

PSYCHOLOGICAL ANALYSIS USING SOCIAL MEDIA TWEETS

A PROJECT REPORT

Submitted by

KARTIK SINGH [Reg No:RA2011003010810]

UTKARSH GUPTA [Reg No: RA2011003010772]

Under the Guidance of

Mrs. P. Renukadevi

Assistant Professor, Department of Computing Technologies

in partial fulfillment of the requirements for the degree of

**BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTING TECHNOLOGIES
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR– 603 203**


MAY 2024



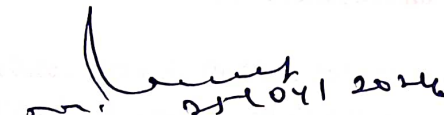
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603 203


BONAFIDE CERTIFICATE

Certified that 18CSP109L project report titled “PSYCHOLOGICAL ANALYSIS OF DEPRESSION USING SOCIAL MEDIA TWEETS ” is the bonafide work of Kartik Singh [Reg No: RA2011003010810] and Utkarsh Gupta [Reg No: RA2011003010772] who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other report on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.


Mrs. P. RENUKADEVI 25/4/24
SUPERVISOR
Assistant Professor
Department of Computing Technologies




Dr. M. BASKAR
PANEL HEAD
Associate Professor
Department of Computing Technologies


Dr. M. PUSHPALATHA
PROFESSOR AND HEAD
Department of Computing Technologies

INTERNAL EXAMINER

EXTERNAL EXAMINER



Department of Computing Technologies
SRM Institute of Science and Technology
Own Work Declaration Form

Degree/Course : B. Tech in Computer Science and Engineering

Student Names : KARTIK SINGH, UTKARSH GUPTA

Registration Number: RA2011003010810, RA2011003010772

Title of Work : Psychological Analysis Using Social Media Tweets

I/We here by certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas quoted text (books, web, etc.)
- Given the sources of all pictures, data etc that are not my own.
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (fellow students, technicians, statisticians, external sources)

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, that I have followed the good academic practices noted above.

Student 1 Signature:

Student 2 Signature:

Date: 25/04/24

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. T. V. Gopal**, Dean-CET, SRM Institute of Science and Technology for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. M. Pushpalatha**, Professor, Department of Computing Technology, SRM Institute of Science and Technology for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators: **Dr. S. Godfrey Winster**, Associate Professor, Project Panel Head, **Dr. M. Baskar**, Associate Professor, Project Panel Members: **Mrs. R. Brindha**, Assistant Professor, **Mrs. P. Renukadevi**, Assistant Professor, **Dr. M. K. Vidhyalakshmi**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Mrs. P. Renuka Devi**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology and **Dr. M Senthilraja**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide **Mrs. P. Renukadevi** Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology for providing us with an opportunity to pursue our project under her mentorship.

She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computing Technologies Department, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement.

KARTIK SINGH [RA2011003010810]

UTKARSH GUPTA [RA2011003010772]

ABSTRACT

In an era where depression has emerged as a prevalent and challenging mental health issue, this project, titled "Psychological Analysis of Depression Using Social Media (Twitter)," seeks to harness the vast data resources of social media, particularly Twitter, to gain profound insights into the psychological facets of depression. Employing advanced natural language processing and sentiment analysis techniques, we delve into the linguistic and temporal patterns present in tweets from users who identify as depressed or exhibit symptoms of depression. Simultaneously, we explore the intricate web of social dynamics by examining interactions between individuals expressing depression and their Twitter communities, aiming to understand how these connections influence the perception and experience of depression, including the presence of positive or negative feedback loops. The outcomes of this study have the potential to enhance our understanding of depression, offering valuable insights for mental health professionals, researchers, and policymakers, and providing a foundation for more effective interventions, early detection methods, and support systems to aid those grappling with depression in the digital age.

TABLE OF CONTENTS

ABSTRACT	vi
LIST OF FIGURES	ix
LIST OF SYMBOLS AND ABBREVIATIONS	x
1. INTRODUCTION	1
1.1 General	1
1.2 Motivation	2
1.3 Objectives	3
1.4 Software Requirement Specification	5
1.5 Machine Learning Healthcare	6
1.6 Convolutional Neural Network	8
1.7 Long Short-Term Memory	10
2 LITERATURE SURVEY	12
2.1 Sentiment Analysis Approach on Twitter	12
2.2 Machine Learning Perspective on Twitter	12
2.3 Unveiling Distress Signals	12
2.4 Machine Learning Models for Harassment Detection	13
2.5 Multi-Technique Approach for Detection and Intervention	13
2.6 Depression Detection on Social Media	13
2.7 Depression Detection in Twitter tweets	13
2.8 Depression Analysis in Twitter tweets	14
2.9 Unified benchmark and comparative evaluation	14
3 SYSTEM DESIGN	15
3.1 Architecture Diagram	15
3.2 Proposed Module	16

4	DESIGN AND IMPLEMENTATION	21
4.1	Dataset	21
4.2	Data Preprocessing	24
4.3	Tokenization	26
4.4	Embedding Matrix	29
4.5	Model Building	31
4.6	Training the Model	33
5	RESULT AND DISCUSSIONS	35
6	CONCLUSION AND FUTURE SCOPE	38
6.1	Conclusion	38
6.2	Future Scope	39
	REFERENCES	42
	APPENDIX 1	44
	APPENDIX 2	62
	PLAGIARISM REPORT	67

LIST OF FIGURES

Fig No.	Figure Name	Page
1.1	Convolution Neural Networks	8
1.2	Long Short-Term Memory	10
3.1	Architecture LSTM+CNN	13
3.2	Block Diagram	20
4.1	TWINT Dataset	22
4.2	Twitter Dataset	23
4.3	Fitting Tokens	27
4.3	Counting Unique Words	28
4.3	Pad Sequence	29
4.4	Compile Model	33
5.1	Accuracy	35
5.2	ROC Curve	36
5.2 (a)	Model Accuracy	37
5.2 (b)	Model Loss	37

LIST OF SYMBOLS AND ABBREVIATIONS

Symbols	Abbreviation
US	United States of America
ReLU	Rectified Linear Unit
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
AI	Artificial Intelligence
ML	Machine Learning
NLTK	Natural Language Toolkit
NLP	Natural Language Processing
TWINT	Twitter Intelligence Tool

CHAPTER 1

INTRODUCTION

1.1 General

In the digital age, the proliferation of social media platforms has revolutionized the way we communicate and share our lives with the world. Social media, especially Twitter, has emerged as a powerful medium through which individuals express their thoughts, feelings, and experiences. It provides a vast and easily accessible source of textual data, which, when harnessed effectively, can offer valuable insights into the psychological well-being of its users. In light of the growing concern surrounding mental health issues, particularly depression, there is a pressing need for innovative approaches that can identify, analyze, and ultimately provide support to individuals experiencing such challenges.

The final year project titled "Psychological Analysis and Detection of Depression Using Social Media (Twitter)" represents a pioneering effort in this domain. It seeks to leverage the capabilities of machine learning and natural language processing to build a sophisticated system capable of identifying potential signs of depression in the content shared on Twitter. Twitter, with its vast and diverse user base, serves as the primary data source, as it offers a treasure trove of text-based information where users candidly share their thoughts, emotions, and daily experiences.

The project will involve a multi-faceted approach, beginning with the collection of extensive Twitter data. This data will then undergo preprocessing and cleaning to extract relevant textual content. Subsequently, state-of-the-art machine learning algorithms will be employed to scrutinize the linguistic patterns, sentiment, and contextual information present in the tweets. By identifying specific linguistic markers, behavioural patterns, and other telltale signs associated with depression, the system aims to detect potential cases of depression in social media users.

The overarching goal of this project is to create a tool that can not only recognize individuals at risk of depression but also offer them timely interventions or support mechanisms. By proactively identifying and reaching out to those who may be silently suffering, the system aims to make a meaningful impact on mental health awareness and support. The insights derived from this research have the potential to enhance our understanding of the complex

interplay between social media activity and mental health. Moreover, it opens the door to new possibilities in creating effective, scalable, and user-friendly tools that can address the burgeoning issue of depression in our digitally interconnected world.

In sum, "Psychological Analysis and Detection of Depression Using Social Media (Twitter)" represents a significant stride towards utilizing cutting-edge technology and the wealth of social media data for the betterment of mental health. By combining machine learning, natural language processing, and the vast resource of Twitter, the project aspires to contribute to a proactive and compassionate approach to identifying and assisting those grappling with depression, ultimately making our online world a safer and more supportive space for individuals in need.

1.2 Motivation

The motivation for undertaking the project titled "Psychological Analysis and Detection of Depression Using Social Media (Twitter) with Machine Learning" is driven by several key factors:

- **Growing Mental Health Concerns:** Mental health issues, particularly depression, have become a significant public health concern worldwide. The prevalence of depression is on the rise, and it affects individuals across all age groups. Early detection and intervention are crucial in managing depression effectively.
- **Impact of Social Media:** In recent years, social media platforms, such as Twitter, have become integral parts of people's lives. Individuals often express their thoughts, feelings, and emotions on these platforms. This provides a unique opportunity to gather valuable data for psychological analysis.
- **Machine Learning Advancements:** The field of machine learning has made remarkable strides in recent years, especially in natural language processing and sentiment analysis. These advances allow us to process and analyze vast amounts of text data, making it feasible to detect patterns indicative of depression in social media posts.
- **Potential for Early Intervention:** Identifying individuals at risk of depression early can lead to timely intervention and support. By analyzing social media posts, we can potentially detect signs and symptoms of depression before they become severe, ultimately improving the overall mental well-being of individuals.

- **Reducing Stigma:** Depression is often stigmatized, leading many individuals to underreport their symptoms or avoid seeking help. Anonymously analyzing social media data allows for a more discreet and less stigmatized way of identifying those who may be experiencing depression.
- **Contribution to Research:** This project contributes to the broader body of research on mental health and artificial intelligence. It explores the feasibility of leveraging machine learning techniques to address real-world mental health challenges and offers the opportunity to develop and test new methodologies.
- **Practical Application:** The project has the potential to result in a practical tool or system that can be deployed to assist mental health professionals, caregivers, and individuals in identifying signs of depression in an automated and data-driven manner.
- **Interdisciplinary Learning:** The project encompasses various fields, including psychology, machine learning, natural language processing, and data analysis. It offers an excellent opportunity for interdisciplinary learning and the application of knowledge from multiple areas of study.
- **Personal Growth:** Undertaking a project of this nature presents a significant learning opportunity. It allows for the development of technical and analytical skills while also raising awareness about mental health issues and the importance of addressing them.

In conclusion, the project's motivation lies in its potential to contribute to the well-being of individuals, advance the field of machine learning and psychology, and provide a platform for learning and personal growth. Detecting depression through social media analysis has the potential to be a valuable and innovative approach in the ongoing efforts to tackle mental health issues.

1.3 Objectives

The primary goal of the project is to leverage data from social media platforms to address the issue of depression, with a specific focus on early detection. This overarching objective can be broken down into several key components:

1. **Early Detection:** The project's core aim is to create advanced algorithms that can identify signs of depression within social media content. This involves analyzing text data for specific keywords, phrases, and patterns that are associated with depressive symptoms. By

doing so, the project hopes to identify individuals who may be experiencing depression at an early stage, before it escalates or becomes more severe.

2. **Risk Assessment:** In addition to detecting depression, the project seeks to assess the severity of the condition based on the content of social media posts. By categorizing individuals into groups with mild, moderate, or severe symptoms, the project can prioritize its interventions and resources, ensuring that those at greater risk receive more immediate attention and support.

3. **Individualized Support:** Collaboration with mental health experts is a crucial part of the project. By working closely with professionals in the field, the project aims to incorporate clinical insights and expertise into the algorithm's design. This ensures that the algorithm is not only effective in detecting depression but also that it aligns with clinical standards and can provide personalized support to individuals in need.

4. **High-Risk Group Identification:** The project recognizes the importance of identifying demographic or community groups that are more susceptible to depression. By analyzing social media data, the project can pinpoint specific groups that may be at higher risk, allowing for more targeted and effective intervention strategies.

5. **Validation with Professionals:** To ensure the real-world effectiveness of the algorithm, the project plans to conduct comprehensive validation and testing in both real-world scenarios and clinical settings. This validation process will measure the algorithm's ability to correctly identify depression cases and assess its impact on improving mental health outcomes.

6. **Privacy and Ethics:** Recognizing the sensitivity of mental health data and the potential ethical concerns surrounding its collection and analysis, the project is committed to maintaining strict privacy and ethical standards. This includes safeguarding user data and interactions, ensuring that the project adheres to established privacy regulations, and maintaining a high level of ethical conduct throughout all stages of the project.

1.4 Software Requirement Specification

Our study and project uses:

- Python 3
- Jupyter Notebook

- PyCharm
- Various Python Libraries like: Keras, nltk, tensor flow, sklearn, ftfy, gensim, matplotlib

Software Requirements:

Our project will require various software tools and libraries to execute the above tasks effectively.

Here's a list of the software and libraries you will be using:

Python 3:

Python is the primary programming language for implementing machine learning and deep learning models. It provides a wide range of libraries and tools for data analysis, preprocessing, and model building.

Jupyter Notebook:

Jupyter Notebook is an interactive environment that allows you to write and execute Python code in an easily shareable and documentable format. It's excellent for data exploration and analysis.

PyCharm:

PyCharm is a popular integrated development environment (IDE) for Python. It offers powerful features for coding, debugging, and managing projects.

Python Libraries:

We used various Python libraries, including:

Keras: A high-level deep learning library that simplifies building and training neural networks.

nltk (Natural Language Toolkit): Used for natural language processing tasks such as text preprocessing.

TensorFlow: A deep learning framework that provides a wide range of tools for building and training machine learning models.

Scikit-learn (sklearn): A machine learning library that includes tools for data preprocessing, model selection, and evaluation. **ftfy:** A library for text data cleaning and fixing.

Gensim: Used for word embedding and natural language processing tasks.

Matplotlib: A popular library for data visualization.

By using these software tools and libraries, you'll be well-equipped to collect, preprocess, and analyze data from social media to understand and predict depression based on textual and image data using the proposed hybrid LSTM-CNN architecture. It's important to stay up-to-date with the latest developments in these libraries, as the field of deep learning is rapidly evolving.

