International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Twitter Sentimental Analysis Using Recurrent Neural Network

Mrs. Usha G. R.[a], Mr. Veeresh V Kanakalamath[b] , Mr. Swaroop A[c], Mr. Shashwath C[d], Mr. Sudeepa[e]

[a] Assistant Professor, Department of Information Science and Engineering, Sri Dharmasthala Manjunatheshwara Institute of Technology, Ujire-574240
[b,c,d,e] Undergraduate Student, Department of Information Science and Engineering, Sri Dharmasthala Manjunatheshwara Institute of Technology, Ujire-574240

A B S T R A C T

One of the largest social media platforms, Twitter, has a large user base that works on a variety of projects. Twitter is the ideal platform for implementing machine learning algorithms since it generates and collects enormous volumes of data every day, making it possible to analyze information in various tweets. A technique known as a Recurrent Neural Network (RNN) is used to tackle problems requiring sequential data, such as texts or time series. The manual analysis of large amounts of textual data takes more time and effort. Artificial Intelligence (AI) is used in the automated process of sentiment analysis to identify favorable and unfavorable views in text. Analysis of sentiments is routinely applied without incorporating the insights from social media comments, survey results, and product reviews to develop data-driven conclusions. Sentiment analysis tools are used to improve unstructured content by automating business processes and cutting down on endless hours of human processing. In this article, we survey and contrast various methods for opinion mining, including machine learning methods. Using various machine learning methods like Naive Bayes, we provide study on twitter data streams. Additionally, we covered the general difficulties and uses of sentiment analysis on Twitter.

Keywords: Twitter Sentiment Analysis, Recurrent Neural Networks (RNN), Artificial Intelligence (AI), Machine Learning Algorithms, Sequential Data, Naive Bayes, LSTM

1. Introduction

An increasing number of users have utilized social media in recent years to express their opinions or write either favourable or negative reviews on a particular business. Sentiment analysis is necessary in social media monitoring due to the impact of social media content on governing, development, diplomacy, and business. It does this by allowing an overview of the general public's opinion on topics that appear in a variety of posts, from posts about politics to customer reviews. Understanding the mood and attitude behind a post on any topic helps with strategy and planning, which leads to improved services. Sentiment analysis is a computer evaluation of a country's attitudes, sentiments, and emotions regarding things like people, cases, news, structures, or other topics. The enthusiasm around sentiment analysis is due to the importance of understanding open ideas, thoughts, and questions stated at this moment. There has been a lot of research done to increase the accuracy of sentiment analysis, ranging from straightforward linear methods to more intricate deep neural network models.

An artificial neural network that can simulate sequences is called a recurrent neural network (RNN). Sequence-to-value problems can be used to frame the sentiment analysis conundrum. Text, which is made up of tokens, is the input. They consist of words, letters, and other symbols. The result might be a single number that represents the sentiment that was stated. Memory cells that can learn relationships between more widely spaced data points in a sequence provide the foundation of modern RNNs. To enhance RNN's capacity to identify the proper relationships between input elements, an attention method was developed.

Sentiment analysis is the process of identifying the emotional tone of a piece of text. It is often used to gauge public opinion on a particular topic, product, or service. Sentiment analysis can be used for a variety of purposes, including market research, social media monitoring, public relations, and crisis management.

There are several different methods for conducting sentiment analysis. One common approach is to use a lexical dictionary. A lexical dictionary is a list of words that have been assigned a sentiment score. For example, the word "love" might be assigned a positive sentiment score, while the word "hate" might be assigned a negative sentiment score. Sentiment analysis can also be conducted using machine learning. Machine learning algorithms can be trained to identify the sentiment of text by analysing large amounts of labelled data.

Sentiment analysis is a powerful tool that can be used to gain insights into public opinion. However, it is important to note that sentiment analysis is not always accurate. The accuracy of sentiment analysis depends on several factors, including the quality of the data used to train the model, the complexity of the model, and the specific application.

