance of sentiment analysis for businesses, particularly through the analysis of Twitter data to detect and prevent the spread of hateful messages. We compared various machine learning algorithms, such as Naïve Bayes, Support Vector Machine, and Logistic Regression, with a deep learning model using RNN with Bi-LSTM. Our results showed that the Random Forest algorithm outperformed the others with an accuracy of 97.74% on the training set and 96.24% on the testing set. While deep learning typically excels with larger datasets, our dataset of 16,000 tweets was relatively small, resulting in lower accuracy for the Bi-LSTM model compared to the machine learning models. Despite this, both machine learning and deep learning approaches provided commendable accuracy.

For future work, several enhancements can be explored, Multilingual Sentiment Analysis: Expanding the model to handle sentiments in various languages through machine translation, albeit with a potential performance trade-off. Unlabeled Data: Incorporating techniques to handle unlabeled data, as real-world datasets often lack labels. Audio Sentiment Analysis: Adding the capability to derive sentiments from audio, converting it to text for further analysis. Hybrid Approaches: Integrating machine learning with lexicon-based methods to create a hybrid model that leverages the strengths of both approaches, potentially improving accuracy. In conclusion, while machine learning models showed superior performance with smaller datasets, future work with larger datasets and additional features could enhance the efficacy and accuracy of sentiment analysis models.

X. DATA AVAILABILITY

We obtained the dataset from Kaggle that was used for us training purposes.

XI. DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] M. R. Hasan, M. Maliha and M. Arifuzzaman, "Sentiment Analysis with NLP on Twitter Data," 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), 2019, pp. 1-4, doi: 10.1109/IC4ME247184.2019.9036670.

[2] Naseeb Grewal,Rahul Nijhawan & Ankush Mittal Email Spam Detection Using Machine Learning and Feature Optimization Method,pp 435–447 Conference paper, First Online: 12 September 2022.

[3] Naeem Ahmed and Rashid Amin Machine Learning Algorithms for Depression: Diagnosis, Insights, and Research Directions,March 2022,DOI:10.3390/electronics11071.

[4] Bhowmik, Nitish Ranjan, et al. "Sentiment Analysis on Bangla Text Using Extended Lexicon Dictionary and Deep Learning Algorithms." Array, Elsevier, 1 Jan. 2022, /science/article/pii/S259000562100059X.

[5] Richa Dhanta, R., Sharma & H., Kumar,Twitter sentimental analysis using machine learning. International Journal of Communication and Information Technology,4(1),71–83.(2023).DOI: 10.33545/2707661x.2023.v4.i1a.63.

[6] T. Toma, S. Hassan and M. Arifuzzaman, "An Analysis of Supervised Machine Learning Algorithms for Spam Email Detection," 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI) pp. 1-5,2021 doi: 10.1109/ACMI53878.2021.9528108.

[7] L. Povoda, R. Burget and M. K. Dutta, "Sentiment analysis based on Support Vector Machine and Big Data," 2016 39th International Conference on Telecommunications and Signal Processing (TSP), pp. 543-545, 2016, doi: 10.1109/TSP.2016.7760939.

[8] M. M. Nasr, E. M. Shaaban, and A. M. Hafez, "Building sentiment analysis model using graphlab," International Journal of Scientific and Engineering Research, vol. 8, p.1155–1160, 2017.

[9] M. Moh, A. Gajjala, S. C. R. Gangireddy and T. -S. Moh, "On Multi-tier Sentiment Analysis Using Supervised Machine Learning," 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), pp. 341-344, 2015, doi: 10.1109/WI-IAT.2015.154.

[10] S.A. Bahraini an and A. Dengel, "Sentiment analysis using sentiment features," in 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 3, p. 26–29, IEEE, 2013.

[11] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentimental reviews using machine learning techniques," Procedia Computer Science, vol. 57, p. 821–829, 2015.

[12] Machine Learning Algorithm - Javatpoint", www.javatpoint.com, [Online]. Available: https://www.javatpoint.com/machine-learning-random- forest-algorithm, 2022.

[13] Deep Learning Algorithm - Javatpoint", www.javatpoint.com, [Online]. Available: https://www.javatpoint.com/deep-learning-lstm algorithm, 2022.

[14] "Kaggle: Your Machine Learning and Data Science Community", Kaggle.com, 2022. [Online]. Available: http://kaggle.com/. [Accessed: 26- Apr- 2022].

[15] S. -A. Bahrainian and A. Dengel, "Sentiment Analysis Using Sentiment Features," 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), pp. 26-29, 2013, doi: 10.1109/WI-IAT.2013.145.

[16] A. S. Rathor, A. Agarwal, and P. Dimri, "Comparative study of machine learning approaches for amazon reviews," Procedia computer science, vol. 132, p. 1552–1561, 201.

[17] M. A. Badresiya and J. Teraiya, "Performance analysis of supervised techniques for review spam detection," Int J Adv Netw Appl Special, , no. 21-24, 2014.

[18] N. Shrestha and F. Nasoz, "Deep learning sentiment analysis of amazon.com reviews and ratings," 2019, arXiv preprint arXiv:1904.04096.

[19] Huang, R. Chen, X. Zheng and Z. Dong, "Deep Sentiment Representation Based on CNN and LSTM," 2017 International Conference on Green Informatics (ICGI), pp. 30-33, 2017, doi: 10.1109/ICGI.2017.45.

[20] X. Fang and J. Zhan, ""Sentiment analysis using product review data," Journal of Big Data, Vols. 2, no. 1, p. 1–14, 2015.

[21] A. J. Shamal, R. G. H. Pemathilake, S. P. Karunathilake and G. U. Ganegoda, "Sentiment Analysis using Token2Vec and LSTMs : User Review Analyzing Module," 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 48-53, 2018, doi: 10.1109/ICTER.2018.8615600.

[22] Choudhari, and Rajankar, Introduction to Natural Language Processing With Python, International Journal of Modern Trends in Engineering and Research (IJMTER)Volume 02, Issue 04, [April – 2015].