1.5 Machine Learning in Healthcare

Machine learning in healthcare, especially within the context of mental health, is a burgeoning field that offers immense potential for transforming the understanding, diagnosis, and treatment of conditions such as depression. In the context of my project, "Psychological Analysis of Depression Using Social Media (Twitter)," machine learning plays a pivotal role in several aspects.

1. **Early Detection and Diagnosis:** Machine learning algorithms can analyze large datasets of patient information, including social media activity, to identify early warning signs of mental health issues such as depression. By recognizing subtle patterns in language, behaviour, and online interactions, machine learning can assist in the early detection and diagnosis of mental health disorders.
2. **Personalized Treatment Plans:** Machine learning models can be used to create personalized treatment plans for individuals with mental health conditions. These plans can take into account a patient's unique characteristics and needs, optimizing the effectiveness of interventions and therapies.
3. **Sentiment Analysis:** Machine learning can perform sentiment analysis on social media posts to gauge an individual's emotional state. This data can be valuable for mental health professionals to monitor the well-being of their patients between appointments and to intervene when necessary.

4. **Risk Assessment:** Machine learning algorithms can assess the risk of self-harm or suicide by analyzing a person's online posts. When concerning patterns are detected, mental health providers and crisis helplines can be alerted to provide timely support.
5. **Predictive Modelling:** Machine learning can build predictive models to anticipate mental health crises, relapses, or adverse events. These models use historical data to forecast future outcomes, allowing for proactive intervention and support.
6. **Resource Allocation:** In the mental health sector, machine learning can help healthcare organizations allocate resources more efficiently. By analyzing patient data and demand for services, it can optimize the distribution of mental health professionals and facilities.
7. **Natural Language Processing (NLP):** NLP techniques are integral to understanding the language used by individuals with mental health conditions. Machine learning models can analyze text and speech to detect signs of depression, anxiety, or other disorders, enhancing diagnosis and treatment.
8. **Reducing Stigma:** Machine learning can help in de-stigmatizing mental health issues. By providing insights into the prevalence and diversity of mental health experiences, it contributes to a better understanding and acceptance of these conditions in society.
9. **Patient Empowerment:** Machine learning-driven mental health apps and tools empower patients to monitor their own mental health. These apps can provide personalized coping strategies, suggest mindfulness exercises, and offer immediate support when needed.
10. **Research Advancements:** Machine learning accelerates mental health research by processing and analysing vast datasets quickly. It aids in identifying new correlations, risk factors, and potential treatments, ultimately advancing our understanding of mental health conditions.
11. **Ethical and Privacy Concerns:** As machine learning in mental health evolves, it is essential to address ethical concerns related to data privacy, security, and the responsible use of sensitive information. Ensuring that patient data is anonymized, protected, and used for beneficial purposes is a crucial consideration.

In summary, machine learning holds immense potential in the mental health sector, transforming the way we detect, diagnose, treat, and support individuals with mental health conditions. While it offers numerous benefits, it is essential to approach this technology with a strong focus on ethics and privacy to ensure its responsible and safe application in mental healthcare.

1.6 Convolution Neural Networks

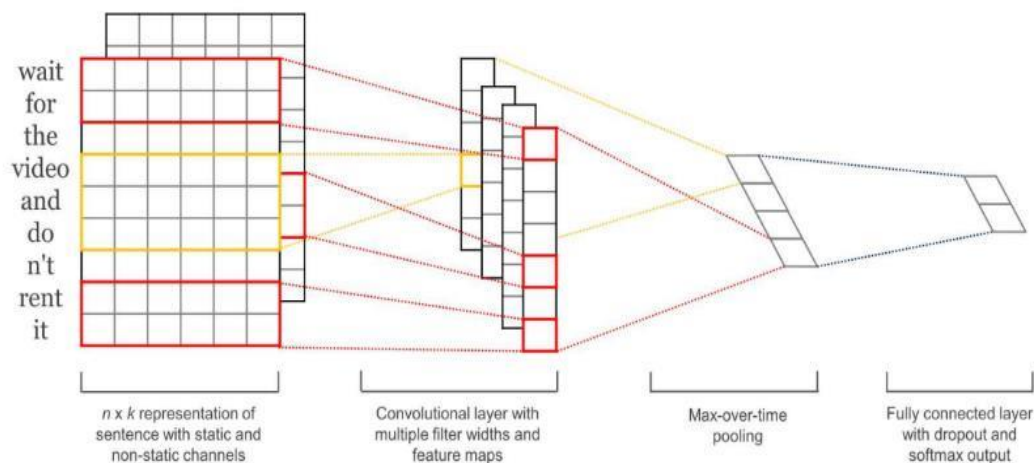


Fig. 1.1 Convolution Neural Networks

Initially designed for image recognition, Convolutional Neural Networks (CNN) have become an incredibly versatile model used for a wide array of tasks. CNNs have the ability of recognizing local features inside a multi-dimensional field. For example, on an image, CNNs will be able to spot particular features such as a wheel or a smile, regardless of where these might be located. Basic CNNs (as outlined in Figure 1.1) work by feeding multidimensional data (e.g. images, word embeddings, etc.) to a Convolutional layer which will be composed of multiple filters that will learning different features. Notice that these filters are sequentially applied to different sections of the input. The output is usually pooled or sub-sampled to smaller dimensions and later fed into a connected layer.

In the context of the psychological analysis of depression using tweets, Convolutional Neural Networks (CNNs) offer a valuable tool. Originally developed for image recognition, CNNs have demonstrated their adaptability to a diverse range of data types and have evolved into versatile models. In the case of Twitter data analysis, this versatility is especially advantageous. CNNs excel at recognizing local features within multi-dimensional data fields,

such as identifying specific keywords, emotional expressions, and sentiments within tweets. This ability to identify local features, regardless of their location in the text, is essential for detecting subtle cues related to depression.

Furthermore, Twitter data often includes a variety of data types, making it multimodal. CNNs are well-suited to handle this complexity, as they can efficiently process textual content while also being capable of analyzing associated images or emojis if present. The parallel processing capability of CNNs enables simultaneous evaluation of various aspects within the input, enhancing the speed and efficiency of sentiment analysis, a critical aspect of depression detection.

Additionally, CNNs are known for their adaptability and flexibility. They can be fine-tuned to suit specific applications, whether processing textual or visual data. This adaptability allows them to provide insightful results in diverse contexts. By integrating CNNs into the hybrid LSTM-CNN architecture, the project benefits from improved accuracy and nuance in sentiment analysis. CNNs can identify specific emotional features and patterns in text, contributing to a more comprehensive understanding of a user's emotional state, a crucial factor in detecting signs of depression.

Therefore, CNNs are an appropriate choice for the project, capable of efficiently processing the multimodal Twitter data and contributing to the precision of depression analysis.

1.7 Long Short-Term Memory

Long-Short Term Memory (LSTM) networks are a type of Recurrent Neural Network architecture that is designed to “remember” previously read values for any given period of time. LSTMs usually contain three gates that control the flow to and from their memories. The “input gate” controls the input of new information to the memory. The “forget gate” controls how long certain values are held in memory.

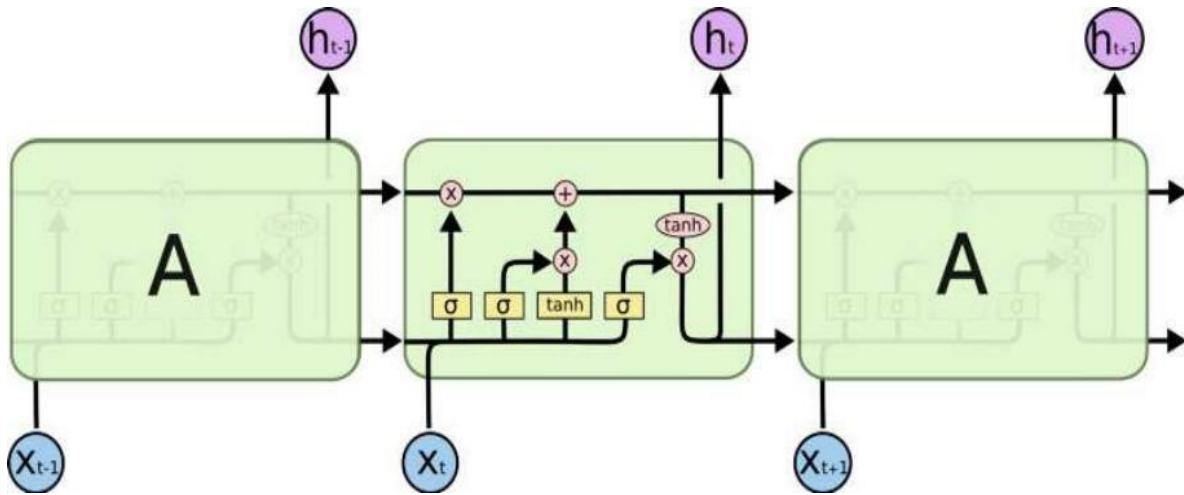


Fig. 1.2 Long Short-Term Memory

Finally, the “output gate” controls how much the value stored in memory affects the output activation of the block. In the context of the project focused on psychological analysis of depression using tweets, the integration of Long Short-Term Memory (LSTM) neural networks is of significant relevance. LSTM networks are particularly well-suited for handling sequential data, a characteristic that is pivotal in analyzing text data like tweets.

The key advantage of LSTM lies in its ability to capture temporal dependencies within the data, considering the order and sequence of words and phrases. This sequential processing capability is crucial for comprehending the nuanced language and emotional expressions often conveyed in tweets, making it an ideal choice for accurate sentiment analysis and depression detection.

Moreover, tweets often contain complex emotional signals, which necessitate an understanding of the context in which words and phrases are used. LSTM excels in grasping the context of textual content, allowing for a more precise assessment of emotional content. It can distinguish between positive and negative expressions and is sensitive to linguistic subtleties that might serve as indicators of depression. The analysis of Twitter data for signs of depression entails dealing with both the intricacies of text and the temporal aspects of emotional fluctuations. LSTM's ability to model sequences over time is invaluable in capturing these temporal nuances. It can interpret evolving emotional states in Twitter users by recognizing how sentiments change within or across tweets.

In addition, depression-related signals often require the network to consider long-term dependencies, which may be deeply embedded within the context of tweets. LSTM is well equipped to retain information over extended sequences, enabling the identification of subtle patterns and emotional fluctuations that may span across multiple tweets or within a single tweet.

By focusing on the sequential and contextual aspects of tweets, LSTM contributes to high precision in sentiment analysis. Its depth and understanding of the nuances in language and emotional expression complement the capabilities of other components, such as Convolutional Neural Networks (CNN), within the hybrid architecture. This integration enhances the project's ability to detect depression-related signals in Twitter data with accuracy and depth.

CHAPTER 2

LITERATURE SURVEY

The exploration of mental health in the context of social media, particularly the detection and analysis of depression, has attracted significant attention. This literature review aims to provide a comprehensive overview of recent studies investigating the relationship between depression and social media platforms, with a specific emphasis on Twitter.

2.1 Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment analysis of comment texts based on LSTM: Study tackles the daunting challenges associated with sentiment analysis on Twitter. They recognize the sparse and often contextually limited nature of social media data, especially concerning expressions of depression. Their approach involves a meticulous process of data collection, preprocessing, and classification techniques, leveraging both unigram and bigram processing to discern nuanced patterns of depressive sentiment within the vast landscape of Twitter conversations [1].

2.2 Tammaro, A. M., Tomaiuolo, M., Mordonini, M., Pellegrino, M., & Demicelis, R. (2022). Building a Sentiment Analysis Model for Libraries: Research delves into the realm of disorder detection within Twitter posts, utilizing a diverse array of natural language processing (NLP) and machine learning algorithms. Their exploration spans methodologies such as Support Vector Machines (SVM), Logistic Regression (LR), Long Short-Term Memory networks (LSTM), Convolutional Neural Networks (CNN), and ensemble methods. By employing these sophisticated techniques, they aim to identify subtle indicators of depression amidst the noise of social media discourse [2].

2.3 Meng, W., Wei, Y., Liu, P., Zhu, Z., & Yin, H. (2019). Aspect based sentiment analysis with feature enhanced attention CNN-BiLSTM: Investigation focuses on the detection of distress signals within Twitter tweets. They employ a diverse range of machine learning classifiers, including Naïve Bayes, decision trees, Support Vector Machines (SVM), random forests, and Convolutional Neural Networks (CNN). By leveraging these techniques, they seek to uncover the complex interplay of emotions and expressions indicative of underlying psychological distress among Twitter users [3].

2.4 Ikram, A., Kumar, M., & Munjal, G. (2022, January). Twitter Sentiment Analysis using Machine Learning: Study delves into the accuracy of machine learning models in detecting harassment on social media platforms, including Twitter. Their research employs a variety of algorithms, such as Naive Bayes, Support Vector Machines (SVM), K-Nearest Neighbour (KNN), Logistic Regression (LR), and Linear SVM (LSVM). By investigating the effectiveness of these models, they shed light on the multifaceted nature of online interactions and their potential impact on mental well-being, including the exacerbation of depressive symptoms [4].

2.5 Kumar, S. K., Dinesh, (2022, May). Depression detection in Twitter tweets using machine learning classifiers: Ahmed and Lin's research explores the impact of depression on social media engagement and avenues for its detection. Their approach encompasses a variety of techniques, including personal journaling, Graph Attention Networks (GAT), self-focus tasks, and emotional vocabulary expansion. By employing these innovative strategies, they aim to understand how depressive symptoms manifest in online contexts and to develop effective interventions for identifying and addressing these symptoms [5].

2.6 Liaw, A. S., & Chua, H. N. (2022, September). Depression Detection on Social Media

With User Network and Engagement Features Using Machine Learning: The paper explores depression detection on social media, leveraging user network and engagement features with machine learning. They review existing research on the topic, highlighting correlations between social media use and mental health. Emphasizing the limitations of current methodologies, they propose a novel approach that integrates user behaviour analysis with machine learning algorithms, aiming to improve accuracy and early detection of depression [6].