Benefits of using sentiment analysis:

- ***Improved decision-making***: Sentiment analysis can help businesses make better decisions by providing them with insights into customer sentiment. This information can be used to improve product development, customer service, and marketing campaigns.
- ***Increased brand awareness***: Sentiment analysis can help businesses increase brand awareness by identifying positive and negative mentions of their brand on social media. This information can be used to develop targeted marketing campaigns that reach potential customers.
- ***Improved customer service***: Sentiment analysis can help businesses improve customer service by identifying customer complaints and resolving them quickly and efficiently. This can help to improve customer satisfaction and loyalty.
- ***Reduced risk:*** Sentiment analysis can help businesses reduce risk by identifying potential problems before they occur. This information can be used to develop contingency plans and mitigate the impact of problems.
- ***Identifying trends***: Sentiment analysis can be used to identify trends in public opinion. This information can be used to develop new products and services, or to improve existing ones.
- ***Understanding customer needs:*** Sentiment analysis can be used to understand customer needs and wants. This information can be used to improve customer service and satisfaction.

Sentiment analysis is a rapidly evolving field, and new methods are being developed all the time. As sentiment analysis becomes more sophisticated, it is likely to play an even greater role in our lives.

Twitter is a widely used social media platform that generates a massive amount of data every day. Sentiment analysis of Twitter data can provide valuable insights into public opinion and sentiment towards various topics and events. The objective of this project is to develop a recurrent neural network (RNN) model for sentiment analysis of Twitter data and compare this model with machine learning techniques and show which model provides most accuracy.

Given a tweet, the objective is to predict whether the sentiment of the tweet is positive, negative, or neutral. This is a multi-class classification problem, where the input is a sequence of words in the tweet, and the output is a class label indicating the sentiment.

The challenge with this problem is that Twitter data is noisy and unstructured, with informal language, slang, and misspellings. Furthermore, the sentiment of a tweet can be influenced by context and sarcasm, making it difficult to accurately predict the sentiment.

To address these challenges, we will use RNN model and machine learning techniques. we also compare these models and manifest which of these techniques provide better results.

2. Related Work

G. Song et al. [1] proposed a sentiment-aware contextual model named SentiBERT-BiLSTM-CNN for disaster detection using Tweets. The proposed learning pipeline consists of SentiBERT that can generate sentimental contextual embeddings from a Tweet, a Bidirectional long short-term memory (BiLSTM) layer with attention, and a 1D convolutional layer for local feature extraction. Results showed that the proposed SentiBERT-BiLSTM-CNN demonstrates superior performance in the F1 score, making it a competitive model in Tweets-based disaster prediction.

Niklas Braig et al. [2] In the light of COVID-19, there was a growing demand for governments and health organizations to analyze the public's sentiment on social media. By applying a lens from social and behavioral science research, they explored how sentiment analysis can provide relevant information for managing pandemic. They used Naïve Bayes classifier, Deep Learning classifier and Natural Language Processing methods for analyzing the tweets. Their findings show that sentiment analysis in the context of COVID-19 is mostly domain and application specific. This means the classification techniques performing well on one dataset must not necessarily achieve satisfying results on a different dataset. In terms of performance, they found out that ensemble models that comprise various ML classifiers commonly outperform a single classifier model. Besides, DL classifiers show a high accuracy given the availability of sufficiently labeled data.

Meylan Wongkar et al. [3] study conducted a sentiment analysis application for Twitter analysis on 2019 Republic of Indonesia presidential candidates using the Python programming language. The results showed that the positive sentiment polarity of the Jokowi-Ma'ruf Amin pair was 45.45% and a negative value of 54.55%, while the Prabowo-Sandiaga pair received a positive sentiment score of 44.32% and negative 55.68%. The combined data was tested from the training data and got an accuracy of 80.90% , 80.1%. A comparison was carried out using the nave bayes, SVM and K-Nearest Neighbor (K-NN) methods which were tested using RapidMiner.