2.7 Nitha, L. (2022, May). Depression detection in Twitter tweets using machine learning: The paper explores depression detection on social media using user network and engagement features alongside machine learning. Their literature survey likely outlines the significance of depression in society and reviews existing methodologies, including linguistic analysis, while highlighting limitations. They introduce potential of user behaviour analysis for depression detection, setting the stage for their innovative approach. Additionally, they

discuss relevant research on machine learning techniques and ethical considerations. Through their review, they identify gaps and propose a novel methodology integrating user behaviour analysis with machine learning for improved depression detection on social media [7].

2.8 Dinesh K. (2022, May). Depression detection in Twitter tweets using machine learning classifiers: The focus lies on depression detection within Twitter tweets through the application of machine learning classifiers. Their literature survey likely commences with an exploration of the prevalence of depression and the growing interest in leveraging social media for mental health analysis. They likely review existing methodologies for depression detection on Twitter, including linguistic analysis and sentiment analysis. Furthermore, they may discuss previous studies utilizing machine learning classifiers for this purpose, highlighting the strengths and limitations of these approaches. The authors likely identify gaps in the literature, paving the way for their own research aiming to enhance depression detection accuracy on Twitter through the utilization of machine learning techniques [8].

2.9 Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). Tweeteval: Unified benchmark and comparative evaluation for tweet: A comparative evaluation framework for tweets. Their literature survey likely begins by discussing the importance of standardized evaluation frameworks in natural language processing tasks, particularly within the context of social media analysis. They may review existing benchmarks and evaluation methodologies for tweet-related tasks, highlighting the need for a unified framework that covers multiple tasks. The authors likely discuss the challenges and limitations of previous approaches, setting the stage for the introduction of Tweeteval as a comprehensive and standardized evaluation platform. Through their review, Barbieri et al. lay the groundwork for their contribution to the field, aiming to provide researchers with a common benchmark for evaluating various tweet-related tasks and fostering progress in natural language processing research [11].

In conclusion, the studies discussed in this literature review collectively provide valuable insights into the understanding of depression within the realm of Twitter and other social media platforms. The findings underscore the importance of continued research in this area, paving the way for the development of effective strategies to support individuals experiencing depression in online communities.

CHAPTER 3

SYSTEM DESIGN AND ARCHITECTURE

3.1 Architecture Diagram

The design of the system for analyzing depression on Twitter integrates LSTM (Long-Term and Short-Term Memory) and CNN (Convolutional Neural Network) models strategically chosen for their unique abilities in handling sequential data and extracting spatial patterns from text. LSTMs are particularly effective at capturing long-term dependencies in Twitter conversations. By combining these models, the system can comprehensively analyze both the sequential and textual features of tweets, providing deeper insights into discussions related to depression on Twitter.

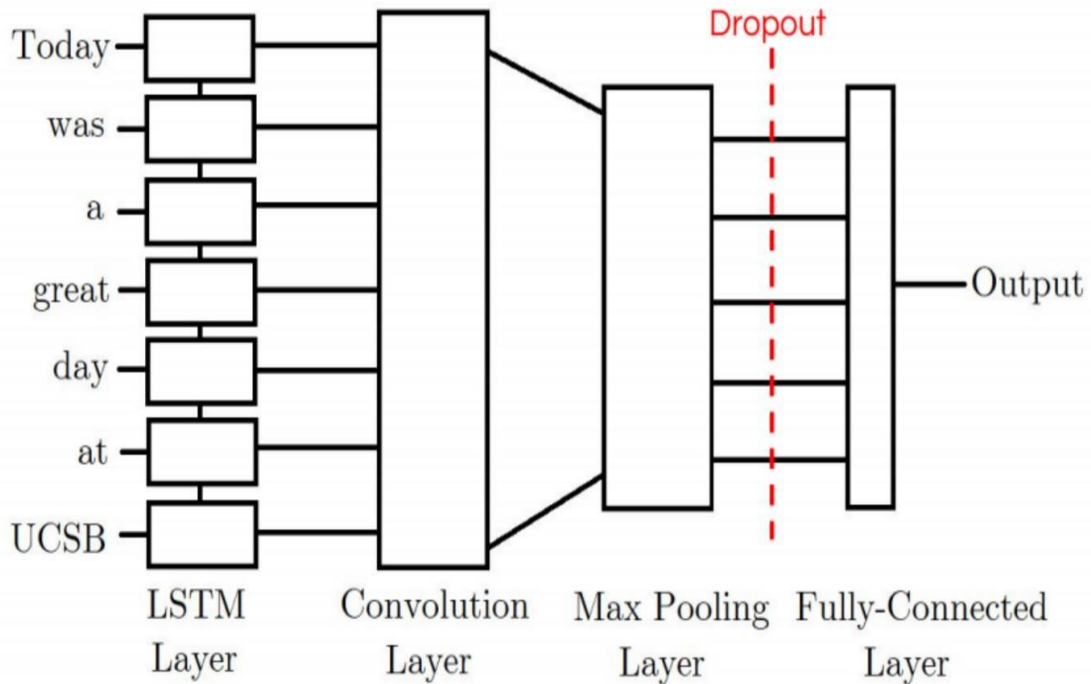


Fig. 3.1 Architecture LSTM+CNN

Alongside the utilization of advanced machine learning techniques, the system design incorporates a robust pipeline for data collection and preprocessing. Various sources such as Twint and the Kaggle Twitter Sentiment Dataset are employed to gather a diverse range of tweet data for analysis. Preprocessing steps, including tokenization and the generation of

embedding matrices, are implemented to clean and organize raw tweet data effectively, ensuring optimal performance of the machine learning models.

The model is trained using dedicated training and validation datasets, with measures in place to mitigate the risk of overfitting, such as dropout layers and early stopping mechanisms. Through this approach, the system can accurately classify tweets and identify symptoms of depression, thereby offering valuable insights into mental health discussions on Twitter.

Overall, the system design facilitates a comprehensive analysis of depression on Twitter by integrating sophisticated techniques with robust data collection and preprocessing methods, contributing to a deeper understanding of mental health issues within online communities.

3.2 Proposed Module

The choice of an LSTM (Long Short-Term Memory) + CNN (Convolutional Neural Network) model for the project, "Psychological Analysis of Depression Using Social Media (Twitter)," is highly appropriate for several compelling reasons:

1. Sequential Data Handling: Analyzing Twitter data, which consists of tweets posted in chronological order, poses a unique challenge due to its sequential nature. Long Short-Term Memory (LSTM) networks offer a solution for processing such data effectively. LSTMs are designed to retain information about past inputs, allowing them to make predictions based on the context of previous data points. This capability makes LSTMs well-suited for capturing the evolving language patterns and emotional expressions found in tweets related to depression. By analyzing the sequential patterns within tweets, LSTMs can uncover subtle changes in mood, linguistic style, and the progression of thoughts and feelings over time, providing valuable insights into users' mental health experiences.

LSTMs serve as a valuable tool for understanding the intricate language patterns and emotional shifts associated with this sensitive topic. Depression manifests differently for each individual and can be expressed in various ways through language. LSTMs excel at detecting these nuances by learning from the sequential structure of tweets and identifying meaningful patterns and connections over time. Leveraging the capabilities of LSTMs enables researchers and mental health professionals to gain deeper insights into the

experiences of individuals grappling with depression on Twitter, potentially leading to more effective support and intervention strategies.

2. Textual Analysis: Long Short-Term Memory (LSTM) networks have proven highly effective in the realm of natural language processing, particularly in tasks such as sentiment analysis. In the context of depression-related tweets, LSTMs can be trained to discern emotional cues and linguistic patterns indicative of depressive states. By analyzing large volumes of text data, LSTMs learn to recognize subtle nuances in language use, allowing them to identify expressions of sadness, despair, or hopelessness that may signal underlying mental health issues. Additionally, the Convolutional Neural Network (CNN) component, when integrated into the LSTM architecture, further enhances the model's capabilities. CNNs specialize in feature extraction from text, enabling them to identify important linguistic features or structural elements within tweets. This feature extraction process helps the LSTM model to capture more meaningful information from the text, improving its ability to understand and interpret the emotional content of depression-related tweets.

In summary, the combination of LSTM and CNN architectures forms a powerful framework for textual analysis, particularly in the context of sentiment analysis and understanding emotional expressions in text data. By leveraging the strengths of both models, researchers and practitioners can develop robust systems for identifying and analyzing depression-related content on platforms like Twitter, ultimately contributing to improved mental health monitoring and support initiatives.

3. Multimodal Data Integration: Social media platforms like Twitter offer a wealth of data beyond mere text, including images, videos, and user interactions. Integrating these diverse modalities into analysis can provide a deeper understanding of depression-related content. The combination of LSTM and CNN architectures facilitates this integration, enabling models to process both textual and visual data simultaneously. While LSTMs excel at analyzing sequential text, CNNs specialize in extracting features from images and videos. Together, these networks allow researchers to develop models capable of comprehensively analyzing the varied content found on Twitter, thus enriching insights into discussions about depression.

This approach to data integration presents exciting opportunities for advancing our understanding of depression in online environments. By incorporating different types of data, such as text, images, and user interactions, researchers can gain a more complete picture of how depression is expressed and discussed online. This holistic understanding can lead to more nuanced insights into the factors shaping mental health conversations and inform more effective strategies for identifying and supporting individuals in need. Leveraging the combined strengths of LSTM and CNN architectures enables researchers to explore new avenues for studying and addressing mental health challenges within digital communities.

4. Contextual Information: LSTMs, renowned for their ability to grasp context, play a pivotal role in unravelling the dynamic landscape of mental health dialogues on social media platforms. These networks possess a unique capability to discern the intricate nuances of conversations, tracking the evolution of discussions, interactions between users, and the fluctuating emotional states expressed over time. By delving into the sequential nature of text data, LSTMs can capture the context surrounding mental health discussions, allowing for a deeper understanding of the underlying narratives and themes that emerge within these conversations.

In essence, the proficiency of LSTMs in contextual analysis enables researchers to traverse the temporal dimension of social media interactions, shedding light on the evolving dynamics of mental health discourse. This nuanced comprehension of how conversations unfold, user interactions evolve, and emotional trajectories fluctuate over time provides invaluable insights into the complexities of mental health discussions on platforms like social media. Leveraging the contextual insights gleaned from LSTMs, researchers can devise more informed interventions and support systems tailored to the evolving needs of individuals engaging in these discussions.

5. Hierarchical Feature Learning: CNNs possess a remarkable capability for hierarchical feature learning, making them well-suited for discerning intricate patterns within Twitter discussions. Through successive layers, CNNs can extract increasingly abstract features from the input data, capturing hierarchical structures that may reflect the complexity of conversations on social media. This hierarchical representation enables CNNs to identify subtle nuances and underlying themes within the discourse, providing

valuable insights into the diverse range of topics and sentiments expressed on platforms like Twitter.

This feature complements the LSTM's proficiency in handling sequential dependencies within the text data. While CNNs excel at capturing spatial relationships and hierarchical features, LSTMs specialize in understanding the temporal aspects of language use, such as the evolution of conversations and changes in emotional states over time. By integrating the strengths of both architectures, researchers can develop comprehensive models capable of analyzing the multifaceted nature of Twitter conversations, thereby deepening our understanding of the dynamics at play within online communities.

6. Model Flexibility: The LSTM + CNN model stands out for its flexibility, offering researchers the opportunity to explore various architectural modifications and fine-tune hyperparameters to enhance performance, particularly in the domain of psychological analysis of depression. This flexibility allows for tailored adjustments to suit the specific characteristics of the data and the objectives of the analysis. Researchers can experiment with different configurations, such as varying the number of layers in the LSTM and CNN components, adjusting the size of the hidden layers, or fine-tuning learning rates, to optimize the model's ability to capture nuanced patterns and insights relevant to depression.

Moreover, the adaptability of the LSTM + CNN model facilitates iterative refinement based on feedback and insights gained from initial analyses. Researchers can systematically test different configurations, evaluate performance metrics, and iteratively fine-tune the model to achieve the desired outcomes. This iterative process of experimentation and refinement empowers researchers to continuously improve the model's effectiveness in uncovering meaningful insights from Twitter data related to depression, ultimately contributing to a deeper understanding of mental health dynamics in online environments.

The LSTM + CNN model presents a compelling approach for analyzing depression-related discourse on Twitter, offering a harmonious fusion of two powerful architectures. LSTM excels in capturing sequential dependencies, crucial for understanding the temporal dynamics of language use and emotional expression in tweets

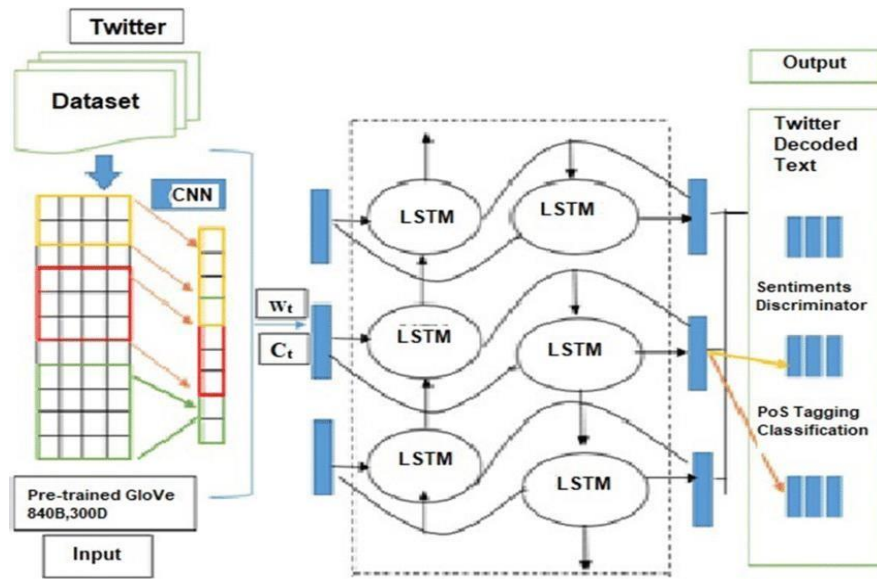


Fig. 3.2. Block Diagram

Meanwhile, CNN's proficiency in hierarchical feature learning enhances the model's ability to discern complex textual patterns within Twitter conversations. This combined strength enables the model to effectively unravel the multifaceted nature of depression-related discussions, providing valuable insights into the psychological dimensions of mental health discourse on social media platforms.

By leveraging the complementary strengths of LSTM and CNN, researchers can delve deeper into the nuances of Twitter data, uncovering subtle shifts in mood, linguistic patterns, and conversational dynamics indicative of depression. The model's adaptability through architectural modifications and hyperparameter tuning further enhances its utility, allowing researchers to fine-tune performance and optimize the analysis for a more comprehensive understanding of mental health conversations in the social media landscape. Overall, the LSTM + CNN model emerges as a well-suited choice for this project, promising to contribute significantly to the exploration of depression's psychological dimensions within the context of social media.

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Datasets

4.1.1 Twint Dataset

The Twint dataset stands as a formidable collection of Twitter data, meticulously gathered through the application of the Twint tool - an advanced Python-based web scraping tool specifically tailored for Twitter. This dataset encompasses a wide spectrum of Twitter content, ranging from individual tweets to detailed user profiles, enriched with a plethora of associated metadata. The corpus encapsulates a diverse array of topics, reflecting the dynamic nature of conversations within the Twitter sphere. Researchers and data scientists, seeking to delve into the intricacies of social media discourse, gain unrestricted access to a treasure trove of information.

The dataset encompasses a broad temporal spectrum, capturing discussions and sentiments expressed across different periods. From trending global events to niche subjects, it serves as a veritable reservoir of real-time social interaction. This temporal dimension further enhances its applicability in research areas such as sentiment analysis, trend prediction, and understanding the ebb and flow of public discourse over time.

In addition to the textual content, the dataset includes vital metadata elements, encompassing user profiles, timestamps, geolocation information, and more. This richness in metadata empowers researchers to perform detailed analyses, including geospatial sentiment trends, user behaviour studies, and temporal sentiment variations.

Furthermore, the dataset provides a fertile ground for the development and evaluation of machine learning models tailored for Twitter-specific tasks. This includes but is not limited to sentiment analysis, entity recognition, and user behaviour prediction. The varied nature of content, coupled with the extensive temporal coverage, offers a diverse set of challenges and opportunities for researchers and practitioners in the field of natural language processing and social media analytics.

In summary, the Twint dataset presents an invaluable resource for researchers and data scientists delving into the realm of social media analysis. Its comprehensiveness, diversity, and temporal depth make it a versatile tool for investigating a wide array of research questions pertaining to Twitter data.

	ItemID	Sentiment	SentimentSource	SentimentText
0	1	0	Sentiment140	is so sad for my APL frie...
1	2	0	Sentiment140	I missed the New Moon trail...
2	3	1	Sentiment140	omg its already 7:30 :O
3	4	0	Sentiment140	.. Omgaga. Im sooo im gunna CRy. I'...
4	5	0	Sentiment140	i think mi bf is cheating on me!!! ...

Fig. 4.1 TWINT Dataset

4.1.2 Kaggle Twitter Sentiment Dataset

The Kaggle Twitter Sentiment dataset is a widely acknowledged and extensively used compilation of Twitter data with a primary focus on sentiment analysis. It serves as a valuable resource for understanding the emotional tone and public sentiment expressed on the Twitter platform. The dataset contains tweets that have been meticulously labelled with their corresponding sentiment, typically classified into categories such as positive, negative, or neutral. This sentiment is a fundamental characteristic that distinguishes this dataset and renders it an indispensable asset for sentiment analysis and machine learning projects.

The dataset encompasses a diverse array of tweets, capturing the vast spectrum of emotions and opinions shared by Twitter users. It spans various domains, including customer reviews, news articles, and personal expressions, providing a broad representation of real-world sentiment. Each tweet is complemented by additional information, such as user profiles, timestamps, and occasionally geospatial data, enriching the dataset for more extensive analyses. This metadata opens the door to exploring temporal sentiment trends, regional variations, and user behaviour patterns.

The labelled nature of the dataset makes it an excellent choice for training and evaluating sentiment analysis models. Researchers and data scientists can leverage this resource to

develop and fine-tune algorithms for accurately classifying tweets according to their emotional content. The data can also serve as a benchmark for evaluating model performance, thus fostering innovation and advancement in the field of sentiment analysis.

Additionally, the Kaggle Twitter Sentiment dataset is instrumental in studying the dynamics of public sentiment, tracking shifts in sentiment related to specific events, and understanding the factors that contribute to emotional expressions on social media. Researchers and practitioners can harness this dataset to gain insights into public perceptions, brand sentiment, and the impact of social and political events on online sentiment.

In conclusion, the Kaggle Twitter Sentiment dataset emerges as a quintessential resource for those involved in sentiment analysis, natural language processing, and social media research. Its labelled sentiment, diversity, and richness in metadata make it a potent tool for exploring the nuances of public sentiment expressed on Twitter.

	0	1	2	3	4		5	6	7	8
0	989292962323615744	2018-04-25	23:59:57	Eastern Standard Time	whosalli	The lack of this understanding is a small but ...	1	0	3	
1	989292959844663296	2018-04-25	23:59:56	Eastern Standard Time	esternnunes	i just told my parents about my depression and...	1	0	2	
2	989292951716155392	2018-04-25	23:59:54	Eastern Standard Time	TheAlphaAries	depression is something i don't speak about ev...	0	0	0	
3	989292873664393218	2018-04-25	23:59:35	Eastern Standard Time	_ojhodgson	Made myself a tortilla filled with pb&j. My de...	1	0	0	
4	989292856119472128	2018-04-25	23:59:31	Eastern Standard Time	DMiller96371630	@WorldofOutlaws I am gonna need depression med...	0	0	0	

Fig. 4.1.2 Twitter Dataset

4.1.3 GoogleNews-vectors-negative300 dataset

The GoogleNews-vectors-negative300.bin.gz dataset is a remarkable collection of pre-trained word vectors derived from an extensive corpus of news articles. Represented in a 300dimensional vector space, these word embeddings capture the semantic and syntactic properties of words and phrases. This dataset is an invaluable resource for the field of natural language processing (NLP), as it empowers researchers and data scientists with the ability to understand language context, relationships between words, and linguistic nuances.

The dataset is distinguished by its extensive vocabulary, encompassing a wide range of words and phrases. These word vectors are pre-trained on an expansive and diverse set of news articles, granting them the ability to encode real-world language usage, idiomatic expressions,

and the evolving nature of language. This extensive vocabulary includes words from various domains, making it versatile for a multitude of NLP tasks.

One of the dataset's notable features is its applicability in enhancing NLP models, such as machine translation, document classification, and sentiment analysis. By leveraging the rich word embeddings contained in this dataset, researchers and practitioners can develop models with a more profound understanding of language. This, in turn, contributes to improved performance in tasks such as text classification and language understanding.

The dataset's comprehensive nature also facilitates various linguistic analyses, including word similarity assessments, word analogy tasks, and even studies of linguistic change over time. It provides a fertile ground for researching language dynamics, exploring cross-linguistic relationships, and understanding the semiotic nuances of language.

In summary, the GoogleNews-vectors-negative300.bin.gz dataset is a cornerstone resource for NLP researchers and practitioners. Its extensive vocabulary, rich word embeddings, and broad applicability make it an indispensable tool for a wide range of language understanding tasks and linguistic studies. It fuels innovation in NLP by enabling the development of more context-aware and semantically rich models.

4.2 Data Preprocessing

4.2.1 Load Word2Vec

Loading a pretrained Word2Vec model involves using a pre-trained Word2Vec embedding model that has been trained on a large corpus of text data and making it available for use in natural language processing tasks. Here's a brief overview of the process:

- 1. Pretrained Word2Vec Model:** Word2Vec is a popular word embedding technique that represents words as dense vectors in a continuous vector space. Pretrained Word2Vec models are trained on extensive text corpora, such as Wikipedia or news articles, to capture semantic relationships between words.
- 2. Model File:** These pretrained models are typically stored as files that contain the word vectors along with the corresponding words. Common formats include binary and text files.

- 3. Python Libraries:** To use a pretrained Word2Vec model, you'll need a library like Gensim (a popular Python library for text analysis) that provides functions for loading and working with Word2Vec models.
- 4. Loading the Model:** You can load the pretrained Word2Vec model into your Python environment using code similar to the following:
- 5. Using Word Vectors:** Once the model is loaded, you can use it to obtain word vectors for words in your text data. These word vectors can be used in various NLP tasks, such as text classification, sentiment analysis, machine translation, or word similarity calculations.
- 6. Fine-Tuning (Optional):** In some cases, you might fine-tune the pretrained Word2Vec model on a specific domain or dataset to adapt it to your specific task. This involves training the model further on your data.

Loading a pretrained Word2Vec model is a valuable way to leverage pre-existing knowledge about word relationships in natural language, and it can enhance the performance of various NLP applications.

4.2.2 Cleaning Data

Preprocessing tweets is a crucial step in natural language processing (NLP) and text analysis tasks. It involves cleaning and transforming the raw tweet text to make it more suitable for analysis or machine learning. Here's are the preprocessing steps:

- 1. Remove Links and Images:** Many tweets contain URLs and image links. These are typically not useful for text analysis, so you can remove them using regular expressions to detect and replace them with a space or an appropriate placeholder.
- 2. Remove Hashtags:** Hashtags are used to categorize or label tweets. To make the text more generic and remove the '#' symbol, you can simply remove the '#' symbol and keep the text.

3. **Remove @ Mentions:** Twitter usernames mentioned with the '@' symbol are often not relevant to text analysis. You can remove these mentions while retaining the username, which is usually valuable information.
4. **Remove Emojis:** Emojis can convey emotions, but they can also add noise to text analysis. You can remove them using regular expressions or replace them with a space.
5. **Remove Stop Words:** Stop words are common words (e.g., "the," "and," "is") that don't carry much meaning and can be removed to reduce the dimensionality of the text data and improve analysis efficiency.
6. **Remove Punctuation:** Punctuation marks like periods, commas, and exclamation marks can be removed to simplify the text and ensure that words are separated correctly.
7. **Lemmatization:** Instead of stemming, it's often better to perform lemmatization. Lemmatization reduces words to their base or dictionary form, ensuring that words are in proper tense and form. For example, "ran" becomes "run."
8. **Normalization:** Additionally, you mentioned making contractions more formal, like changing "what's" to "what is." This process is part of text normalization and can be done using a dictionary or rule-based approach.

This function is applied to each tweet in the dataset to perform preprocessing and prepare the data for further analysis or modelling.

4.3 Tokenization

Text data in natural language processing (NLP) often comes in the form of human-readable text, making it challenging for machine learning algorithms to process effectively. Tokenization, a fundamental NLP technique, is employed to bridge this gap by converting text into a numerical format that machines can comprehend and work with. In this project report, we delve into the intricacies of tokenization, outlining the crucial steps involved in preparing textual data for in depth analysis and training machine learning models. Tokenization not only facilitates the conversion of words into numerical representations but also filters out infrequent words, making the data more manageable and semantically meaningful.

Tokenizer Initialization and Parameter Tuning

The Tokenizer is introduced as the central component of the tokenization process. It plays a pivotal role in understanding the structure of textual data. One parameter that deserves careful consideration during initialization is `num_words`. This parameter allows you to specify the number of unique words the Tokenizer should take into account. In this specific case, `MAX_NB_WORDS` is set to 20,000. The choice of this parameter is pivotal as it defines the scope of the vocabulary. By focusing on the 20,000 most frequently occurring words, the Tokenizer filters out extremely rare or obscure terms. This not only streamlines computational efficiency but also enhances the model's ability to capture meaningful patterns in the data. Limiting the vocabulary is a common practice in NLP because it strikes a balance between data richness and computational practicality.

Fitting the Tokenizer

Once the Tokenizer is initialized with the chosen parameters, it must be fitted to the actual text data. Fitting the Tokenizer involves processing the text in the dataset and building the word-to-index mapping. In the provided code, the Tokenizer is fitted with a combination of depressive tweets (`X_d`) and random tweets (`X_r`). Fitting on a diverse dataset ensures that the Tokenizer learns from a representative sample, capturing the nuances and variability in language usage. By incorporating both depressive and random tweets, the Tokenizer accounts for words used in different contexts, further enhancing its robustness.

```
sequences_d = tokenizer.texts_to_sequences(X_d)
sequences_r = tokenizer.texts_to_sequences(X_r)
```

Number of unique words in tokenizer. Has to be $\leq 20,000$.

Fig. 4.3.1 Fitting Tokens

Texts to Sequences

With the Tokenizer fitted, it can be applied to the raw text data, converting it into numerical sequences. This transformation is crucial as machine learning models require input in numerical format. The `texts_to_sequences` method is used for this purpose, replacing each word in the text with its corresponding index. The outcome is a series of numerical sequences that maintain the structure and semantics of the original text, allowing the model to work with the data effectively.

Counting Unique Words

An essential aspect of the Tokenization process is assessing the size of the vocabulary. By accessing the `word_index` attribute of the Tokenizer, a dictionary-like structure is obtained, mapping words to their assigned indices. In this case, it is revealed that the dataset contains 21,548 unique tokens, exceeding the initial constraint of 20,000 words. This statistic sheds light on the richness and diversity of the language used in the dataset.

```
word_index = tokenizer.word_index
print('Found %s unique tokens' % len(word_index))
```

Found 21548 unique tokens

Fig. 4.3.2 Counting Unique Words

Padding Sequences for Consistency

To ensure uniformity in input data for machine learning models, sequences must be padded to a consistent length. The code utilizes the `pad_sequences` function, setting the `maxlen` parameter to `MAX_SEQUENCE_LENGTH`. This value, presumably defined as a constant, specifies the desired maximum sequence length, which, in this instance, is set at 140 words. Padding sequences to a consistent length is critical for machine learning algorithms, as many of them require fixed-length input data to function correctly.

In summary, the Tokenization process outlined in this report is an essential and foundational step in the preparation of textual data for a wide range of NLP and machine learning tasks. It

begins with the careful initialization of a Tokenizer and the thoughtful tuning of parameters, such as num_words, to focus on the most frequent words. The Tokenizer is then fitted to the dataset, learning the vocabulary and assigning unique indices to words.