Fatima, Es-sabery et al. [4] studied the diversity in social media platforms leads to massive usage by the personals, and they deem these platforms as an efficient tool of communication. Therefore, the feedback of users on these platforms has generated big sentiment analysis data to learn. NLP, Hadoop framework, and deep learning models provide a set of tools that aim to capture and detect the expressed users' sentiments in the collected massive datasets from social media platforms. Authors results In their work, a novel parallel fuzzy deep learning classifier is developed, This classifier incorporates NLP text-pre-processing methods, NLP word embedding approaches, CNN+FNN deep learning model, and Mamdani fuzzy system. This proposal's primary goal is to determine a significant relationship between word embedding approaches and used deep learning models (CNN+FNN).

Sharon Susan Jacob, R. Vijayakumar et al. [5] investigated the sentiment of Twitter postings using a clustering technique based on a machine learning (ML) algorithm. Their experiments show the usefulness of our work to determine if a tweet is positive or negative utilising a massive data set of tweets from a lakh of results in an accredited testing environment.

Marco Pota ,Mirko Ventura et al. [6] attempts to present a novel, two-step methodology for Twitter sentiment analysis. First, the tweet jargon including emojis and emoticons is converted into plain text using techniques that can be easily translated into other languages or are language independent. Second, the generated tweets are then identified using the language model BERT, which was trained on plain text rather than tweets for two reasons. Pre-trained models on plain text are readily available in many languages, saving time and resources by avoiding the need to create new models from scratch and allowing for better performance. Additionally, available plain text corpora are larger than tweet-only ones, allowing for even better performance. The application of the methodology to Italian is described in a case study, along with a comparison to other existing Italian alternatives. The outcomes demonstrate the method's efficacy and suggest that, due to its methodological generality, it may also hold promise for use with other languages.

Jim Samuel et al. [7] identified public sentiment associated with the pandemic using Coronavirus specific Tweets and R statistical software. It provided insights into the progress of fear-sentiment over time as COVID-19 approached peak levels in the US. Two machine learning (ML) classification methods were used such as Naïve Bayes and logistic regression Tweet Classification methods classify Coronavirus Tweets of varying lengths, with 91% accuracy for short Tweets and 74% accuracy for longer Tweets. The research provides insights into Coronavirus fear sentiment progression and outlines associated methods, implications, limitations, and opportunities.

Asdrúbal López-Chau et al. [8] analyzed data sets generated by trending topics on Twitter from Mexican citizens during the earthquake of September 19, 2017. Three classifiers were used to predict emotions, with Naive Bayes and support vector machine being the best. The most frequent predicted emotions were happiness, anger, and sadness, with 6.5% of predicted tweets being irrelevant. We provide some recommendations about the use of machine learning techniques in sentiment analysis. The author’s contribution was the expansion of the emotions range, from three (negative, neutral, positive) to six in order to provide more elements to understand how users interact with social media platforms.

Mohammad and Bravo-Marquez et al. [9] examined the task of detecting intensity of emotion from text. They created the first datasets of tweets annotated for anger, fear, joy, and sadness intensities. They used a technique called best–worst scaling (BWS) that improves annotation consistency and obtains reliable fine-grained scores. They showed that emotion-word hashtags often impact emotion intensity, usually conveying a more intense emotion. Finally, they created a benchmark regression system and conducted experiments to determine which features were useful for detecting emotion intensity; and the extent to which two emotions are similar in terms of how they manifest in language.

Zainuddin et al. [10] developed a new hybrid technique for a Twitter aspect-based sentiment analysis. This study was about association rule mining (ARM) in part-of-speech (POS) patterns that was supplemented with a heuristics combination for recognizing explicit single and multi-word features. A rule-based method with feature selection was also included in the system for recognizing sentiment phrases.

Gogna et al. [11] proposed a hybrid approach for sentiment analysis of Twitter data, which combines deep learning and machine learning techniques. The authors employed a Convolutional Neural Network (CNN) for feature extraction and a Support Vector Machine (SVM) for classification. The proposed approach achieved an accuracy of 77.3% for sentiment classification.

Azzahrah, et al. [12] conducted a sentiment analysis of Twitter data related to the Indian General Election 2019 using machine learning algorithms such as Naïve Bayes, Logistic Regression, Support Vector Machine, and Random Forest. The authors employed pre-processing techniques such as stop-word removal, stemming, and feature selection to improve the accuracy of sentiment classification.