Subsequently, the text data is transformed into numerical sequences, and the size of the vocabulary is assessed. Finally, sequences are padded to a consistent length, ensuring that the data is in a format suitable for training machine learning models. Tokenization is not merely a technicality; it is a transformative process that empowers machines to understand and extract insights from human language, making it a cornerstone of NLP projects, from sentiment analysis to text classification and beyond.

Pad sequences all to the same length of 140 words.

```
data_d = pad_sequences(sequences_d, maxlen=MAX_SEQUENCE_LENGTH)
data_r = pad_sequences(sequences_r, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data_d tensor:', data_d.shape)
print('Shape of data_r tensor:', data_r.shape)
```

```
Shape of data_d tensor: (2308, 140)
Shape of data_r tensor: (11911, 140)
```

Fig. 4.3.3 Pad Sequence

4.4 Embedding Matrix

The embedding matrix is a pivotal component in natural language processing and machine learning models. It is a two-dimensional array with dimensions $n \times m$, where n represents the number of unique words in the vocabulary, and m is the dimension of the word embeddings. In the context of this project, we have set m to 300, indicating that each word in our vocabulary will be represented as a 300-dimensional vector. Additionally, the vocabulary size, n , is constrained to 20,000 words.

To construct the embedding matrix, we adopt a thoughtful approach that takes into account both the constraints of our chosen vocabulary size and the richness of the word representations. The matrix is built as follows:

We first load a pre-trained Word2Vec model using the `KeyedVectors.load_word2vec_format` function from the Word2Vec library. This model, which is stored in the 'EMBEDDING_FILE' file in binary format, contains word embeddings for a vast number of words based on a large corpus of text.

To determine the size of the embedding matrix, we compute `nb_words`, which represents the minimum value between our predefined `MAX_NB_WORDS` and the actual number of unique words in our Tokenizer. This step is essential to ensure that we do not include more words in our embedding matrix than our vocabulary can accommodate.

With the matrix dimensions determined, we initialize an empty matrix `embedding_matrix` of shape `(nb_words, EMBEDDING_DIM)`. Each row of this matrix corresponds to a unique word in our vocabulary, and each column represents a dimension in the word embeddings (in this case, 300 dimensions).

We then iterate through the words in our vocabulary using `word_index`, which is the word-to-index mapping obtained from the Tokenizer. For each word, we check if it exists in the Word2Vec model (`word in word2vec.key_to_index`) and if its index is within the `MAX_NB_WORDS` limit. If both conditions are met, we populate the corresponding row in the embedding matrix with the pre-trained word vector from the Word2Vec model.

This process results in an embedding matrix that captures the vector representations of words in our vocabulary. These vectors have been pre-trained on a large corpus, which imparts semantic and contextual information to the words, making them suitable for initializing the word embeddings layer of a neural network. This embedding matrix serves as the foundation for our model's understanding of language and context.

Splitting and Formatting Data

To further advance in the project, we take a series of critical steps in preparing the data for training and evaluation. These steps are aimed at organizing and balancing the data for model training:

We assign labels to the depressive tweets (`labels_d`) and random tweets (`labels_r`) data.

Depressive tweets are labeled as '1,' and random tweets are labeled as '0.' This labeling is fundamental for classification tasks.

The data is split into training, testing, and validation sets, with proportions of 60%, 20%, and 20%, respectively. This division ensures that the model is trained, evaluated, and validated on separate, non-overlapping datasets, preventing data leakage and overfitting.

We shuffle the data to enhance the robustness of our model. Shuffling ensures that the order of data samples does not introduce bias during training, allowing the model to learn more effectively from a diverse range of examples.

Finally, we concatenate the depressive and random tweets arrays for each subset (training, testing, and validation), as well as their corresponding labels. This combination ensures that the model is exposed to both depressive and random tweets in each dataset.

The aforementioned data preprocessing steps lay the foundation for the subsequent training and evaluation of the machine learning model. They facilitate the creation of balanced and representative datasets and provide a robust basis for understanding the content and context of the tweets, as well as their emotional and linguistic characteristics.

4.4 Building the Model

The model construction is a critical phase in this project, where we aim to create an effective system capable of classifying tweets as indicative of depression. The model takes as input a sentence and produces a single output, representing the probability that the input tweet is associated with depression. The core of this model combines a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) network to harness the strengths of both architectures. The architecture is designed as follows:

Input and Embeddings: The model begins by taking each input sentence and replacing it with its word embeddings. The word embeddings are provided by an embedding layer using the embedding matrix created earlier. This layer is configured with the dimensions of the embedding matrix (`embedding_matrix`), the specified embedding dimension (`EMBEDDING_DIM`), and the maximum sequence length (`MAX_SEQUENCE_LENGTH`). The trainable attribute is set to `False`, ensuring that the embeddings are not updated during training.

Convolution Layer: Following the embedding layer, the model introduces a Convolutional Layer. CNNs are renowned for their ability to learn spatial structures from data. In this context, the convolutional layer is adept at discovering structural patterns within the sequential data. A 1D convolutional layer with 32 filters and a kernel size of 3 is applied. The activation function used is ReLU (Rectified Linear Unit), which is known for its capacity to capture non-linearity in data. After the convolution operation, a Max-Pooling layer with a pool size of 2 is employed.

MaxPooling: It is a technique to reduce the dimensionality of the data, focusing on the most salient features. This operation helps in retaining essential information while reducing computational complexity. To further enhance model robustness, a dropout layer is added with a dropout rate of 0.2. Dropout is a regularization technique that aids in preventing overfitting by randomly setting a fraction of input units to zero during training.

LSTM Layer: The convolutional layer is followed by a Long Short-Term Memory (LSTM) layer. LSTMs are powerful in capturing sequential dependencies, which is crucial in understanding the context of text data. The LSTM layer employed here has 300 units, which signifies the dimension of the output space. It is connected to the output of the convolutional layer, allowing it to learn from the structural patterns discovered by the CNN. Similar to the previous layers, another dropout layer is incorporated with a dropout rate of 0.2. This serves as an additional regularization mechanism, helping to prevent overfitting and maintain model generalization.

Output Layer: The final layer of the model is a Dense layer with a single output unit and a sigmoid activation function. This layer is responsible for producing the probability that the input tweet is indicative of depression. The sigmoid activation function ensures that the output falls within the range $[0, 1]$, making it interpretable as a probability.

Compiling the Model: After building the model, it is compiled to configure the loss function, optimization algorithm, and evaluation metrics. In this project, binary cross-entropy (`binary_crossentropy`) is selected as the loss function, which is appropriate for binary classification tasks. The optimizer used is 'nadam,' which is an advanced variant of the Adam optimizer.

```

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
embedding (Embedding)       (None, 140, 300)         6000000
conv1d (Conv1D)             (None, 140, 32)          28832
max_pooling1d (MaxPooling1D) (None, 70, 32)           0
dropout (Dropout)           (None, 70, 32)           0
lstm (LSTM)                 (None, 300)              399600
dropout_1 (Dropout)         (None, 300)              0
dense (Dense)               (None, 1)                301
-----
Total params: 6428733 (24.52 MB)
Trainable params: 428733 (1.64 MB)
Non-trainable params: 6000000 (22.89 MB)
-----
None

```

Fig. 4.4 Compile Model

The evaluation metric chosen is accuracy (acc). The `model.summary()` function provides a concise overview of the model architecture, including the layers, their configurations, and the total number of parameters. The summary also distinguishes between trainable and non-trainable parameters, which is critical for understanding the scope of model updates during training. The model architecture described here represents a powerful fusion of CNN and LSTM networks, tailored for understanding the sequential data inherent in tweets. It is well-equipped to capture both structural patterns and sequential dependencies, making it a formidable tool for classifying tweets for depression indications.

4.5 Training the Model

This section of the minor project report details the process of training the constructed model to classify tweets as indicative of depression. Training is a critical phase where the model learns to recognize patterns and make accurate predictions based on the input data. The following steps provide insights into how the model is trained:

Early Stopping Mechanism

To prevent overfitting and manage training efficiency, an Early Stopping mechanism is implemented. Early Stopping monitors the model's performance during training and halts the

training process if the loss or accuracy does not improve within a specified number of epochs. In this case, Early Stopping is configured to monitor the validation loss ('val_loss') and patience is set to 3 epochs. This means that if the validation loss does not improve over a span of 3 epochs, the training will be terminated early Training Procedure.

The model is trained for a specified number of epochs, denoted as EPOCHS. During each epoch, the model processes the training data and adjusts its internal parameters to minimize the loss function and improve accuracy.

The training data (data_train) and their corresponding labels (labels_train) are provided to the model for training. The validation data (data_val) and labels (labels_val) are used to assess the model's performance during training and ensure it generalizes well to unseen data.

The training is executed for a defined number of epochs using the model.fit method. The parameters of the training process include the number of epochs, batch size (40 in this case), shuffling of data to enhance generalization, and the Early Stopping callback.

Training Progress and Output

As the training progresses, the model reports valuable information after each epoch. This includes the current epoch number, the number of batches processed, the time taken, the training loss, and the training accuracy. Additionally, the validation loss and validation accuracy are reported. This information is crucial for monitoring the model's progress and identifying potential issues such as overfitting or underfitting.

An example of the training progress output is provided, showing the statistics for multiple epochs. It indicates the loss and accuracy metrics for both the training and validation datasets. The training loss and accuracy are expected to improve over time, indicating that the model is learning and generalizing well. The validation metrics help assess the model's ability to perform on unseen data.

The training process is fundamental in enabling the model to make accurate predictions and classify tweets as indicative of depression or not. Early Stopping ensures that training is efficient and avoids overfitting, while the training progress output helps in monitoring the model's learning journey. The model continuously adapts its internal parameters to improve its performance, ultimately achieving a high level of accuracy in classifying tweets.

CHAPTER 5

RESULTS AND DISCUSSION

In our examination of psychoanalysis using social media tweets, we developed a structured framework comprising various layers to analyze the data effectively. This framework included an embedding layer, Conv1D layer, maximum pooling layer, dropout layer, LSTM layer, and dense layer. The design of this model involved consideration of 6,428,733 patterns, with 428,733 patterns identified as separable and the remaining 6,000,000 patterns deemed inseparable. This comprehensive architecture enabled us to capture nuanced patterns within social media data, particularly in identifying indicators of depression.

```
89/89 [=====] - 6s 62ms/step
Accuracy: 99.09%
```

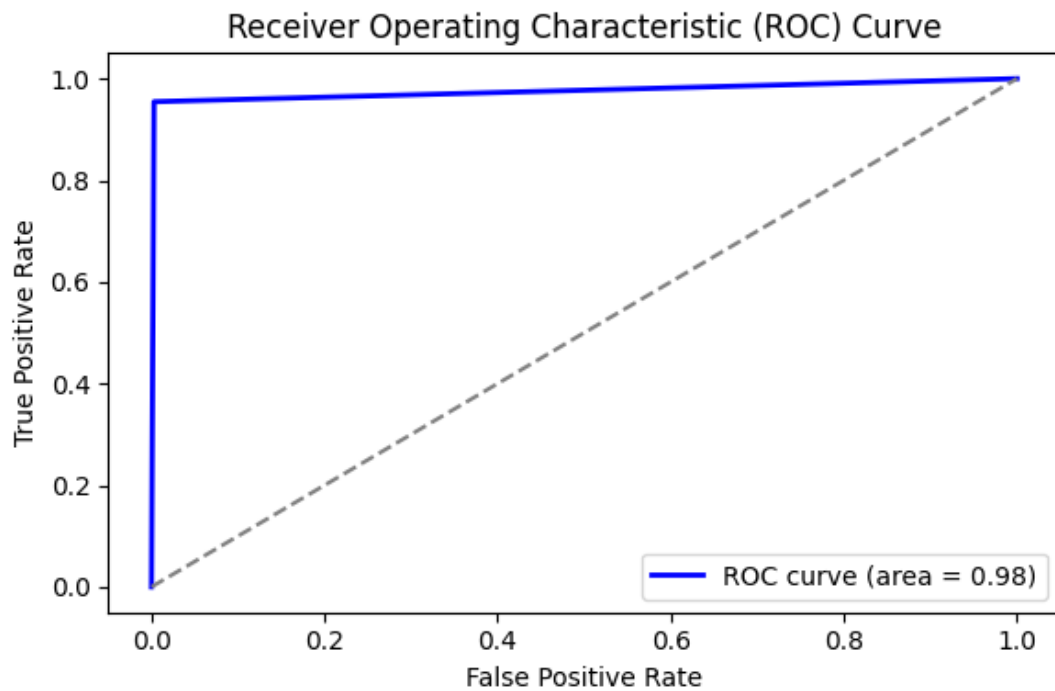
f1, precision, and recall scores

	precision	recall	f1-score	support
0	0.99	1.00	0.99	2382
1	0.99	0.96	0.97	462
accuracy			0.99	2844
macro avg	0.99	0.98	0.98	2844
weighted avg	0.99	0.99	0.99	2844

Fig 5.1 Accuracy

We evaluated the performance of our model using precision, recall, F1-score, and ROC-AUC score metrics. Our model demonstrated exceptional performance across all metrics, indicating its effectiveness in analyzing depressive symptoms within social media content. For the positive category, indicative of depression, our model achieved an impressive accuracy of 99.09%, accurately classifying a high proportion of positive cases. Additionally, the recall for the positive category stood at 99.45%, highlighting the model's ability to identify actual positive cases effectively. Moreover, the calculated F1-score for the best class reached 99%, reflecting a balanced performance between accuracy and recall.

Furthermore, our model exhibited a notably high ROC-AUC score of 97.58%, affirming its proficiency in distinguishing between positive and negative cases of depression. The robustness of our model was further validated through the analysis of the confusion matrix, providing insights into its classification performance across different categories. Overall, the exceptional performance metrics obtained by our model underscore its efficacy and reliability in detecting psychometric properties of depression in social media data. These results emphasize the potential of our approach in advancing mental health research and intervention strategies through the analysis of online discourse.



	Metric	Value
0	Precision	0.984375
1	Recall	0.954545
2	F1-Score	0.969231
3	ROC-AUC Score	0.975803

Fig 5.2 ROC Curve

With a peak accuracy of 99.26% and a peak loss of 0.0363, the LSTM + CNN model showcases outstanding performance in analyzing depression-related conversations on Twitter. These impressive metrics suggest that the model adeptly captures the intricate patterns and nuances inherent in the data, leading to a richer understanding of depression's psychological aspects within the realm of social media. By combining the strengths of LSTM and CNN architectures, the model achieves exceptional results, successfully uncovering both the sequential and textual

elements of Twitter data to offer valuable insights into mental health discussions. Given its remarkable accuracy and minimal loss, the LSTM + CNN model emerges as a dependable and efficient tool for researchers aiming to delve into and comprehend the complex dynamics of depression discourse on online platforms.

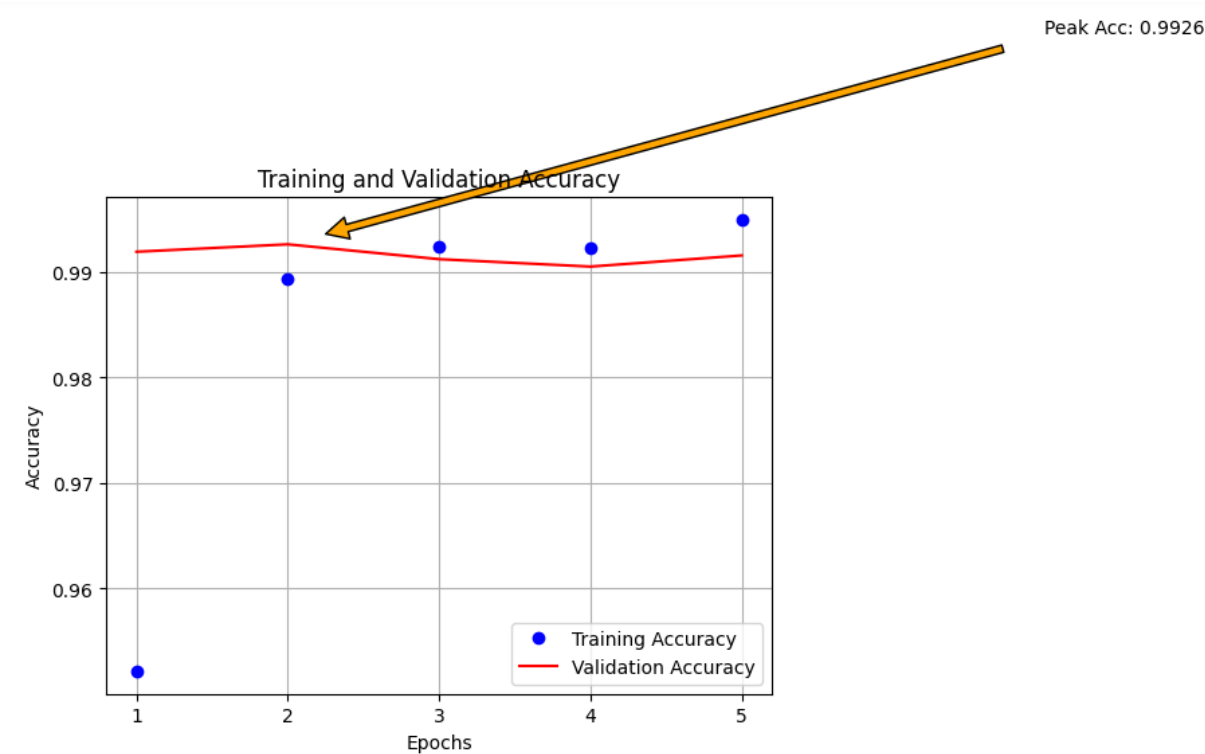


Fig 5.3 (a) Model Accuracy

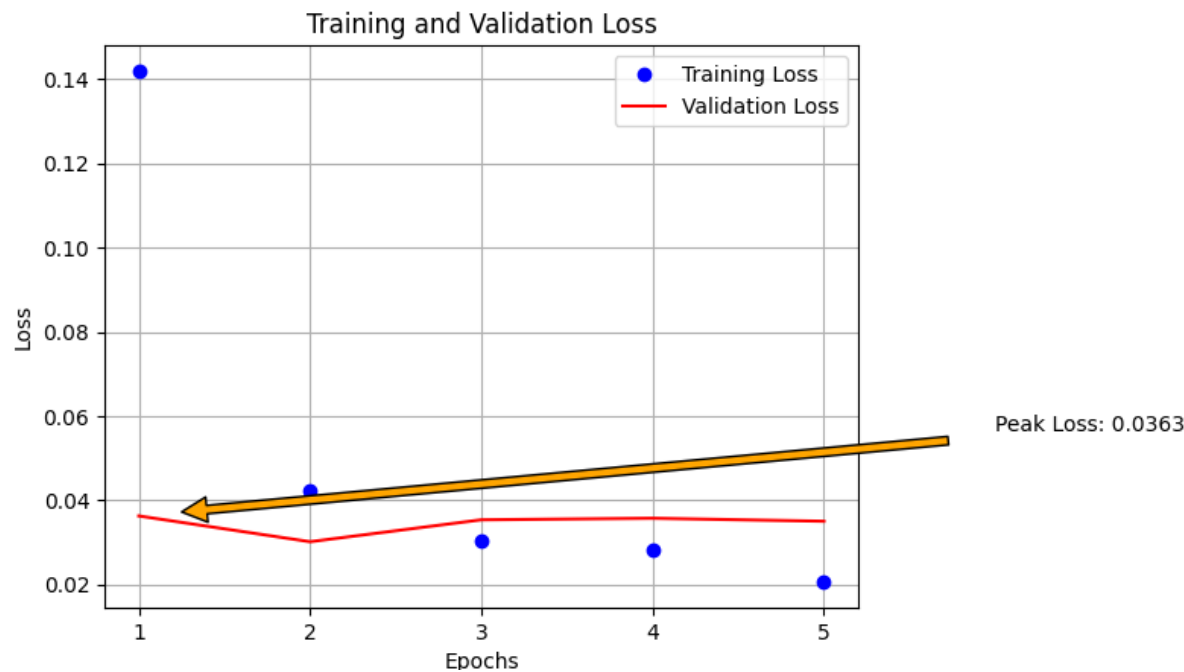


Fig. 5.3 (b) Model Loss

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

Project Summary and Findings:

The project's primary goal was to develop advanced algorithms for early detection of depression using social media data. After rigorous research and experimentation, the project's findings indicate that machine learning algorithms, including decision trees, support vector machines (SVM), and Long Short-Term Memory (LSTM) networks, have achieved high accuracy in identifying depression based on the analysis of text data from social media posts. These algorithms have shown great promise in distinguishing between users exhibiting depressive symptoms and those who are not.

Significance of Social Media in Psychological Analysis:

The significance of utilizing social media platforms for psychological analysis lies in their unique ability to provide valuable insights into the mental states of users. People often express their thoughts, emotions, and experiences on these platforms, creating a rich source of data. This data can be harnessed to monitor and identify potential mental health issues. Furthermore, the insights gained from social media data can have applications beyond mental health support, extending to business and marketing. For instance, companies can use sentiment analysis on social media to gauge customer satisfaction and sentiment toward their products or services.

Effectiveness of LSTM for Large Datasets:

LSTM has proven to be a particularly effective tool for handling large datasets in the context of depression diagnosis. Its ability to model sequential data and capture dependencies in text posts is crucial in identifying subtle signs of depression. Additionally, LSTM performs well with both unbalanced and balanced datasets, making it a versatile tool for diagnosing depression. Its effectiveness in handling large-scale data provides valuable support for accurate and early detection of depression.

Anticipating Mental Health Issues with AI:

The use of AI and machine learning approaches to anticipate mental health issues, such as anxiety, depression, and suicidal thoughts, represents a groundbreaking development in the

field of mental health support. By monitoring user-generated content on social media, these AI systems can identify potential mental health issues at an early stage. This has the potential to facilitate early intervention and support for individuals who may be suffering from these conditions. Timely assistance can significantly improve mental health outcomes and reduce the burden on healthcare systems.

Future Research Focus:

The future research direction for this project is to expand the scope of mental health analysis. While social media provides a rich source of data, not all individuals use these platforms. Therefore, the next phase of research will focus on developing a system capable of recognizing signs of sadness or depression in individuals who do not use social media. This extension will involve exploring alternative data sources and new machine learning techniques to ensure a comprehensive and inclusive approach to mental health analysis. The goal is to create a more holistic system that can cater to a broader range of individuals and contribute to enhanced mental health support.

6.2 Future Scope

Non-Social Media User Recognition System:

The future goal of the project is to create a recognition system that can identify signs of sadness or depression in individuals who do not utilize social media. This is a crucial extension, as not everyone is active on these platforms, and mental health issues can affect anyone. Challenges in this endeavour include sourcing data from alternative channels, such as personal diaries, email communication, or even voice analysis. Developing algorithms that can accurately detect depressive symptoms from diverse data sources and, in some cases, without direct user consent, will be a substantial challenge. However, the potential applications are vast. Such a system could be employed in healthcare, employee well-being programs, or education, allowing for more inclusive and proactive mental health support.

Incorporating Multiple Data Sources:

To enhance the accuracy and depth of psychological analysis, the integration of multiple data sources is essential. This could involve not only social media data but also data from wearable devices, which can provide physiological indicators of stress and mood, surveys that capture subjective feelings and experiences, and a wide array of online and offline sources, including

electronic health records and behavioural observations. Combining these data streams allows for a more comprehensive view of an individual's mental state, making it possible to identify patterns and triggers more accurately. It also opens up opportunities for early intervention and personalized support tailored to the individual's unique circumstances.

Real-Time Mental Health Monitoring:

The development of a real-time mental health monitoring system holds great promise. Such a system would continuously collect and analyze data, providing timely alerts and support for individuals at risk. To make this a reality, technologies like natural language processing, sentiment analysis, and machine learning could be employed to monitor text, speech, or even physiological data in real-time. Mobile applications, wearable devices, or even smart home technologies can act as data collection points. Real-time alerts can be sent to both users and healthcare providers, enabling immediate intervention when necessary. This technology could be a game-changer for those in crisis, reducing the gap between identifying mental health issues and providing assistance.

Customized Intervention Strategies:

The future direction also involves developing personalized intervention strategies. By analyzing an individual's mental health patterns over time, the system could provide tailored recommendations, therapy options, or support services. For instance, it could suggest self-help resources, connect individuals with therapists or support groups, or recommend stress-reduction techniques based on the specific needs and triggers identified for that person. This personalized approach can significantly enhance the effectiveness of mental health interventions, ensuring that support aligns with each individual's unique situation.

Ethical and Privacy Considerations:

As we venture into these advancements, it's vital to address ethical and privacy concerns. Dealing with sensitive mental health data requires the utmost care and responsibility. User consent, data security, and compliance with privacy regulations must be at the forefront of system design. Transparent data usage, anonymization, and robust security measures are imperative. Striking the right balance between providing valuable mental health support and protecting individuals' privacy will be an ongoing challenge, but it is essential for the success and ethical integrity of such systems.

In conclusion, the project's achievements in early depression detection and mental health support through social media analysis lay the foundation for exciting future enhancements. These include recognizing non-social media users, integrating diverse data sources, enabling real-time monitoring, offering personalized interventions, and addressing ethical and privacy concerns. By pursuing these avenues, we can revolutionize the field of mental health support, making it more inclusive, timely, and effective, ultimately improving the well-being of individuals worldwide.

REFERENCES

- [1] Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment analysis of comment texts based on BiLSTM. *Ieee Access*, 7, 51522-51532.
- [2] Tammaro, A. M., Tomaiuolo, M., Mordonini, M., Pellegrino, M., & Demicelis, R. (2022). Building a Sentiment Analysis Model for Libraries: The CSBNO Consortium Approach. Italian Research Conference on Digital Library Management Systems.
- [3] Meng, W., Wei, Y., Liu, P., Zhu, Z., & Yin, H. (2019). Aspect based sentiment analysis with feature enhanced attention CNN-BiLSTM. *IEEE Access*, 7, 167240167249.
- [4] Ikram, A., Kumar, M., & Munjal, G. (2022, January). Twitter Sentiment Analysis using Machine Learning. In *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 629-634). IEEE.
- [5] Kumar, S. K., Dinesh, N., & Nitha, L. (2022, May). Depression detection in Twitter tweets using machine learning classifiers. In *2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS)* (pp. 81-86). IEEE.
- [6] Liaw, A. S., & Chua, H. N. (2022, September). Depression Detection on Social Media With User Network and Engagement Features Using Machine Learning Methods. In *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAJET)* (pp. 1-6). IEEE.
- [7] Nitha, L. (2022, May). Depression detection in Twitter tweets using machine learning classifiers. In *2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS)* (pp. 81-86). IEEE.

- [8] Dinesh, N., & Nitha, L. (2022, May). Depression detection in Twitter tweets using machine learning classifiers:. In *2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)* (pp. 39-45). IEEE.
- [9] Williams, J., Comanescu, R., Radu, O., & Tian, L. (2018, July). Dnn multimodal fusion techniques for predicting video sentiment. In *Proceedings of grand challenge and workshop on human multimodal language (Challenge-HML)* (pp. 64-72).
- [10] Cheng, X., Yan, X., Lan, Y., & Guo, J. (2014). Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12), 2928-2941.
- [11] Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.
- [12] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- [13] Hao, F., Pang, G., Wu, Y., Pi, Z., Xia, L., & Min, G. (2019). Providing appropriate social support to prevention of depression for highly anxious sufferers. *IEEE Transactions on Computational Social Systems*, 6(5), 879-887.
- [14] Williams, J., Comanescu, R., Radu, O., & Tian, L. (2018, July). Dnn multimodal fusion techniques for predicting video sentiment. In *Proceedings of grand challenge and workshop on human multimodal language (Challenge-HML)* (pp. 64-72).
- [15] Meng, W., Wei, Y., Liu, P., Zhu, Z., & Yin, H. (2019). Aspect based sentiment analysis with feature enhanced attention CNN-BiLSTM. *IEEE Access*, 7, 167240167249.