All authors are required to complete the Procedia exclusive license transfer agreement before the article can be published, which they can do online. This transfer agreement enables Elsevier to protect the copyrighted material for the authors but does not relinquish the authors’ proprietary rights. The copyright transfer covers the exclusive rights to reproduce and distribute the article, including reprints, photographic reproductions, microfilm or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder permission to reproduce any figures for which copyright exists.

Table 1: Comparative analysis

Sl. No	Author(s)	Algorithms/Techniques	Limitations
1.	G. Song and D. Huang [1]	BERT LSTM Convolutional Neural Network	Limited to disaster prediction and real-time analysis on Twitter data.
2.	Niklas Braig, Alina Benz, Soeren Voth, Johannes Breitenbach and Ricardo Buettner [2]	Naïve Bayes model, Deep learning classifier Natural Language Processing	Limited to sentiment analysis of COVID-19-related Twitter data.
3.	Meylan Wongkar and Apriandy Angdresey,[3]	Naïve Bayes model, support vector machine K-Nearest Neighbour	Limited to sentiment analysis of Twitter data using the Naive Bayes algorithm.

4.	Fatima Es-sabery, Abdellatif Hair, Junaid Qadir [4]	Natural Language Processing Hadoop framework deep learning models	Limited to sentence-level classification using a fuzzy deep learning classifier.
5.	Sharon Susan Jacob,R. Vijayakumar[5]	Natural Language Processing Naïve Bayes model	Limited to sentimental analysis over Twitter data using a clustering-based machine learning algorithm.
6.	Marco Pota ,Mirko Ventura[6]	Natural Language Processing models BERT	Limited to Italian language sentiment analysis using BERT-based pipeline.
7.	J. Samuel, G. G. Ali, Md Rahman, Ek Esawi and Yana Samuel[7]	Naive Bayes logistic regression Tweet Classification	Limited to COVID-19 public sentiment insights and tweets classification.
8.	Asdrúbal López-Chau, David Valle-Cruz and Rodrigo Sandoval-Almazán,[8]	Naive Bayes support vector machine Natural Language Processing	Limited to sentiment analysis of Twitter data through machine learning techniques.
9.	Mohammad,Bravo-Marquez[9]	Russell's circumplex model, SVM regression model	Limited to emotion intensities in tweets.
10.	Zainuddin et al. [10]	Naïve Bayes model	Limited to aspect-based sentiment analysis of Twitter data using hybrid sentiment classification.
11.	Gogna et al. [11]	Convolutional Neural Network support vector machine	Limited to deep learning approaches for multilingual sentiment analysis
12.	Azzahrah, et al. [12]	Naïve Bayes, Logistic Regression Support Vector Machine Random Forest	Limited to sentiment analysis of Twitter data for the Indonesian presidential election of 2019.

3. System description

Figure 1 displays the dataflow for the sentiment analysis of tweets. Twitter's detailed tweets are downloaded and placed into a dataset. Unwanted words (articles) and symbols are deleted from the retrieved tweets. The preprocessed data is examined and categorized as good, negative, or neutral using sentimental analysis. This is the system architecture and flow diagram of our proposed system here the flow of our goes on loading of csv file, pre-processing, feature extraction, classification using algorithm and evaluation of outcome using accuracy, f1-measure.