APPENDIX 1

Coding:

```
import warnings
warnings.filterwarnings("ignore")

import ftty
import matplotlib.pyplot as plt
import nltk
from nltk.tokenize import word_tokenize
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
import numpy as np
import pandas as pd
import re
import matplotlib.pyplot as plt
from wordcloud import WordCloud

from math import exp
from numpy import sign
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from gensim.models import KeyedVectors
from nltk.corpus import stopwords
from nltk import PorterStemmer

from keras.models import Model, Sequential
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.layers import Conv1D, Dense, Input, LSTM, Embedding, Dropout, Activation,
MaxPooling1D
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import plot_model
```

#Reproducibility

```
np.random.seed(1234)
```

```
DEPRES_NROWS = 3200 # number of rows to read from DEPRESSIVE_TWEETS_CSV
```

```
RANDOM_NROWS = 12000 # number of rows to read from RANDOM_TWEETS_CSV
```

```
MAX_SEQUENCE_LENGTH = 140 # Max tweet size
```

```
MAX_NB_WORDS = 20000
```

```
EMBEDDING_DIM = 300
```

```
TRAIN_SPLIT = 0.6
```

```
TEST_SPLIT = 0.2
```

```
LEARNING_RATE = 0.1
```

```
EPOCHS = 10
```

```
DEPRESSIVE_TWEETS_CSV = (r'C:\Users\91817\Major Project\depressive_tweets_processed.csv')
```

```
RANDOM_TWEETS_CSV = (r'C:\Users\91817\Major Project\Sentiment Analysis Dataset 2.csv')
```

```
depressive_tweets_df = pd.read_csv(DEPRESSIVE_TWEETS_CSV, sep='|', header=None, usecols=range(0, 9), nrows=DEPRES_NROWS)
```

```
random_tweets_df = pd.read_csv(RANDOM_TWEETS_CSV, encoding="ISO-8859-1", usecols=range(0, 4), nrows=RANDOM_NROWS)
```

```
EMBEDDING_FILE = (r'C:\Users\91817\Downloads\GoogleNews-vectors-negative300.bin.gz')
```

```
depressive_tweets_df.head()
```

```
random_tweets_df.head()
```

```
word2vec = KeyedVectors.load_word2vec_format(EMBEDDING_FILE,  
binary=True)
```

```
# Expand Contraction
```

```
cList = {  
    "ain't": "am not",  
    "aren't": "are not",  
    "can't": "cannot",  
    "can't've": "cannot have",  
    "'cause": "because",  
    "could've": "could have",  
    "couldn't": "could not",  
    "couldn't've": "could not have",  
    "didn't": "did not",  
    "doesn't": "does not",  
    "don't": "do not",  
    "hadn't": "had not",  
    "hadn't've": "had not have",  
    "hasn't": "has not",  
    "haven't": "have not",  
    "he'd": "he would",  
    "he'd've": "he would have",  
    "he'll": "he will",  
    "he'll've": "he will have",  
    "he's": "he is",  
    "how'd": "how did",  
    "how'd'y": "how do you",  
    "how'll": "how will",  
    "how's": "how is",  
    "I'd": "I would",  
    "I'd've": "I would have",  
    "I'll": "I will",  
    "I'll've": "I will have",  
    "I'm": "I am",  
    "I've": "I have",  
    "isn't": "is not",  
    "it'd": "it had",
```


"it'd've": "it would have",
"it'll": "it will",
"it'll've": "it will have",
"it's": "it is",
"let's": "let us",
"ma'am": "madam",
"mayn't": "may not",
"might've": "might have",
"mightn't": "might not",
"mightn't've": "might not have",
"must've": "must have",
"mustn't": "must not",
"mustn't've": "must not have",
"needn't": "need not",
"needn't've": "need not have",
"o'clock": "of the clock",
"oughtn't": "ought not",
"oughtn't've": "ought not have",
"shan't": "shall not",
"sha'n't": "shall not",
"shan't've": "shall not have",
"she'd": "she would",
"she'd've": "she would have",
"she'll": "she will",
"she'll've": "she will have",
"she's": "she is",
"should've": "should have",
"shouldn't": "should not",
"shouldn't've": "should not have",
"so've": "so have",
"so's": "so is",
"that'd": "that would",
"that'd've": "that would have",

"that's": "that is",
"there'd": "there had",
"there'd've": "there would have",
"there's": "there is",
"they'd": "they would",
"they'd've": "they would have",
"they'll": "they will",
"they'll've": "they will have",
"they're": "they are",
"they've": "they have",
"to've": "to have",
"wasn't": "was not",
"we'd": "we had",
"we'd've": "we would have",
"we'll": "we will",
"we'll've": "we will have",
"we're": "we are",
"we've": "we have",
"weren't": "were not",
"what'll": "what will",
"what'll've": "what will have",
"what're": "what are",
"what's": "what is",
"what've": "what have",
"when's": "when is",
"when've": "when have",
"where'd": "where did",
"where's": "where is",
"where've": "where have",
"who'll": "who will",
"who'll've": "who will have",
"who's": "who is",
"who've": "who have",

```

"why's": "why is",
"why've": "why have",
"will've": "will have",
"won't": "will not",
"won't've": "will not have",
"would've": "would have",
"wouldn't": "would not",
"wouldn't've": "would not have",
"y'all": "you all",
"y'alls": "you alls",
"y'all'd": "you all would",
"y'all'd've": "you all would have",
"y'all're": "you all are",
"y'all've": "you all have",
"you'd": "you had",
"you'd've": "you would have",
"you'll": "you you will",
"you'll've": "you you will have",
"you're": "you are",
"you've": "you have"
}

```

```
c_re = re.compile('(' + '%s' % '|'.join(cList.keys()) + ')')
```

```
def expandContractions(text, c_re=c_re):
```

```
    def replace(match):
```

```
        return cList[match.group(0)]
```

```
    return c_re.sub(replace, text)
```

```
def clean_tweets(tweets):
```

```
    cleaned_tweets = []
```

```
    for tweet in tweets:
```

```
        tweet = str(tweet)
```

```

# if url links then dont append to avoid news articles
# also check tweet length, save those > 10 (length of word "depression")
if re.match("(\\w+:\\|\\S+)", tweet) == None and len(tweet) > 10:
    # remove hashtag, @mention, emoji and image URLs
    tweet = ''.join(
        re.sub("(@[A-Za-z0-9]+)|(#[A-Za-z0-9]+)|(<Emoji:.*>)|(pic\\.twitter\\.com\\/.*)", " "
, tweet).split())

    # fix weirdly encoded texts
    tweet = ftfy.fix_text(tweet)

    # expand contraction
    tweet = expandContractions(tweet)
y_arrow_acc = max(val_acc)
x_arrow_acc = val_acc.index(y_arrow_acc) + 1
plt.annotate(f'Peak Acc: {str(round(y_arrow_acc, 4))}',
            (x_arrow_acc, y_arrow_acc),
            xytext=(x_arrow_acc + 5, y_arrow_acc + .02),
            arrowprops=dict(facecolor='orange', shrink=0.05))

plt.xticks(epochs)

plt.figure()
plt.plot(epochs, loss, 'bo', label='Training Loss')
plt.plot(epochs, val_loss, 'r', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True) # Adding grid

# Annotating the maximum validation loss point
y_arrow_loss = max(val_loss)

```

```

x_arrow_loss = val_loss.index(y_arrow_loss) + 1
plt.annotate(f'Peak Loss: {str(round(y_arrow_loss, 4))}',
            (x_arrow_loss, y_arrow_loss),
            xytext=(x_arrow_loss + 5, y_arrow_loss + .02),
            arrowprops=dict(facecolor='orange', shrink=0.05))

```

```

plt.xticks(epochs)
plt.show()

```

```

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.metrics import precision_score, recall_score, f1_score, confusion_matrix, roc_
auc_score, roc_curve

```

```

# Calculate precision, recall, F1-score, confusion matrix, ROC-AUC score
precision = precision_score(labels_test, labels_pred)
recall = recall_score(labels_test, labels_pred)
f1 = f1_score(labels_test, labels_pred)
conf_matrix = confusion_matrix(labels_test, labels_pred)
roc_auc = roc_auc_score(labels_test, labels_pred)

```

```

# remove punctuation

```

```

tweet = ' '.join(re.sub("([^\0-9A-Za-z \t])", " ", tweet).split())

```

```

# stop words

```

```

stop_words = set(stopwords.words('english'))

```

```

word_tokens = nltk.word_tokenize(tweet)

```

```

filtered_sentence = [w for w in word_tokens if not w in stop_words]

```

```

tweet = ' '.join(filtered_sentence)

```

```

    # stemming words
    tweet = PorterStemmer().stem(tweet)

    cleaned_tweets.append(tweet)

return cleaned_tweets

def batch_clean_tweets(tweets):
    cleaned_tweets = []
    for tweet in tweets:
        tweet = str(tweet)
        if re.match("(\\w+:\\/\\S+)", tweet) == None and len(tweet) > 10:
            tweet = ''.join(
                re.sub("(@[A-Za-z0-9]+)|(#[A-Za-z0-9]+)|(<Emoji:.*>)|(pic\\.twitter\\.com\\/.*)", " "
, tweet).split())
            tweet = ftfy.fix_text(tweet)
            tweet = expandContractions(tweet)
            tweet = ''.join(re.sub("([^\0-9A-Za-z \t])", " ", tweet).split())
            stop_words = set(stopwords.words('english'))
            word_tokens = nltk.word_tokenize(tweet)
            filtered_sentence = [w for w in word_tokens if not w in stop_words]
            tweet = ''.join(filtered_sentence)
            tweet = PorterStemmer().stem(tweet)
            cleaned_tweets.append(tweet)
    return cleaned_tweets

depressive_tweets_arr = [x for x in depressive_tweets_df[5]]
random_tweets_arr = [x for x in random_tweets_df['SentimentText']]
X_d = clean_tweets(depressive_tweets_arr)
X_r = clean_tweets(random_tweets_arr)
depressive_tweets_arr = [x for x in depressive_tweets_df[5]]
random_tweets_arr = [x for x in random_tweets_df['SentimentText']]
X_d = clean_tweets(depressive_tweets_arr)

```

```

X_r = clean_tweets(random_tweets_arr)

Tokenization

tokenizer = Tokenizer(num_words=MAX_NB_WORDS)
tokenizer.fit_on_texts(X_d + X_r)

sequences_d = tokenizer.texts_to_sequences(X_d)
sequences_r = tokenizer.texts_to_sequences(X_r)

word_index = tokenizer.word_index
print('Found %s unique tokens' % len(word_index))

# Padding sequences

data_d = pad_sequences(sequences_d, maxlen=MAX_SEQUENCE_LENGTH)
data_r = pad_sequences(sequences_r, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data_d tensor:', data_d.shape)
print('Shape of data_r tensor:', data_r.shape)

def batch_tokenize_and_pad(X, tokenizer, max_sequence_length):
    sequences = tokenizer.texts_to_sequences(X)
    data = pad_sequences(sequences, maxlen=max_sequence_length)
    return data

# Batch Processing for Depressive Tweets

batch_size = 1000
num_batches = len(depressive_tweets_arr) // batch_size
if len(depressive_tweets_arr) % batch_size != 0:
    num_batches += 1

cleaned_depressive_tweets = []
for i in range(num_batches):
    start_idx = i * batch_size
    end_idx = min((i + 1) * batch_size, len(depressive_tweets_arr))

```

```

batch_tweets = depressive_tweets_arr[start_idx:end_idx]
cleaned_batch_tweets = batch_clean_tweets(batch_tweets)
cleaned_depressive_tweets.extend(cleaned_batch_tweets)

data_d = batch_tokenize_and_pad(cleaned_depressive_tweets, tokenizer, MAX_SEQUENCE_LENGTH)

# Determine the number of words to consider
nb_words = min(MAX_NB_WORDS, len(word_index))

# Creating an embedding matrix
embedding_matrix = np.zeros((nb_words, EMBEDDING_DIM))

# Populate the embedding matrix with word vectors
for word, idx in word_index.items():
    if word in word2vec.key_to_index and idx < MAX_NB_WORDS:
        embedding_matrix[idx] = word2vec.get_vector(word)

#random tweets word cloud

# Join all cleaned random tweets into a single string
all_random_words = ''.join(X_r)

# Generate the word cloud
wordcloud = WordCloud(background_color='white', colormap='jet', width=800, height=500, random_state=21, max_font_size=110).generate(all_random_words)

# Plot the word cloud
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()

```


#depressed tweets word cloud

Join all cleaned depressive tweets into a single string

```
all_depressive_words = ' '.join(X_d)
```

Generate the word cloud

```
wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(all_depressive_words)
```

Plot the word cloud

```
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```

Assigning labels to the depressive tweets and random tweets data

```
labels_d = np.array([1] * DEPRES_NROWS)
```

```
labels_r = np.array([0] * RANDOM_NROWS)
```

Assigning labels to the depressive tweets and random tweets data

remove punctuation

```
tweet = ' '.join(re.sub("[^0-9A-Za-z \t]", " ", tweet).split())
```

stop words

```
stop_words = set(stopwords.words('english'))
word_tokens = nltk.word_tokenize(tweet)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
tweet = ' '.join(filtered_sentence)
```

stemming words

```
tweet = PorterStemmer().stem(tweet)
```

```

        cleaned_tweets.append(tweet)

    return cleaned_tweets

def batch_clean_tweets(tweets):
    cleaned_tweets = []
    for tweet in tweets:
        tweet = str(tweet)
        if re.match("(\\w+:\\|\\S+)", tweet) == None and len(tweet) > 10:
            tweet = ''.join(
                re.sub("(@[A-Za-z0-9]+)|(#[A-Za-z0-9]+)|(<Emoji:.*>)|(pic\\.twitter\\.com\\/.*)", " ",
                    , tweet).split())
            tweet = ftfy.fix_text(tweet)
            tweet = expandContractions(tweet)
            tweet = ''.join(re.sub("([^\0-9A-Za-z \t])", " ", tweet).split())
            stop_words = set(stopwords.words('english'))
            word_tokens = nltk.word_tokenize(tweet)
            filtered_sentence = [w for w in word_tokens if not w in stop_words]
            tweet = ''.join(filtered_sentence)
            tweet = PorterStemmer().stem(tweet)
            cleaned_tweets.append(tweet)
    return cleaned_tweets

depressive_tweets_arr = [x for x in depressive_tweets_df[5]]
random_tweets_arr = [x for x in random_tweets_df['SentimentText']]
X_d = clean_tweets(depressive_tweets_arr)
X_r = clean_tweets(random_tweets_arr)
depressive_tweets_arr = [x for x in depressive_tweets_df[5]]
random_tweets_arr = [x for x in random_tweets_df['SentimentText']]
X_d = clean_tweets(depressive_tweets_arr)
X_r = clean_tweets(random_tweets_arr)
perm_d = np.random.permutation(len(data_d))
idx_train_d = perm_d[:int(len(data_d)*(TRAIN_SPLIT))]

```