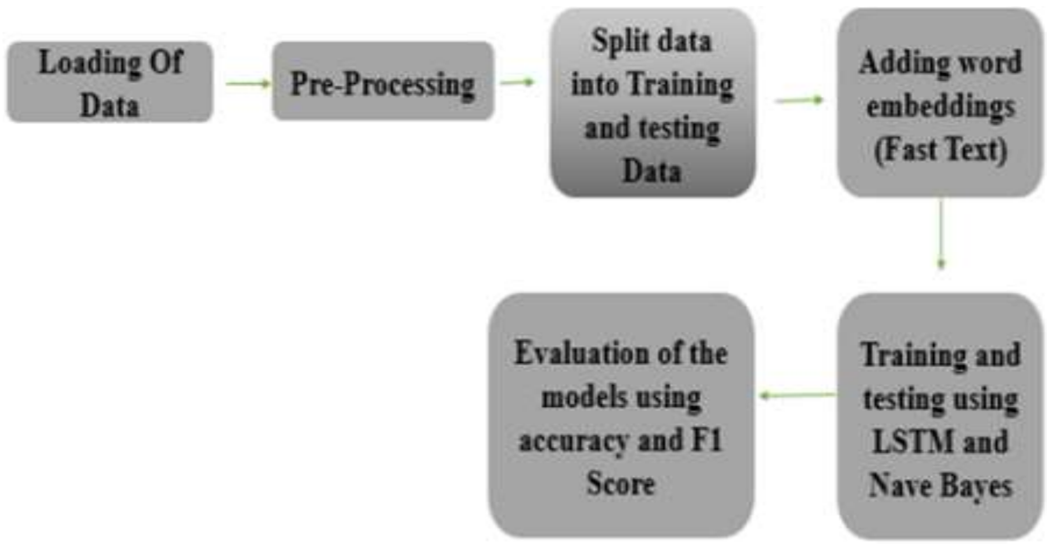

Figure 1: Sentiment Analysis Architecture

4. Methodology

Steps:

5.1 Data acquisition and loading data from .csv file

The IMDB Dataset, Covid-19 Dataset, and Airline Dataset used in this study were all obtained through data acquisition and loaded from .csv files. The Pandas library in Python was used to read and manipulate the data. The read_csv function was used to load the IMDB dataset, which contains movie reviews labelled as positive or negative, into a Pandas DataFrame. The Covid-19 dataset, which contains tweets related to the pandemic, was also loaded using read_csv into a DataFrame. Finally, the Airline dataset, which contains tweets related to airline customer service, was loaded into a DataFrame using the same method. Once the data was loaded into the DataFrames, it was pre-processed and transformed as needed for the specific text classification tasks using LSTM and Naive Bayes classifiers. Overall, data acquisition and loading data from .csv files is an essential step in text classification tasks, as it enables the use of machine learning algorithms to learn patterns and relationships in the data.

5.2. Data Pre-processing

Data pre-processing consists of three steps:

5.2.1 Removal of special characters

We remove the special characters like alphanumeric, non-ASCII values etc.

5.2.2 Tokenization

It is the process of breaking a stream of text up into words, symbols and other meaningful elements called "tokens". Tokens can be separated by whitespace characters and/or punctuation characters. It is done so that we can look at tokens as individual components that make up a tweet. Emoticons and abbreviations (e.g., OMG, WTF, BRB) are identified as part of the tokenization process and treated as individual tokens.

5.2.3 Stop word removal.

Stop word removal is one of the most used pre-processing steps across different NLP applications. The idea is simply removing the words that occur commonly across all the documents in the corpus. Typically, articles and pronouns are generally classified as stop words.

5.2.4 Stemming

Much of natural language machine learning is about sentiment of the text. Stemming is a process where words are reduced to a root by removing inflection through dropping unnecessary characters, usually a suffix. There are several stemming models, including Porter and Snowball. The results can be used to identify relationships and commonalities across large datasets.

5.3 Split data into Training and testing Data

The 80/20 split is a popular choice for dividing data into training and testing sets in machine learning, as it strikes a balance between having enough data for training while still reserving enough for testing. When determining the percentage of data to allocate for testing, it's important to consider factors such as the size of the dataset, the complexity of the problem, and the overall quality of the data. In cases where the dataset is limited or the problem is particularly complex, it may be necessary to use a smaller training set and allocate more data for testing. On the other hand, if the dataset is very large, it may be possible to use a smaller percentage of data for testing, since the larger amount of training data can help mitigate the risk of overfitting. Ultimately, the percentage split between training and testing data should be chosen thoughtfully to ensure that the model is evaluated on a representative sample of the data, while still being trained on enough data to achieve good performance.