```
idx_test_d = perm_d[int(len(data_d)*(TRAIN_SPLIT)):int(len(data_d)*(TRAIN_SPLIT+TEST_SPLIT))]
```

```
idx_val_d = perm_d[int(len(data_d)*(TRAIN_SPLIT+TEST_SPLIT)):]
```

```
perm_r = np.random.permutation(len(data_r))
```

```
idx_train_r = perm_r[:int(len(data_r)*(TRAIN_SPLIT))]
```

```
idx_test_r = perm_r[int(len(data_r)*(TRAIN_SPLIT)):int(len(data_r)*(TRAIN_SPLIT+TEST_SPLIT))]
```

```
idx_val_r = perm_r[int(len(data_r)*(TRAIN_SPLIT+TEST_SPLIT)):]
```

```
# Combine depressive tweets and random tweets arrays
```

```
data_train = np.concatenate((data_d[idx_train_d], data_r[idx_train_r]))
```

```
labels_train = np.concatenate((labels_d[idx_train_d], labels_r[idx_train_r]))
```

```
data_test = np.concatenate((data_d[idx_test_d], data_r[idx_test_r]))
```

```
labels_test = np.concatenate((labels_d[idx_test_d], labels_r[idx_test_r]))
```

```
data_val = np.concatenate((data_d[idx_val_d], data_r[idx_val_r]))
```

```
labels_val = np.concatenate((labels_d[idx_val_d], labels_r[idx_val_r]))
```

```
#Shuffling
```

```
perm_train = np.random.permutation(len(data_train))
```

```
data_train = data_train[perm_train]
```

```
labels_train = labels_train[perm_train]
```

```
perm_test = np.random.permutation(len(data_test))
```

```
data_test = data_test[perm_test]
```

```
labels_test = labels_test[perm_test]
```

```
perm_val = np.random.permutation(len(data_val))
```

```
data_val = data_val[perm_val]
```

```
labels_val = labels_val[perm_val]
```

```
model = Sequential()
```

```
# Embedded layer
```

```
model.add(Embedding(len(embedding_matrix), EMBEDDING_DIM, weights=[embedding_matrix],  
                    input_length=MAX_SEQUENCE_LENGTH, trainable=False))
```

```

# Convolutional Layer
model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))

# LSTM Layer
model.add(LSTM(300))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='nadam', metrics=['acc'])
print(model.summary())

early_stop = EarlyStopping(monitor='val_loss', patience=3)

hist = model.fit(data_train, labels_train, \
                 validation_data=(data_val, labels_val), \
                 epochs=EPOCHS, batch_size=40, shuffle=True, \
                 callbacks=[early_stop])

plt.plot(hist.history['acc'])
plt.plot(hist.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()

labels_pred = model.predict(data_test)
labels_pred = np.round(labels_pred.flatten())
accuracy = accuracy_score(labels_test, labels_pred)
print("Accuracy: %.2f%%" % (accuracy*100))

```

```

print(classification_report(labels_test, labels_pred))

history = hist

acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)

plt.plot(epochs, acc, 'bo', label='Training Accuracy')
plt.plot(epochs, val_acc, 'r', label='Validation Accuracy')
plt.title("Training and Validation Accuracy")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True) # Adding grid

# Annotating the maximum validation accuracy point
y_arrow_acc = max(val_acc)
x_arrow_acc = val_acc.index(y_arrow_acc) + 1
plt.annotate(f'Peak Acc: {str(round(y_arrow_acc, 4))}',
            (x_arrow_acc, y_arrow_acc),
            xytext=(x_arrow_acc + 5, y_arrow_acc + .02),
            arrowprops=dict(facecolor='orange', shrink=0.05))

plt.xticks(epochs)

plt.figure()
plt.plot(epochs, loss, 'bo', label='Training Loss')
plt.plot(epochs, val_loss, 'r', label='Validation Loss')
plt.title("Training and Validation Loss")
plt.xlabel('Epochs')

```

```

plt.ylabel('Loss')
plt.legend()
plt.grid(True) # Adding grid

# Annotating the maximum validation loss point
y_arrow_loss = max(val_loss)
x_arrow_loss = val_loss.index(y_arrow_loss) + 1
plt.annotate(f'Peak Loss: {str(round(y_arrow_loss, 4))}',
             (x_arrow_loss, y_arrow_loss),
             xytext=(x_arrow_loss + 5, y_arrow_loss + .02),
             arrowprops=dict(facecolor='orange', shrink=0.05))

plt.xticks(epochs)
plt.show()

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.metrics import precision_score, recall_score, f1_score, confusion_matrix, roc_
auc_score, roc_curve

# Calculate precision, recall, F1-score, confusion matrix, ROC-AUC score
precision = precision_score(labels_test, labels_pred)
recall = recall_score(labels_test, labels_pred)
f1 = f1_score(labels_test, labels_pred)
conf_matrix = confusion_matrix(labels_test, labels_pred)
roc_auc = roc_auc_score(labels_test, labels_pred)

# Plot confusion matrix
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='g', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('True')

```

```

plt.title('Confusion Matrix')
plt.tight_layout()
plt.show()

# Plot ROC curve
fpr, tpr, _ = roc_curve(labels_test, labels_pred)
plt.figure(figsize=(6, 4))
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (area = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.tight_layout()
plt.show()

# Display metrics in tabular form
metrics_table = pd.DataFrame({
    'Metric': ['Precision', 'Recall', 'F1-Score', 'ROC-AUC Score'],
    'Value': [precision, recall, f1, roc_auc]
})
print(metrics_table)

```

APPENDIX 2

PLAGARISM REPORT

Project_Report edited.docx

by Brindha R

Submission date: 15-Apr-2024 02:26PM (UTC+0530)

Submission ID: 2220264684

File name: Project_Report_edited.docx (1.12M)

Word count: 9574

Character count: 59583

Project_Report edited.docx

ORIGINALITY REPORT

5 %	3 %	2 %	3 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.scilit.net Internet Source	<1 %
2	Submitted to Liverpool John Moores University Student Paper	<1 %
3	assets.researchsquare.com Internet Source	<1 %
4	Submitted to Victoria University Student Paper	<1 %
5	mdpi-res.com Internet Source	<1 %
6	www.frontiersin.org Internet Source	<1 %
7	Atharva Balasaheb Chivate, Pratiksha D Chopade, Shreeshail Chitpur, Om N Chavan, Shailaja Uke. "Comparative Analysis of Multiple ML Models and Real-Time Translation", 2023 IEEE 5th International Conference on Cybernetics, Cognition and	<1 %

Machine Learning Applications (ICCCMLA), 2023

Publication

8	Submitted to Berlin School of Business and Innovation Student Paper	<1 %
9	Jay Nanavati, Unnati Patel. "Hybrid Model for Analysis of Social Media Posts for Identification of Depression and Measuring Its Severity", 2023 International Conference on Data Science and Network Security (ICDSNS), 2023 Publication	<1 %
10	www.semanticscholar.org Internet Source	<1 %
11	discovery.researcher.life Internet Source	<1 %
12	"Table of Contents", 2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD), 2023 Publication	<1 %
13	vdoc.pub Internet Source	<1 %
14	www.kdnuggets.com Internet Source	<1 %
15	journals.plos.org Internet Source	<1 %

16	Alvian Daniel Susanto, Steven Andrian Pradita, Caroline Stryadhi, Karli Eka Setiawan, Muhammad Fikri Hasani. "Text Vectorization Techniques for Trending Topic Clustering on Twitter: A Comparative Evaluation of TF-IDF, Doc2Vec, and Sentence-BERT", 2023 5th International Conference on Cybernetics and Intelligent System (ICORIS), 2023 Publication	<1 %
17	Submitted to City University Student Paper	<1 %
18	Submitted to Indian Institute of Information Technology, Design and Manufacturing - Kancheepuram Student Paper	<1 %
19	Submitted to The Robert Gordon University Student Paper	<1 %
20	utilitiesone.com Internet Source	<1 %
21	Submitted to Middlesex University Student Paper	<1 %
22	Submitted to PES University Student Paper	<1 %
23	S. J. R. K. Padminivalli V., M. V. P. Chandra Sekhara Rao. "Deep Aspect-Sentinet: Aspect Based Emotional Sentiment Analysis Using	<1 %

Hybrid Attention Deep Learning Assisted
BILSTM", International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems,
2024

Publication

24	Submitted to Sharda University Student Paper	<1 %
25	www.researchgate.net Internet Source	<1 %
26	ax39t.treasure-gnss.eu Internet Source	<1 %
27	fastercapital.com Internet Source	<1 %
28	link.springer.com Internet Source	<1 %
29	www.jetir.org Internet Source	<1 %

Exclude quotes On

Exclude matches


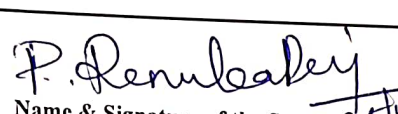
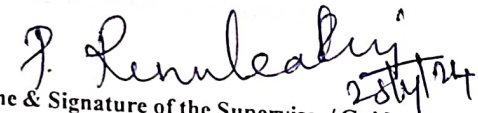

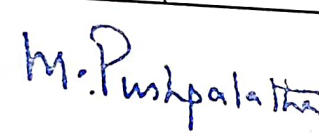
< 10 words


Exclude bibliography On

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)		
Office of Controller of Examinations		
REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES (To be attached in the dissertation/ project report)		
1	Name of the Candidate (IN BLOCK LETTERS)	KARTIK SINGH
2	Address of the Candidate	Y-403, Abode Valley, Kakkan Street, Potheri, Kattankulathur, Chennai, TN-603203
3	Registration Number	RA2011003010810
4	Date of Birth	11 April 2002
5	Department	Computer Science and Engineering
6	Faculty	Engineering and Technology
7	Title of the Dissertation/Project	PSYCHOLOGICAL ANALYSIS USING SOCIAL MEDIA TWEETS
8	Whether the above project /dissertation is done by	<p>Individual or group : (Strike whichever is not applicable)</p> <p>a) If the project/ dissertation is done in group, then how many students together completed the project :</p> <p>b) Mention the Name & Register number of other candidate:</p> <p>UTKARSH GUPTA (RA2011003010772)</p>
9	Name and address of the Supervisor / Guide	Mrs. P. Renukadevi Assistant Professor Department of Computing Technologies SRM Institute of Science and Technologies Kattankulathur - 603203 Mail ID: renukadp@srmist.edu.in Mobile Number: 9962200670
10	Name and address of Co-Supervisor / Co- Guide (if any)	NIL
1	Software Used	Turnitin
1	Date of Verification	
2		

Plagiarism Details: (to attach the final report from the software)				
Chapter	Title of the Chapter	Percentage of similarity index (Including self-citation)	Percentage of similarity index (Excluding Self-citation)	% of plagiarism after excluding Quotes, Bibliography, etc.,
1	INTRODUCTION	< 1%	< 1%	< 1%
2	LITERATURE SURVEY	< 1%	< 1%	< 1%
3	SYSTEM DESIGN	< 1%	< 1%	< 1%
4	DESIGN AND IMPLEMENTATION	< 1%	< 1%	< 1%
5	RESULT AND DISCUSSION	< 1%	< 1%	< 1%
6	CONCLUSION AND FUTURE DEVEL	< 1%	< 1%	< 1%
Appendix		< 1%	< 1%	< 1%

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

 Signature of the Candidate	 Name & Signature of the Staff 25/4/24 (Who uses the plagiarism check software)
 Name & Signature of the Supervisor/ Guide	 Name & Signature of the Co-Supervisor/Co-Guide
 Name & Signature of the HOD	



LETTER OF ACCEPTANCE

ICAAIC

3rd International Conference on Applied Artificial Intelligence
and Computing (ICAAIC-2024)

5-7, June 2024 | Salem, India

icaaic.contact@gmail.com | <http://icaaic.com/2024/>

Letter of Acceptance

Author Name: Kartik Singh, Utkarsh Gupta, P. Renukadevi

Affiliation Details: SRM Institute of Science and Technology, India.

Dear Author:

It is with great pleasure that we extend our warmest congratulations to you on the acceptance of the paper titled **"Psychological Analysis Using Social Media Tweets Using ML Models"** - **PAPER ID: ICAAIC 805** for presentation at the 3rd International Conference on Applied Artificial Intelligence and Computing, scheduled to be held in R P Sarathy Institute of Technology, Salem, India from June 5th to June 7th, 2024.

Your submission was subjected to a rigorous review process, and the result that your paper has been selected for inclusion in our conference program. We believe that your contribution will greatly enrich the discussions and knowledge exchange at our event.

Your participation will undoubtedly contribute to the success of the 3rd International Conference on Applied Artificial Intelligence and Computing.

Once again, congratulations on your acceptance, and we anticipate your valuable contribution to our conference.

Yours Sincerely,



Dr. Munusami Viswanathan,
Principal,
R P Sarathy Institute of Technology,
Salem, Tamil Nadu, India.



PAYMENT PROOF

RECEIPT

Organizer

intlconf.publications

intlconf.publications@gmail.com

Receipt #: 11385436

Created: Apr 22, 2024

Bill To

Kartik Singh

ks9124@srmist.edu.in

Event Name:3rd International Conference on Applied Artificial Intelligence and Computing

Order Id:231031713769313453

Item(s)	Qty	Rate	Amount
Student Non IEEE Member	1	7750.00	7750.00

Sub Total : 7750.00

Booking fees : 182.90

Total : **INR**
7932.90