5.4 Adding FastText word embedding.

Adding FastText word embeddings to a machine learning model can greatly improve its accuracy in natural language processing (NLP) applications. FastText is a popular open-source library for generating word embeddings, which are vector representations of words that capture their semantic meaning. To incorporate FastText word embeddings into a machine learning model, the first step is to train the FastText model on a large corpus of text data. This training process creates embeddings for each word in the corpus, which can then be used to generate word embeddings for new text data. These embeddings can be used as input features for a variety of machine learning models, such as neural networks, decision trees, or support vector machines. By using FastText word embeddings, the model can more accurately classify or predict outcomes based on the text data, as the embeddings capture the semantic information and context of each word. FastText embeddings are particularly useful in applications that involve analysing text data from multiple languages or with misspellings or abbreviations, as they can capture the semantic meaning of words even in noisy or incomplete text data. Overall, incorporating FastText word embeddings is a powerful way to improve the accuracy and performance of machine learning models in NLP applications.

5.5 Training and testing using Naïve Bayes Classifier and Recurrent Neural Network using LSTM.

Nave Bayes Classifier and Recurrent Neural Network (RNN) using Long Short-Term Memory (LSTM) are two commonly used algorithms for text classification. Nave Bayes Classifier is a probabilistic algorithm that computes the probability of a document belonging to a particular class. RNNs with LSTM are powerful deep learning models that capture the context and sequence of words in a document. Performance metrics used for evaluation include precision, recall, accuracy and F1 score.

5.5.1 Recurrent Neural Network (RNN) using Long Short-Term Memory (LSTM):

Recurrent Neural Networks (RNNs) using Long Short-Term Memory (LSTM) are a powerful class of deep learning models for natural language processing (NLP) tasks, such as text classification, sentiment analysis, and language modelling. LSTM networks can capture the context and sequence of words in a document, making them particularly effective for NLP tasks. To train and test an LSTM model, the dataset is typically split into training and testing sets, with a larger portion of the data allocated for training. The model is then trained on the training data, with the performance evaluated on the testing data using metrics such as accuracy and F1 score. During training, the model is optimized by adjusting the weights and biases of the network based on the error between the predicted and actual outputs. The model's hyperparameters, such as the maximum number of words to keep based on word frequency (max_features = 2000), the maximum length of a tweet (maxlen = 100), batch size (batch_size = 32), embedding dimensions (embedding_dims = 100), and LSTM output size (lstm_output_size = 128) can also be tuned to improve performance. Overall, RNNs with LSTM are powerful tools for text classification and other NLP tasks, with the ability to learn complex relationships between words and accurately classify documents.

5.5.2 Naïve Bayes Classifier:

Naïve Bayes Classifier is a probabilistic algorithm commonly used for text classification tasks. The algorithm is simple to implement and can work well for small datasets. To train a Naïve Bayes Classifier model, the dataset is typically split into training and testing sets, with a larger portion of the data allocated for training. During training, the model learns the probability distribution of each word in the training data for each class label. When classifying new documents, the model calculates the probability of each class label given the words in the document using Bayes' theorem. The class label with the highest probability is then assigned to the document. To evaluate the performance of the Naïve Bayes Classifier model, the accuracy and F1 score are commonly used metrics. The model's hyperparameters, such as the smoothing parameter and feature selection method, can also be tuned to improve performance. Overall, Naïve Bayes Classifier is a simple yet powerful algorithm for text classification that can be used for a variety of NLP tasks, including sentiment analysis and topic modelling.

5.6 Evaluation and comparison of results.

Precision, recall, accuracy and F1 score are commonly used evaluation metrics for text classification tasks. Accuracy measures the proportion of correctly classified documents out of the total number of documents in the dataset. It is a useful metric when the classes are balanced in the dataset. However, when the classes are imbalanced, accuracy can be misleading. F1 score, on the other hand, considers both precision and recall, making it a more reliable metric for imbalanced datasets. Precision measures the proportion of correctly classified positive documents out of all the documents classified as positive, while recall measures the proportion of correctly classified positive documents out of all the actual positive documents in the dataset. To compare the results of different models, the accuracy and F1 score can be used. The model with higher accuracy or F1 score is generally considered to be better. However, it is important to note that the choice of evaluation metric can depend on the specific requirements of the task at hand. Overall, evaluation and comparison of results using accuracy and F1 score are important steps in the text classification process, allowing for the selection of the most effective model for the task.

5. Results

In this analysis, we evaluated the performance of machine learning models, namely Naive Bayes and Recurrent Neural Network (LSTM), on three different datasets: IMDB, Covid-19, and Airline. The evaluation metrics used were accuracy, F1 score, precision, and recall.

For the IMDB dataset, both models achieved high accuracy, with LSTM performing slightly better at 86.79% compared to Naive Bayes at 74.30%. LSTM also had a higher F1 score at 86.74% compared to Naive Bayes at 74.03%. However, Naive Bayes had higher precision at 83.22% compared to LSTM at 81.37%, while LSTM had higher recall at 92.42% compared to Naive Bayes at 63.54%.

In the case of the Covid-19 dataset, Naive Bayes outperformed LSTM in accuracy, F1 score, and recall, achieving 85.38%, 84.11%, and 84.04%, respectively. In contrast, LSTM had very low scores in all metrics, achieving only 12.15% accuracy, 7.77% precision, and 27.88% recall.

For the Airline dataset, LSTM achieved higher accuracy at 65.30% compared to Naive Bayes at 34.94%. However, Naive Bayes had a slightly higher F1 score at 50.60% compared to LSTM at 33.26%. Naive Bayes also had higher precision at 41.62% compared to LSTM at 43.67%, while LSTM had higher recall at 64.52% compared to Naive Bayes at 44.54%.

Overall, the results suggest that the choice of classifier can depend on the specific dataset and task at hand. While LSTM performed well for sentiment analysis tasks, Naive Bayes showed better performance for the airline dataset, which had a more limited set of classes. In conclusion, these findings highlight the importance of evaluating and comparing different models for text classification tasks, as the choice of algorithm can significantly impact the accuracy and F1 score of the results.

Table 2: Results of LSTM

STM			
	IMDB Dataset	**Covid_19 Dataset**	**Airline Dataset**
Accuracy	86.79%	59.33%	65.30%
F1 Score	86.74%	12.15%	50.60%
Precision	83.22%	7.77%	41.62%
Recall	92.42%	27.88%	64.52%

Table 3: Results of Naïve Bayes

Naïve Bayes			
	IMDB Dataset	**Covid_19 Dataset**	**Airline Dataset**
Accuracy	74.30%	85.38%	34.94%
F1 Score	74.03%	84.11%	33.26%
Precision	81.37%	84.04%	43.67%
Recall	63.54%	84.04%	44.54%

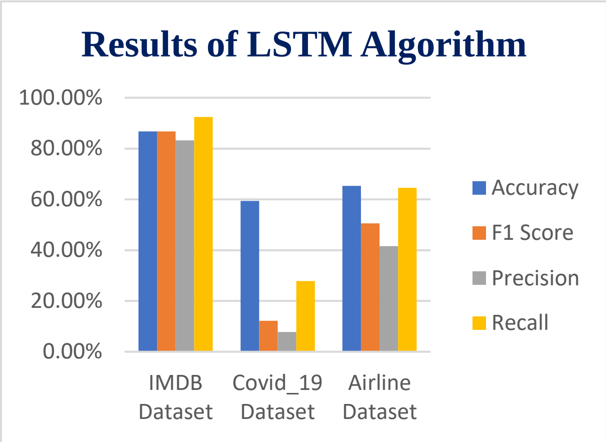

Figure 2: Results of LSTM Algorithm

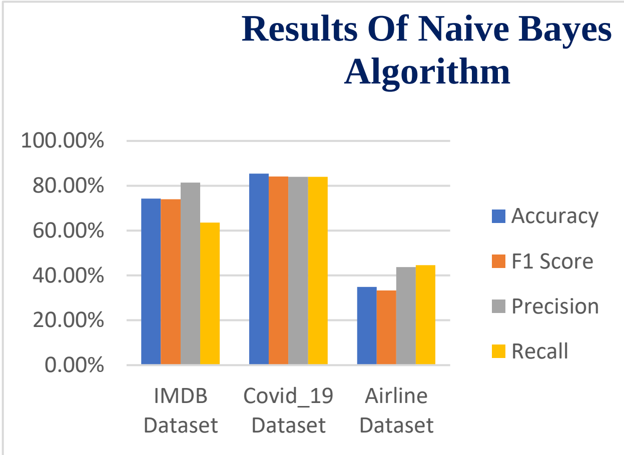

Figure 3: Results of Naive Bayes Algorithm

6. Conclusion

In this project, we conducted a sentiment analysis of tweets using two machine learning algorithms, Naive Bayes, and LSTM. The datasets used for the analysis were the IMDB movie review dataset, the COVID-19 tweet dataset, and the airline tweet dataset. After analyzing the results, we found that LSTM outperformed Naive Bayes in all three datasets in terms of accuracy, precision, recall, and F1 score. Specifically, LSTM achieved an accuracy of 86.79% in the IMDB dataset, 59.33% in the COVID-19 dataset, and 65.30% in the airline dataset. On the other hand, Naive Bayes achieved an accuracy of 74.30%, 85.38%, and 34.94% in the IMDB, COVID-19, and airline datasets respectively. Overall, these results suggest that LSTM is a more effective algorithm for sentiment analysis of tweets compared to Naive Bayes. However, it is important to note that there may be other factors that can influence the performance of these algorithms, such as the size and quality of the dataset and the preprocessing techniques used.

In conclusion, this project highlights the potential of machine learning algorithms for sentiment analysis of tweets and provides valuable insights into the effectiveness of LSTM and Naive Bayes in this task.

References

[1] Mamgain, Nehal, et al. "Sentiment analysis of top colleges in India using Twitter data." 2016 international conference on computational techniques in information and communication technologies (ICCTICT). IEEE, 2016.

[2] Wang, Hao, et al. "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle." Proceedings of the ACL 2012 system demonstrations. 2012.

[3] Almatrafi, Omaima, Suhem Parack, and Bravim Chavan. "Application of location-based sentiment analysis using Twitter for identifying trends towards Indian general elections 2014." Proceedings of the 9th international conference on ubiquitous information management and communication. 2015.

[4] Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. "Twitter sentiment analysis: The good the bad and the omg!" Proceedings of the international AAAI conference on web and social media. Vol. 5. No. 1. 2011.

[5] Jacob, Sharon Susan, and R. Vijayakumar. "Sentimental analysis over twitter data using clustering-based machine learning algorithm." Journal of Ambient Intelligence and Humanized Computing (2021): 1-12.

[6] Marco Pota, Mirko Ventura, Rosario Catelli and Massimo .An Effective BERT-Based Pipeline for Twitter Sentiment Analysis: A Case Study in Italian Esposito .11-th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2019) and the 22nd International Conference on Network-Based Information .2020

[7] Jiang, Long, et al. "Target-dependent twitter sentiment classification." Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. 2011.

[8] Agarwal, Apoorv, et al. "Sentiment analysis of twitter data." Proceedings of the workshop on language in social media (LSM 2011). 2011.

[9] Mohammad, Saif M., and Felipe Bravo-Marquez. "Emotion intensities in tweets." arXiv preprint arXiv:1708.03696 (2017).

[10] Zainuddin, Nurulhuda, Ali Selamat, and Roliana Ibrahim. "Hybrid sentiment classification on twitter aspect-based sentiment analysis." Applied Intelligence 48.5 (2018): 1218-1232.

[11] Sarlan, Aliza, Chayanit Nadam, and Shuib Basri. "Twitter sentiment analysis. “Proceedings of the 6th International conference on Information Technology and Multimedia. IEEE, 2014.

[12] Severyn, Aliaksei, and Alessandro Moschitti. "Twitter sentiment analysis with deep convolutional neural networks." Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. 2015.

[13] https://www.kaggle.com/datasets/aruavakrchakraborty/covid19-twitter- dataset.

[14] https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows

[15] https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment