Hate Speech Detection in Hinglish Text using Deep Learning

A Report submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Technology

Submitted by

Pratham Bhardwaj (191018)

Ritika Tiwari(191028)

Dhruv Thakur(191035)

Under the supervision of

Dr. Naveen Jaglan

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May 2023

Table of Contents

Candidate's Declaration

We hereby certify that the research presented in this report, "Hate Speech Detection in Hinglish Text Using Deep Learning," which was submitted to the Department of Electronics and Communication Engineering at Jaypee University of Information Technology in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology, is an authentic record of our ownwork performed during the period from January 2022 to May 2022 under the direction of **Dr. Naveen Jaglan**, Associate Professor, Department of Electronics and Communication Engineering, Jaypee University Of Information Technology.

We have not applied for any other degree from this Institute/University or any other Institute/University based on the information contained in this report.

(Pratham Bhardwaj, 191018) **(Ritika Tiwari, 191028)** **(Dhruv Thakur, 191035)**

This is to certify that, to the best of my knowledge and belief, the aforementioned statement made by the candidates is accurate.

Dr. Naveen Jaglan

Date: **Associate Professor**

ACKNOWLEDGMENT

We would like to express our heartfelt appreciation to everyone who has assisted us in bringing this project to a successful conclusion. Throughout the course of the job, we encountered numerous difficulties due to our lack of knowledge and expertise; nonetheless, these individuals assisted us in overcoming all of the challenges and in ultimately compiling our concept into a sculpture of a shape.

We therefore take this chance to express our sincere gratitude to our supervisor, **Dr. Naveen Jaglan**, Associate Professor, Department of Electronics and Communication Engineering, Jaypee University Of Information Technology, for his knowledgeable direction, creative suggestions, constant encouragement, and patience with which he handled us and the project. His level-headed approach to the project gave us courage and equipped us to overcome the next challenges.

We will always be thankful to **Dr. Rajiv Kumar**, the department head of the department of electronics and communication engineering, for his support and thoughtfulness while we conducted our study. His ongoing assistance has been really helpful to us as we pursue our studies.

(Pratham Bhardwaj, 191018) **(Ritika Tiwari, 191028)** **(Dhruv Thakur, 191035)**

List of Abbreviations/Acronyms

CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
TP	True Positive
FP	False Positive
Bi-LSTM	Bidirectional Long Short-Term Memory
TN	True Negative
FN	False Negative

List of Figures

List of Tables

ABSTRACT

Humanity is fortunate to have the Internet, but it has also becoming more exploited. Social networking sites like Instagram and Twitter are where users most frequently express their ideas. It may annoy readers when users use harsh or inflammatory language. Code-switching is usually semantically difficult in linguistically diverse, low-resource languages, and there aren't many sophisticated methods for reliably detecting hate speech in real-world data.The goal of this study is to evaluate and compare the effectiveness of a number of deep learning algorithms created to find instances of hate speech on popular social media sites that use Hinglish (an English-Hindi code-mix) language. This article's goal is to look at a number of deep learning algorithms for identifying hate speech on popular social media platforms in English, Hindi, and Hinglish. In order to identify hate speech from tweets and comments in Hinglish, English, and Hindi tweets, we implement and analyse various deep learning approaches as well as a number of word-embedding techniques (Glove, Fasttext, Word2vec) using a consolidated dataset of about 21800 occurrences.

In this paper, we applied and evaluated several deep learning algorithms along with different embedding techniques on a amalgamated dataset of 21748 instances, for speech recognition from comments, tweets, etc. CNN-Bi-LSTM with Fasttext word embedding technique provides the best results. It yield accuracy(0.72), precision(0.69), recall(0.69), F1-score(0.72) and ROC-AUC(0.76).

Chapter 1

Introduction

Along with population expansion during the past decade, the popularity of social media has expanded dramatically. Social networking platforms such as Twitter and Facebook enable individuals to express and share their opinions on a worldwide scale, which has led to an influx of text data in these sites. Although these social media websites give forums for people to engage, they also have a negative side. The extensive use of such platforms has resulted the dissemination of provocative and hostile information, resulting in cyber violence.

Online social networking channels are not only a source of amusement or a place for people to talk to one another; rather, in this age of information technology, they have also become an essential component of people's social lives. The yearly rise in individuals using the internet is 31.2% points in India, which is more than 78 million people. This is mostly due to the fact that children and young, particularly in India, are very enthusiastic about social media sites [3]. At the forefront of our social life is the vast social media platform. Currently, more than 29% of India's population uses social media [10]; this percentage is expanding quickly. Contradictory ideologies and nasty content on the Internet have increased as social media usage has grown. Hate speech is defined as "public communication that expresses hate or advocates violence towards a person or group based on anything such as race, religion, sex, or sexual orientation," according to Wikipedia [13], which cites the Cambridge dictionary definition of the term. The term "hate speech" refers to a number of comments, both written and spoken, that advocate, encourage, support, or excuse acts of hatred, violence, or prejudice directed towards an individual or group of individuals for a variety of reasons [1].

Because of hostile rhetoric, there is the possibility of hatred and violence crimes being committed. Exposure to it has the potential to have dramatic effects on a person's mental state, including increased tension, anxiety, sadness, and desensitisation. There has been a link shown between victimisation in either its direct or indirect form and increased incidence of substance abuse [2]. Handling such a massive amount of data on social media platforms is not a simple task. On average, Twitter processes around 6,000 new tweets every second. In 2021, Instagram had around 1.386 billion active users, while Facebook had 2.85 billion users and 1.3 billion people were using Facebook Messenger [2].

Individuals' mental health is negatively impacted by hateful words [14]. During the COVID-19 outbreak, hate speech on Twitter has increased by around 9 times [21]. From June to September 2019, YouTube reported around 500,000,000 hateful comments [9]. Numerous automated methods have been created to identify and prevent hateful speech, which has been identified as a major societal issue.

After conducting extensive research, we have determined that a substantial amount of research has been conducted on the English language, and that it has been successfully applied in several bots including Twitter. However, its performance in India is low due to the linguistic diversity of its customers. The bulk of Indian users express themselves in Hinglish, a mixture of Hindi and English. Based on a survey of the relevant literature, it has been determined that relatively little research has been conducted employing deep learning-based algorithms for hate speech identification. Nevertheless, there is a significant performance disparity in the identification of hate speech. The intricacy of language, in addition to non-standard variances in syntax, spelling, and interpretation of language, makes it a challenging process to identify hate speech and other objectionable information that may be present in social media. A significant amount of research has been conducted on the identification of hate speech from tweets written in English. However, as the use of local languages in social media continues to develop, there is a greater need for academics to analyse tweets in order to identify instances of hate speech. [5]. Facilities for social networking offer a channel via which information may be spread, as well as our opinions, perspectives, and concepts about information. Because of this, local languages are beginning to incorporate ever more English into their vocabulary. The use of code-mixed Hindi and English, sometimes known as Hinglish, is becoming more common- place among the majority of metropolitan populations across the world. The majority of social networking sites' hate speech detection algorithms are unable to remove hateful and abusive information that is submitted in code-mixed languages.

Deep learning is being studied and developed for the purpose of detecting hate speech in Hinglish text. By combining elements of Hindi with English, the language known as Hinglish is becoming increasingly popular in South Asia and India. Speakers in these regions continue to become more aware of this unusual language phenomena. The peculiar blending of two languages makes it difficult to identify hate speech in Hinglish literature.

The use of deep learning approaches has demonstrated the effectiveness of identifying hate speech across a variety of language domains, including English. Due to the intricacy of the presentation, it was successful in recognising expressions of hatred in a variety of languages.

The absence of significant, labelled datasets presents one of the difficulties in detecting hate speech in Hinglish writing. Such datasets demand a large investment of time and money. However, some initiatives to produce Hinglish hate speech datasets have been made recently, such as the dataset created by Sharma et al. (2021).

Several investigative inquiries have used convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for identifying hateful expressions within Hinglish text as far as the application of advanced learning techniques is concerned. For instance, Singh et al. (2021) classified Hinglish text as hate speech or non-hate speech using a mix of CNNs and RNNs. On a small dataset, their approach produced accuracy of about 80%.

Transfer learning, in which a model that has already been trained on a sizable English language dataset is refined on a smaller Hinglish dataset, is another method that has been utilised for hate speech identification in Hinglish text. Given the dearth of labelled datasets, this technique's use has proven effective in a number of additional languages, including Arabic and Turkish. The efficiency of transfer learning for detecting hate speech in Hinglish literature, however, requires further study. In the internet age, fake reviews have become a major problem. Fake reviews may ruin businesses' reputations, mislead customers, and cost money. To recognise false reviews and stop them from spreading, employ machine learning algorithms.

Determine whether a review is genuine or not using hate speech recognition. Large datasets of reviews and their related labels (true or false) may be used to train machine learning algorithms to identify patterns and characteristics that discriminate between real and bogus reviews.

Consequently, the global hate speech detection rate, which is around 45 percent, reduces even more when the material in Indian vernacular languages and slang's is taken intoconsideration. Major contribution of the suggested work:

- Outlining a method for deep learning to identify hate speech.
- Here, twelve distinct deep learning models are examined.
- A dataset of 21,748 records is used to evaluate the proposal 21,748 records.Threepublicly available datasets Kumar 2018 [32], Bohra 2018 [8], and HASOC 2021 [19].

Application

There are several applications regarding the present attempt. We may utilise hate speech identification in messaging bots and other apps to automatically filter out hostile information. Also, law enforcers and authorities may utilise it in order to detect hate speech and keep the peace and preserve law and order, particularly during protests and social disturbance. As is often observed, hate speech is used to incite mob violence against a certain religion, caste, or gender. This study may aid in reducing such negative occurrences in society, particularly against minority communities.

Structure of Report

The report is divided into five sections:

- The second chapter outlines some earlier research work that inspired this study.
- The third chapter discusses the proposed model with training dataset and assessment metrics and explains the main principles employed in its development.
- The fourth chapter shows the produced findings and a comparison of the suggested work to the state-of-the-art, as well as the created caption with images.
- Chapter 5 describes the conclusion and future work.

Chapter 2

Literature Survey

Objective

Over the last decade, social media has adult dramatically in terms of each scale and utility as a communication tool. Because of the character of social media, anybody could publish no matter they want, in any position, whether it's informative, conflicting, or simply concerning everywhere. These messages may be seen by ample people, consistent with the site. Different communities have multiple interpretations of improper content and ways of recognizing it, but because of the volume of help, computerized approaches are a significant part of the job. This helps Hate speech is a major component of this offensive material. In the past, apps that used Deep Learning also established milestones in activities that were connected [6]. In this part, a literature overview on the identification of hate speech using deep learning algorithms is presented, along with publications providing data sets linked to the code mixing dataset in English and Hindi [27].

Previous Papers

- A dataset with a mix of Hindi and English has been included in the study [16]. In the course of this research, the writers compile and annotate a collection of data in English and Hindi sourced from Twitter and Facebook. The corpus is created on three layers using the data from 18 thousand Tweets and 21 thousand comments posted on Facebook. The dataset in issue did not undergo any experimental testing over the course of this investigation.

- The research presents a concept for an autonomous vehicle that takes use of a compiled database of comments users have made on YouTube films. [18]. The dataset in question is comprised of 50,000 comments left on Youtube clips, and its data structure and potential applications are also included. In addition, the authors address the case study that was suggested for the corpus in order to acquire a better grasp of how the general public feels about driverless cars and how people react to accidents involving cars.

- The authors of the work suggested text categorization by making use of English-Hindi Mixed text. [24]. In addition, the writers obtained random English-Hindi Mixed data from Facebook comments and news articles. They offered many different identification techniques based on the TF-IDF representation, and they came to the conclusion that the Radial Basis Function Neural Network was the strategy that produced the best results for classifying Hinglish text.

- Deep Learning (DL) approaches for the identification of hate speech from English-Hindi code mix data on benchmark datasets were suggested by the authors of the research [15]. They tested with DL models. The results with the CNN model had an accuracy of 82.62 percent, a precision of 83.34 percent, and an F-score of 80.85 percent.

- The study [28] suggests analyzing data from social media platforms that include both Hindi and English code mixes. The monolingual embedding was used in the first part of the research, but in the second part, they used a supervised classifier that used transfer learning on an English dataset and then tested it on a code-mix dataset. The transfer learning was performed on an English dataset. They recorded an increase of 0.019 points in the F1 score.

- In a similar vein, the research [20] proposes a deep learning model to identify objectionable tweets written in Hindi or English. The authors of this work offer a unique twitter dataset, which they call ”Hindi-English Offensive Tweets,” in the course of their study (HEOT). The tweets were divided into three groups: those that were not objectionable, those that were abusive, and those that constituted hate speech. They evaluated the data using the CNN model and obtained an accuracy of 83.90%, a precision of 80.20%, a recall of 69.98%, and an F1-score of 71.45%.

- The authors of the article [22] developed a hate detection technique in each of the three languages they used (English, Spanish and Italian). They offered the methodology for measuring the association between abusive language and sexism, and they discussed the breadth of the problem when applied to cross-lingual platforms. Automatic Misogyny Identification (AMI) system provides the data that they use in their research, which they then analyse. As a way to wrap up their investigation, they observe that sexism is a kind of abusive language and that the suggested design offers dependable performance across all languages.

- The authors ofthe research [11] developed a pipeline for the identification of hate-speech on social media platforms using English-Hindi code-mix data. Before making the suggested system their final product, the authors conducted regress comparison experiments using various benchmark datasets that were accessible. In addition to this, they examined the link between hate embeddings and social network-based characteristics and found that the suggested approach outperformed the state of the art in this regard.

- They suggest a method based on deep learning in the study [26] for identifying the emotional state of hate speech. The authors compiled a dataset comprising over 10000 Hindi-English code mixes. They made the feature vectors with a bilingual model and used a deep neural network as the classification model.

- Cyberbullying may be detected using an unsupervised learning algorithm proposed by Michele Di Capua and his colleagues [12]. Nearly 54,000 YouTube data sets were collected, and each one was meticulously annotated. The GHSOM was trained and tested using a K-fold technique with K = 10. As a consequence, they've improved their accuracy by 64%.

- Using deep learning, Hugo Rosa et al. (2018) [25] suggested a method for identifying cyberbullying. Using the Kaggle platform, they have gathered data for their study. Using CNN's approach, they were able to get a 64.9 percent success rate with Google embeds.

- Viviana Patti et al. (2019) [23] suggested a method that is based on hybridization to identify instances of hate speech. In this particular piece of research, they used two different models. They used a LSVC in their first model, and then they used a LSTM neural model of word embeddings in their second model. Both models were developed by the same team. Joint learning to use a multilingual word embedding model had the highest results, scoring 68.7 percent on the F1-score scale.

- Safa Alsafari et al. in 2020 [4] proposed a hate speech detection technique for Arabic social media.The data collection for this study was compiled utilising the Twitter search API. The data set was cleaned up during the data preprocessing by removing words like stopwords, URLs, punctuation, symbols, and tags that weren't necessary.To obtain 75.51 percent of the F1-score, they created a three-class categorization using CNN and Bert.

When considering data that included a combination of Hindi and English, it was found during the course of a review of the pertinent literature that the method for identifying instances of hate speech has serious flaws. As may be seen in the following, three significant problems were located and fixed. One of the main problems is, to begin with, the lack of large datasets accessible for training. Second, the difficulties posed by mixed-code data have been overcome by combining tried-and-true techniques with state-of-the-art new deep learning techniques. In the end, we conduct a detailed examination of all the methods using all the performance measures currently available.

In particular, the following goals of machine learning-based hate speech identification are included:

1. Machine learning models may be trained to recognise the patterns and characteristics of fake reviews, such as the use of certain phrases, the frequency of reviews from particular users, or unique review timings.

2. Improving review quality: By eliminating fake reviews, users will have access to more accurate and useful information overall thanks to the platform's reviews.

3. Maintaining the standing of the review platform: Customers may cease trusting reviews from a website that has a reputation for publishing fake reviews. By utilising machine learning to recognise and remove fake reviews, the platform can maintain its reputation for providing accurate and trustworthy ratings.4. Conserving resources and time

The process of manually checking each and every review for validity takes a lot of time and resources. Platform owners may save time and resources that could be utilised for other things by employing machine learning to automatically detect bogus reviews.

Generally speaking, the main objective of hate speech identification by machine learning is to ensure that users can make informed decisions based on real and reliable assessments.

1. Obtaining data The initial step is to collect information from other websites, such as social networking platforms, online marketplaces, and review websites. These data sources are able to provide information that is labelled and includes both fake and genuine reviews. It is essential to ensure that the data reliably and diversely covers various product categories, industries, and languages.

2. Data pre-processing: After the data have been collected, they must be pre-processed to remove irrelevant information and perform text-cleaning operations like tokenization, stemming, and stop-word removal. The pre-processed data is used to produce a numerical representation, like a bag.

3. Feature selection: Choosing the right features is essential for improving the model's performance and reducing the dimensionality of the input. In this step, the qualities that are most crucial for identifying whether reviews are genuine or fake are picked. Examples of feature selection techniques include the chi-square test, mutual information, and correlation-based feature selection.

4. Model selection: Reviews may be classified using machine learning models such random forests, decision trees, logistic regression, and support vector machines. The data's characteristics and the essential performance requirements have an impact on the model selection. Either an unsupervised learning method, where the model finds patterns in the data without any labels, or a supervised learning strategy, where labelled data is utilised to train the model, can be used to teach the models.

5. Model evaluation: Following model training, it is essential to evaluate the model's performance using measures like recall, accuracy, and F1 score. Cross-validation methods, such as k-fold cross-validation, can be used to ensure that the model is generalizable.

6. Model Compression: Out of all the models utilised, the best one is selected.

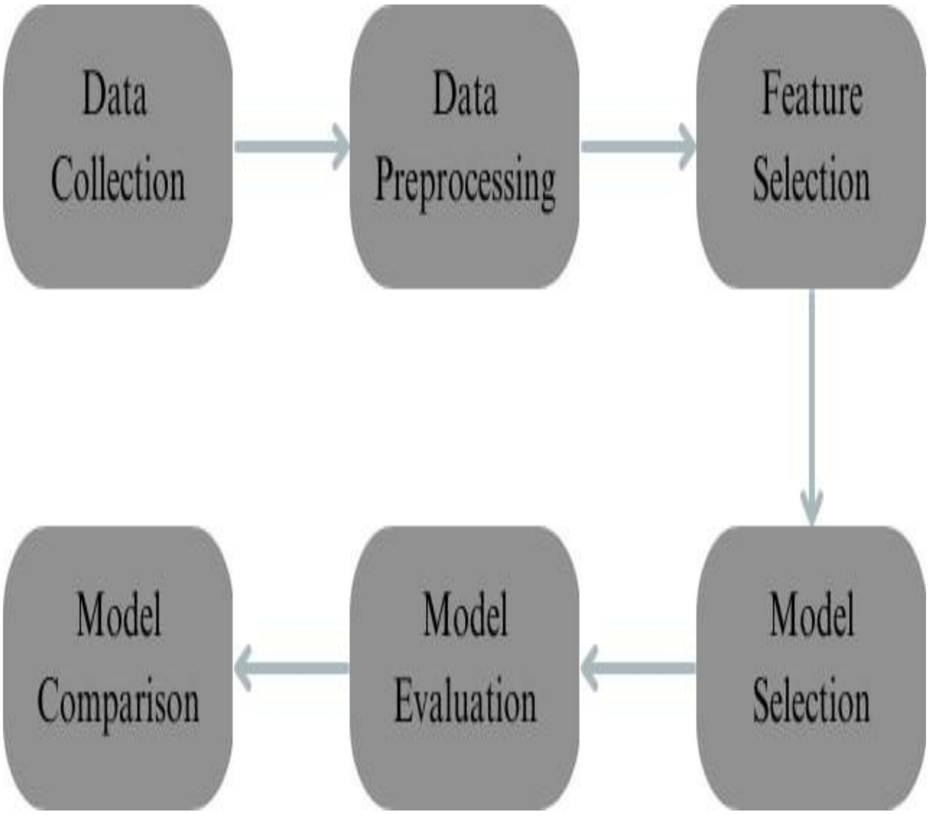


Figure 2.1 System Methodology

Chapter 3

Proposed Work

In this section we describes the datasets we used for the work. It states about the tasks we did regarding the pre-processing of the data. We also evaluated the deep learning approach in one of the subsections. In this part, an unique methodology for achieving state-of-the-art performance in identifying hate speech is presented.

Table 3.1: Bohra-2018 dataset

Text of the Tweet	Label
Pappu ka ye saja phansi se kam nahi	0
Sir phansi nahi. .. sirf looted money wapas chaiya	0
Code phatta hai toh phatne do, Mujhe project se nafrat ho gayi hai, Mujhe development team se hatne do	1
Narendra modi ura hopeless PM... Apne garib ko maar daala...... I hate u.......Sirf aap bol bacchan dete ho karte kuch bhi nhi........U r fake person	1
Provinces or Banana to achi baat hay. Magar Pakistani bun Jana to Fakhar ki baat hui na. Baat baat p nafrat. Achi nahi	0
mujhe to dono se hi nafrat he. par BJP se mujhe itni nafrat he Zindagi me ise vote nhi dunga. kyonki ye party vote ke laayak hi nhi he.	1

Dataset Description

In the field of hate speech detection the researchers are now doing work from many years. We have quite a good availability of datasets in English language and also in some other native languages also like Arabic, French and Hindi. But main problem arise in the case of Hinglish datasets where we found that there are limited datasets and also relatively smaller. So, we

Table 3.2: hasoc-2021 eng-hin

Text of the Tweet	Label
@srivatsayb Twitter is best of all app IStandWithTwitterIndia Modi want to control it to win election by spreading fake propeganda	0
@srivatsayb Feku will be in tension over losing 20 million followers and 48 million Bots if he bans Twitter!! Amit Malware will be doubly tensed as he will have to work double shifts to recreate all those bots in KOOAPP !!! I DONT think self loving feku will go for the ban.	1
@srivatsayb CONgress' gulam dog always has to bark for every decision taken by GOI.	1
@srivatsayb We need India's own Twitter, why not India own Twitter kind of app	0
@srivatsayb yup lets make it an authoritarian regime, please help	1
@srivatsayb Rahul Ka Gulam chamcha..gyan de RHA hai	1

Table 3.3: Kumar-2018

Text of the Tweet	Label
Only 13% of jio customers converted into prime, so they extended date	0
Loot of people's mandate, is it democracy?	1
again the same thing in eyes of judiciary he is not terrorist he is accused in our eyes is terrorist and he deserved to kill who ever keep a bad eye on our nation he are she should be killed without no mercy	1
POSITIONAL BUY JPASSOCIATE ON EVERY DECLINE 13.00-10.50-7.50-5.50 FOR TARGET 36 AND 56 (TIME FRAME 2-3 YEARS)	0
Abhijeet Ravi Yechuri don't have an option than to rest in his laurels.	0
Seriously? Bheem with those fat hands?	1

merged three datasets publicly available to overcome the problem of smaller datasets. Below are the details of the three datasets used in our research including Kumar 2018 [29], Bohra 2018 [8], and HASOC 2021 [19].

- Bohra 2018 [8]: In this dataset, we have 4579 rows of tweets out of 2,918(63.72%) rows are labelled as ”No” that we converted into ’0’ as ”Non-hate” speech and 1,661 rows(36.28%) are labelled as ”Yes” which we converted into ’1’ as ”Hate”. Majority number of tweets contained in the dataset are a combination of Hindi and English and are written using the conventional letters of the Roman script. Sample rows of the dataset are given in the Table 3.1

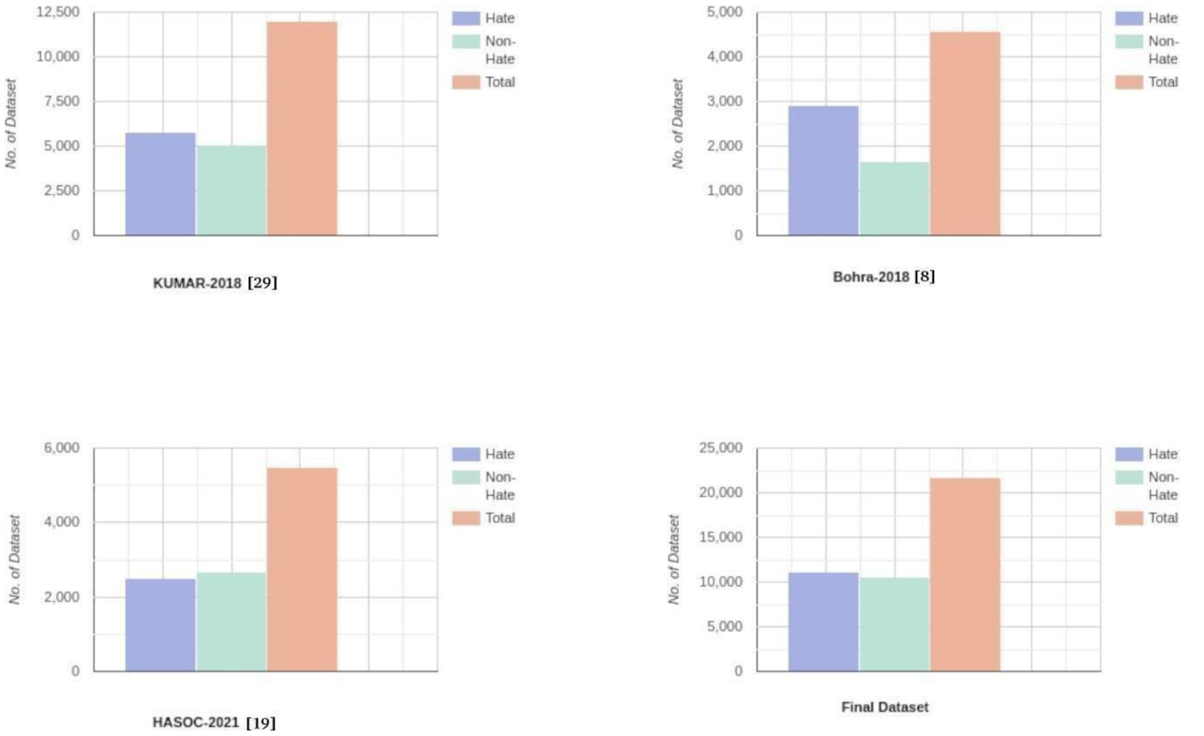


Figure 3.1: Data Distribution

- HASOC 2021 [19]: Tweets from social media platforms have been collected in a coding mix of Hindi and English for this dataset. The data for the Hind-English language mix consists of 5,170 different cases, all of which are classified as either ”Hate” (2504(48.43%) different instances) or ”Non-hate” (2666(51.56%) instances). Some of the rows are provided in Table 3.2

- Kumar 2018 [29]: The majority of the comments left on social media sites were used to compile this dataset. The information was first segmented into three classes: Overtly Aggressive (OAG), Covertly Aggressive (CAG), and Not Aggressive (NAG). We consolidated the two classes, which were originally OAG and CAG, into (”hate” speech) one class, which we referred to as Hate speech and NAG for the other class(“Non-hate” speech). This was done so that it would be comparable to the other two datasets. This dataset contains a total of 11,999 cases, out of which there are 5790(57.90%) instances classified as “Hate” and 5051(42.10%) instances classified as “Non-Hate”. Some samples are shown in Table 3.3

- Final dataset: Final dataset consists a total of 21,748 rows, in which 11,113(51.09%) instances are labelled “Hate” as ’1’ and rest 10,635(48.91%) instances are labelled as “Non-Hate” as ’0’. We got final dataset as a balanced dataset. Finally, the dataset is divided into training(70%), testing(20%) and 10% for validation.

Data Preprocessing

Dataset we got finally can not be fed directly into the model, it needs to be preprocessed. We identified some issues and tried to solve them.

- Removing emojis: Since there are lot number of emoji’s present in our dataset and it can’t be converted directly into the vectors. We tried to detect them and replace them with words that can be better suited. We used demoji library for the required purpose.

- Handling words with ’#’: We found that in many sentences ’#’ character is used like #IStayWithIndia. We make a dictionary of them and then mapped it with like for the example given above - IStayWithIndia : I Stay With India. We made a mapped dictionary for the required work.

- Removing words with ’@’: We found that word occurring with ’@’ are generally the usernames or names that are not of much of significance, so we removed them.

- Converting Devnagri sentences to English sentences : We also found that in our dataset some sentences are written in devnagari or Hindi language. We used a Google API for this purpose and used ’googletrans’ for the required purpose.

- Handling Hinglish words: We tried to convert hinglish word to english words as much extent as possible by assessing the positions of their in the sentence whether their successor words or predecessor words are of hinglish or not and firstly converted them to hindi language using google translate API and then converted them to english language again using ‘googletrans’.

- Replacing common abbreviations: We have created a dictionary of words like shd, k, bcz to should, okay, because to replace these common words as much as possible.

- Removing punctuation: After all the tasks stated above we removed all the punctuation left and replaced them with white-spaces.

- Removing Stopwords: After this we removed stopwords(am, are, we), then tokenised themand lemmatised using NTLK. We remove stopwords using nltk.corpus.stopwords.words('english').

- Stemming and Lemmatization: We then perform stemming and lemmatization using nltk.stem.PorterStemmer() and nltk.WordNetLemmatizer() respectively.

Word embedding techniques used:

- word2vec: It employs a neural network model. Upon training, such a model may identify synonyms or propose new words for a portion of a phrase that has already been completed. According to the program's name, word2vec represents each word with a unique vector of integers. The degree of semantic similarity that exists between the words represented by each vector using the cosine similarity.

- fastText: Developed by Facebook's AI Research team for learning word embeddings and text categorization. The model may be used to develop an unsupervised or supervised learning method for vector representations of words. Unlike other word vector generators, fastText can build word vectors for unknown words or from vocabulary words. This is because it takes morphological properties of words into consideration while creating the word vector for an unknown word.

- GloVe: A paradigm for distributed word representation, was coined from Global Vectors. Vector representations of words may be learned using an unsupervised approach. Accomplished by relating the distance between words to the degree of semantic similarity in a meaningful space. The resultant representations exhibit fascinating linear substructures in the word vector space. Training is based on global co-occurrence of word-word information gathered from a corpus.

Model Architecture

The model that we use in this experiment makes use of both the CNN and long short term memory. CNN is responsible for the extraction of the spatial information. Long-term relationships can be learned by using LSTM. In the first stage of the process, a one-dimensional convolutional layer is utilised so that the input vectors can be analysed and the feature points that are present at the text level can be retrieved. This enables the process to be referred to asa text level feature extraction. In order for the BiLSTM layer to learn the long-term reliance of the local characteristics in tweets and categorise them as either hateful or not hateful, the following steps are taken.

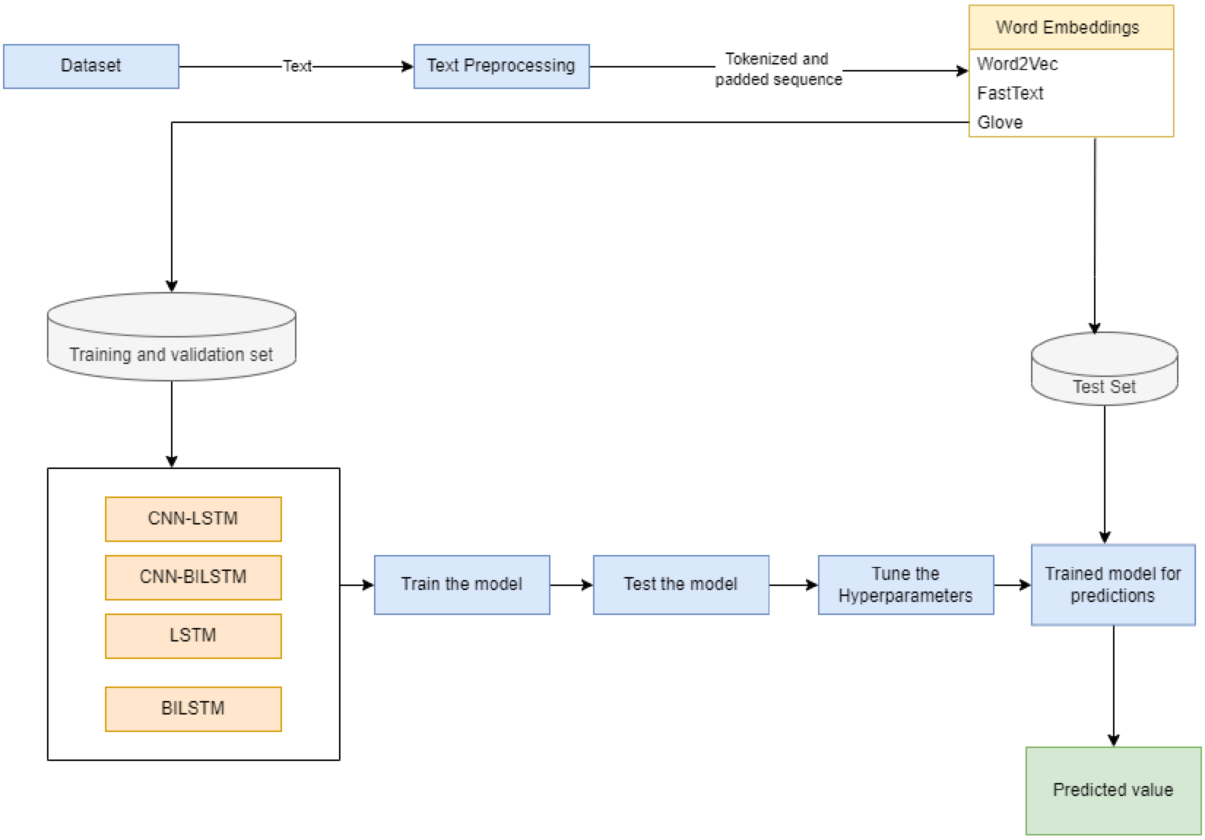


Figure 3.2: Framework used in this work

Implementation Details

The Google Colab is used to aid in the construction of the models. For Jupyter notebooks, it is a cloud environment. Tensor processing units and graphic processing units are integrated for computing.

Vectorization

During this step, the text in question is first tokenized, then turned into vectors. To convert text- based data into an integer sequence, we make use of a tokenizer. The length of the vector is determined according our dataset(80 in our case). The vectors are given pre padding in order to ensure that they are of the same size. In addition to that, we have made use of pretrained word embeddings such as FastText, Word2vec, and glove. In this particular effort, the dimension of embedding has been set at 100. The outputs of these embeddings are sent to the model in order to be used as input for the embedding layer.

Proposed Model Details

The Keras API was used to develop the model . The Sequential model can be broken down into a few different layers.

- The very first layer of the network is known as the embedding layer.By providing the prepared embedding matrix as beginning weights, the pre-trained word embeddings areused in the model.
- The next layer is a Dropout layer with a rate of 0.5.It is used to prevent the overfiting.
- The next layer is a one-dimensional CNN layer known as Conv1D. It consists of 64 filters of size 5x5 and uses activation function which is Rectified linear unit.
- The third layer is pooling layer. It uses MaxPooling1D of filter size 4.
- The following layer, the BiLSTM layer, is the one that gets input from the pooling layer.This data is used to train a Bidirectional LSTM model, which outputs long-term dependent aspects of input while maintaining memory. 128 is used to represent the out-put’s dimension.
- The layer after bidirectional Lstm is a Dropout layer with a rate of 0.5.
- The last layer is the dense layer with one output and uses sigmoid as activation function.

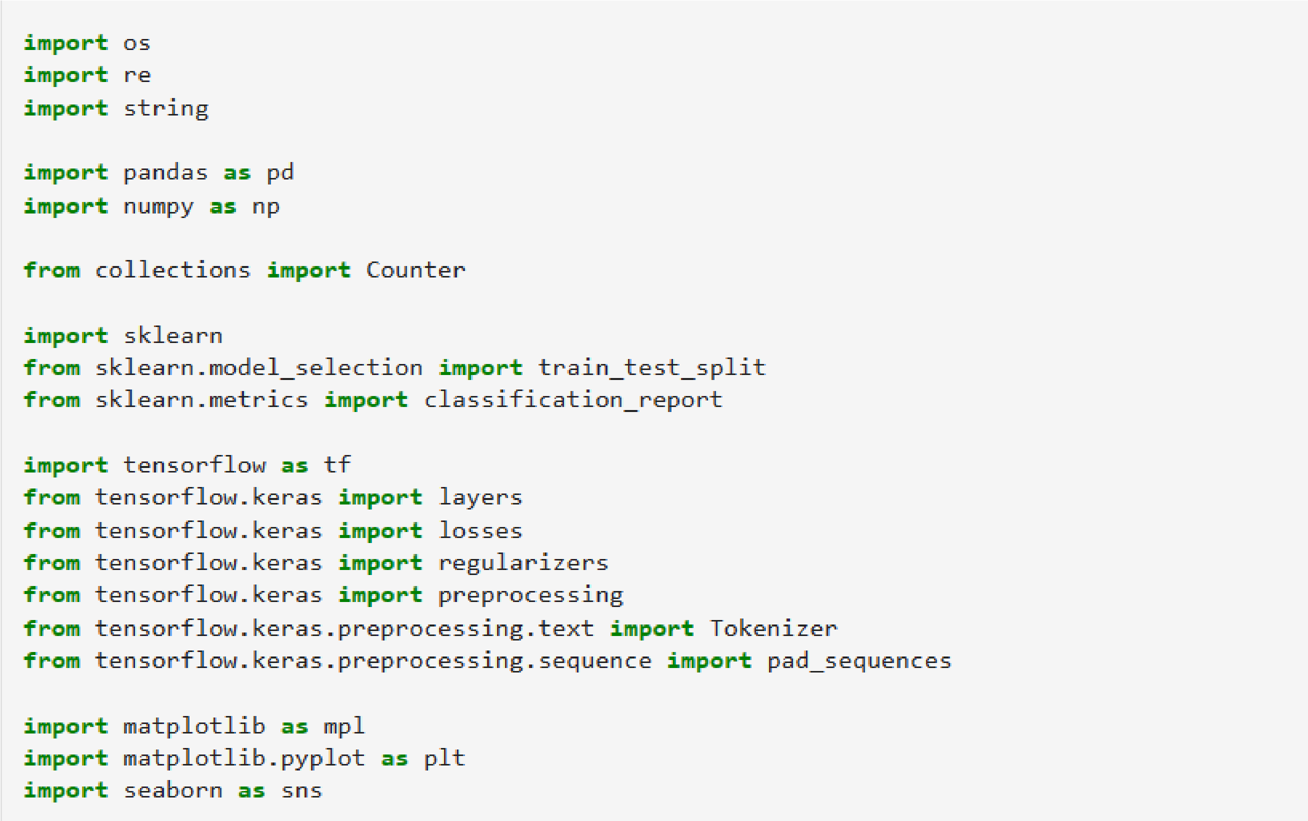
✓ **The Dataset has the following features as shown in the table below:**

1. String: This is a built-in module, therefore we must import it before utilising any of its classes or constraints.

2. Math: There is a built-in module for using it for mathematical tasks.

3. Sklearn: Also known as Scikit-learn, this is the most effective and reliable package for using machine learning in Python.

The code for that is displayed below:



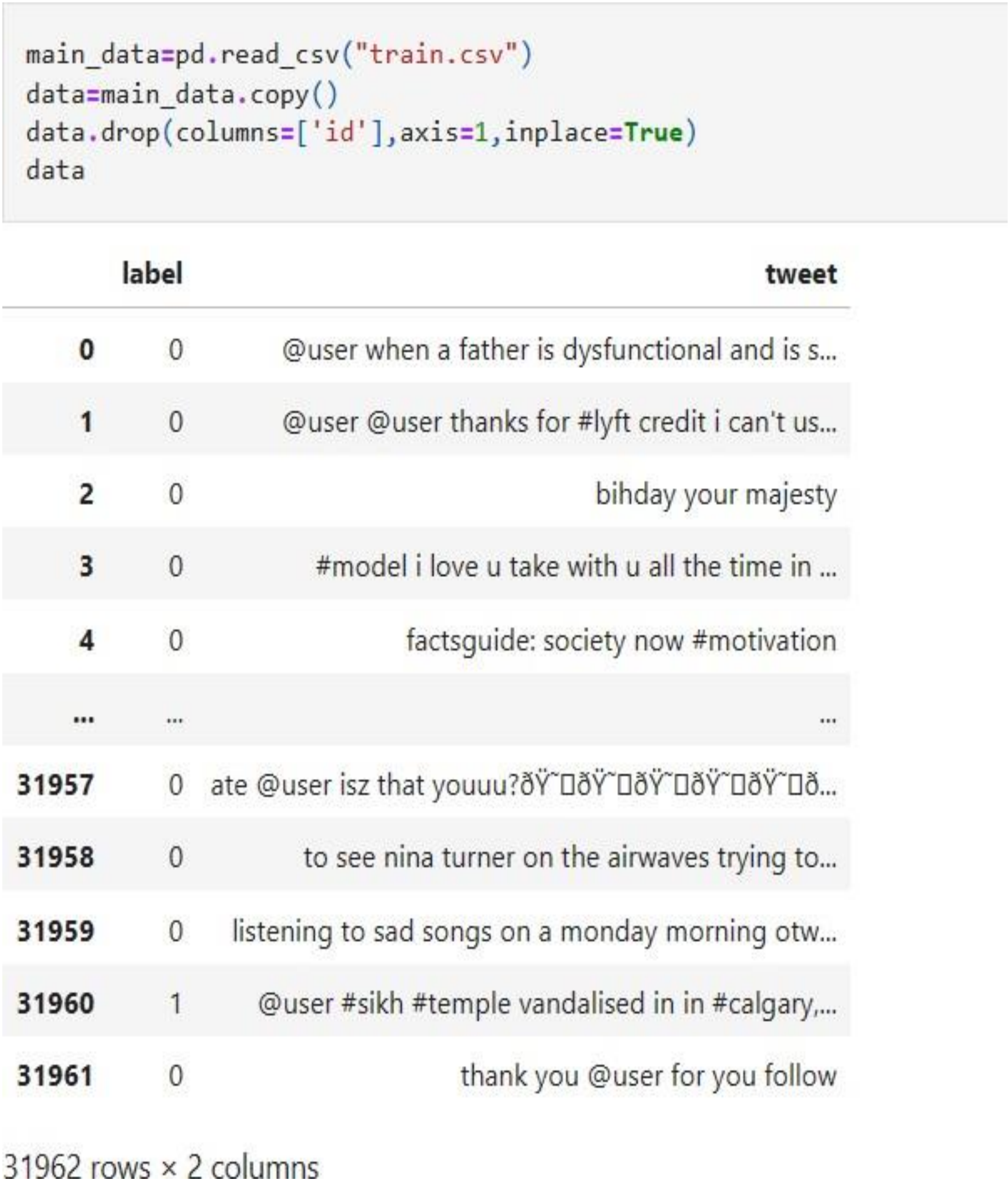
```
import os
import re
import string

import pandas as pd
import numpy as np

from collections import Counter

import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras import losses
from tensorflow.keras import regularizers
from tensorflow.keras import preprocessing
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 3.3: Importing Libraries

➢ **Loading and analyzing the dataset**

In this step we loaded the dataset using the read_csv function of Pandas module and also analyzed it by checking the shape of dataset, name of columns, datatype of each column and at last checked a few entries in the dataset.

The code of the following is shown below:

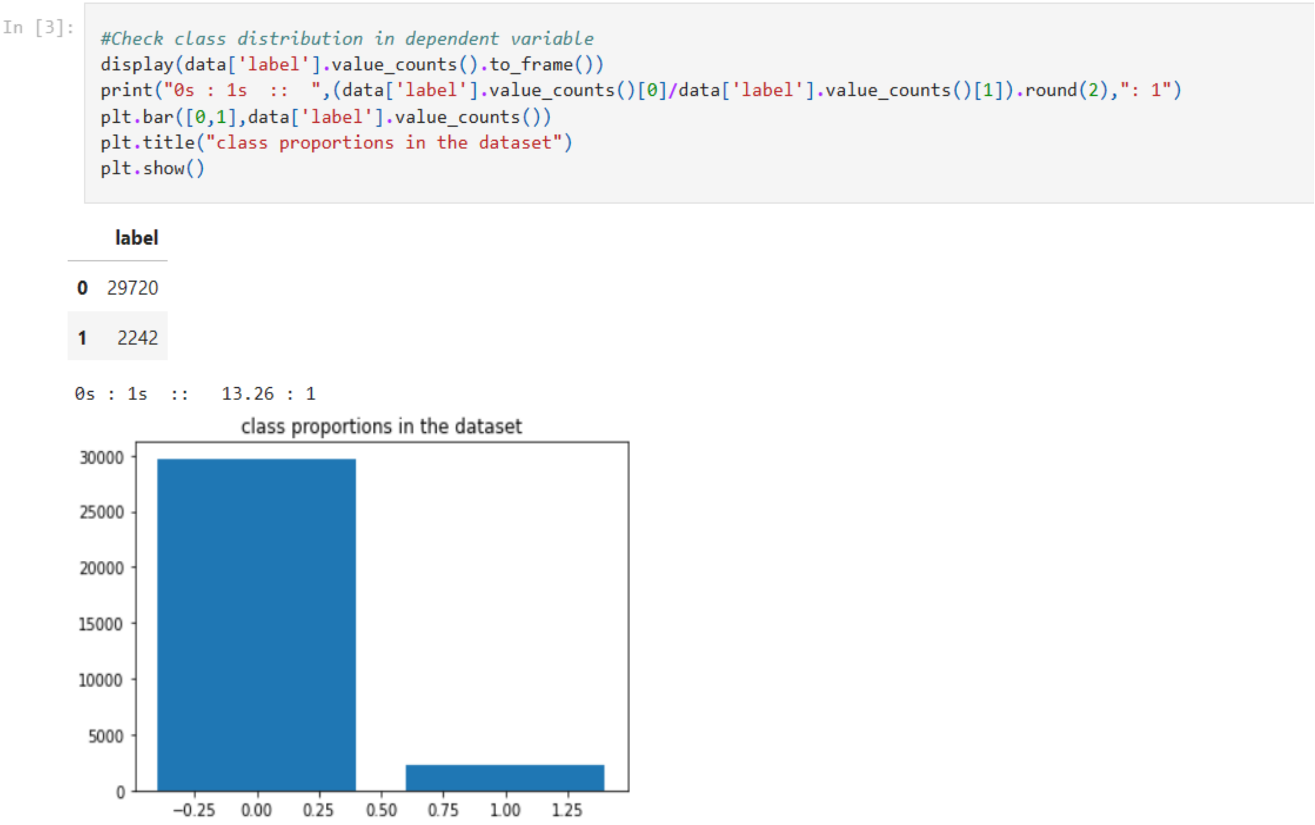


```
main_data=pd.read_csv("train.csv")
data=main_data.copy()
data.drop(columns=['id'],axis=1,inplace=True)
data
```

	label	tweet
0	0	@user when a father is dysfunctional and is s...
1	0	@user @user thanks for #lyft credit i can't us...
2	0	bihday your majesty
3	0	#model i love u take with u all the time in ...
4	0	factsguide: society now #motivation
...	...	...
31957	0	ate @user isz that youuu?ðŸ˜□ðŸ˜□ðŸ˜□ðŸ˜□ðŸ˜□ð...
31958	0	to see nina turner on the airwaves trying to...
31959	0	listening to sad songs on a monday morning otw...
31960	1	@user #sikh #temple vandalised in in #calgary,...
31961	0	thank you @user for you follow

31962 rows × 2 columns

Figure 3.4: Importing Packages



```
In [3]: #Check class distribution in dependent variable
        display(data['label'].value_counts().to_frame())
        print("0s : 1s  ::  ",(data['label'].value_counts()[0]/data['label'].value_counts()[1]).round(2),": 1")
        plt.bar([0,1],data['label'].value_counts())
        plt.title("class proportions in the dataset")
        plt.show()
```

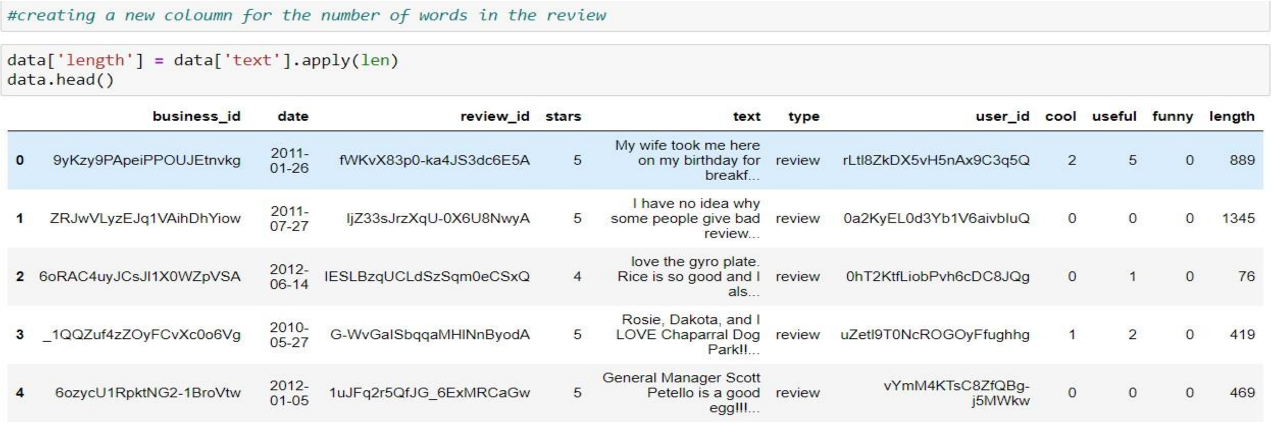
	label
0	29720
1	2242

```
0s : 1s  ::   13.26 : 1
```

Figure 3.5: Loading and Analyzing the Dataset

Adding a new column for the number of words in review

In this stage, we established a new column to enter the length or number of words in a review using the apply(len) function; the code for this is seen below:



```
#creating a new coloumn for the number of words in the review
```

```
data['length'] = data['text'].apply(len)
data.head()
```

	business_id	date	review_id	stars	text	type	user_id	cool	useful	funny	length
0	9yKzy9PApeiPPOUJEtnvkg	2011-01-26	fWKvX83p0-ka4JS3dc6E5A	5	My wife took me here on my birthday for breakf...	review	rLtl8ZkDX5vH5nAx9C3q5Q	2	5	0	889
1	ZRJwVLyzEJq1VAihDhYiow	2011-07-27	IjZ33sJrzXqU-0X6U8NwyA	5	I have no idea why some people give bad review...	review	0a2KyEL0d3Yb1V6aivbIuQ	0	0	0	1345
2	6oRAC4uyJCsJl1X0WZpVSA	2012-06-14	IESLBzqUCLdSzSqm0eCSxQ	4	love the gyro plate. Rice is so good and I als...	review	0hT2KtfLiobPvh6cDC8JQg	0	1	0	76
3	_1QQZuf4zZOyFCvXc0o6Vg	2010-05-27	G-WvGaISbqqaMHINnByodA	5	Rosie, Dakota, and I LOVE Chaparral Dog Park!!...	review	uZetl9T0NcROGOyFfughhg	1	2	0	419
4	6ozycU1RpktNG2-1BroVtw	2012-01-05	1uJFq2r5QfJG_6ExMRCaGw	5	General Manager Scott Petello is a good egg!!!...	review	vYmM4KTsC8ZfQBg-j5MWkw	0	0	0	469

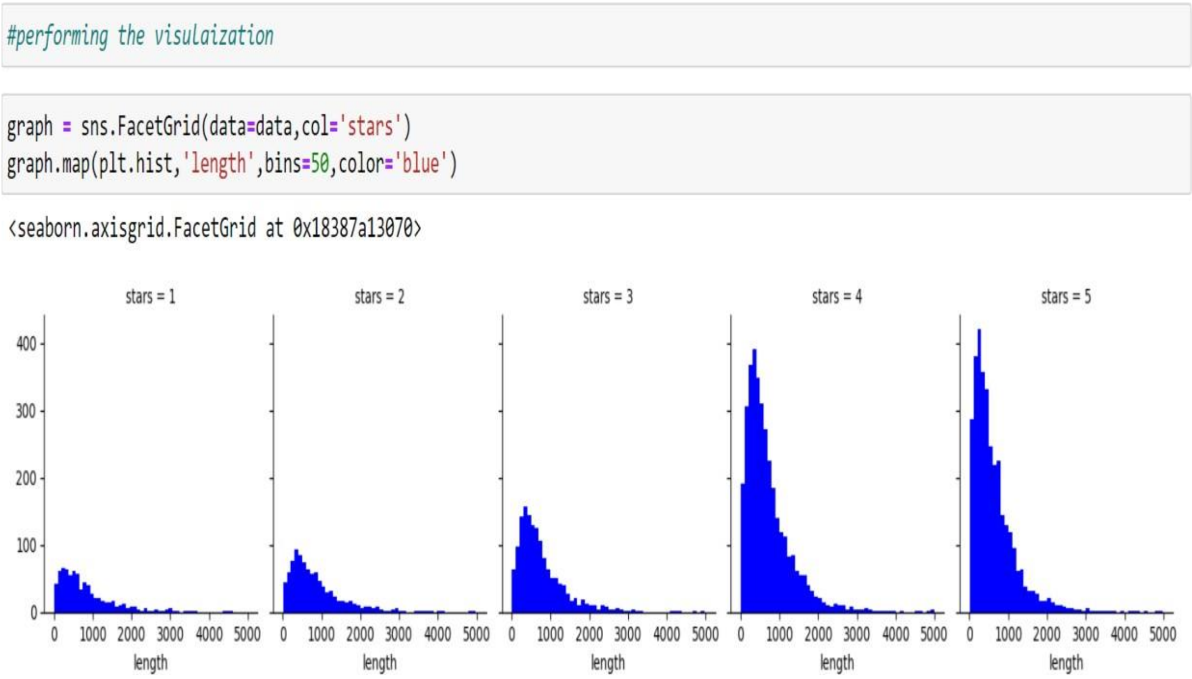
Figure 3.6: Creation of new column length

Visualisation

Using Seaborn, we plot the graph in this stage, which involves visualising the dataset to determine the relationship between the number of stars and the length of the reviews.

Applying the algorithms involves several steps, one of which is visualisation, which aids in a better understanding of the dataset and its features.

By creating a Histogram of the length and stars, we will be able to see the dataset.



```
#performing the visulaization
```

```
graph = sns.FacetGrid(data=data,col='stars')
graph.map(plt.hist,'length',bins=50,color='blue')
```

```
<seaborn.axisgrid.FacetGrid at 0x18387a13070>
```

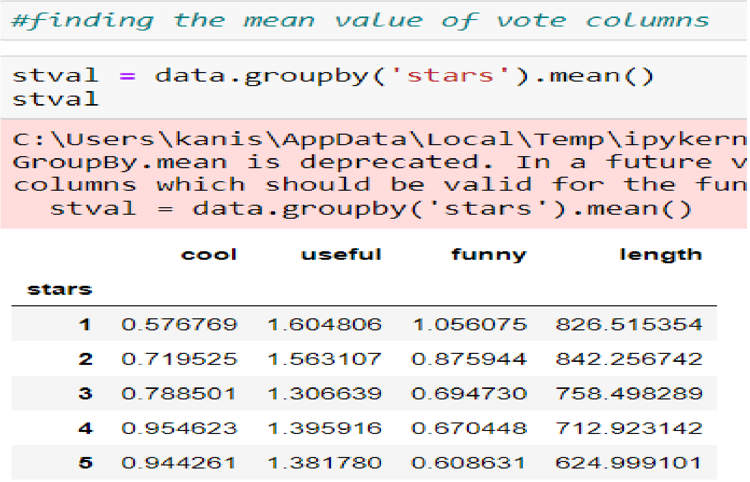
Figure 3.7: Visualization

➢ Finding mean values of the vote columns

There are three vote columns - Cool, Useful and Funny , so in this step we will find the mean values of the votes with respect to the starts given to the reviews.

Here we will be using the mean function

The code of the following is :



```
#finding the mean value of vote columns

stval = data.groupby('stars').mean()
stval
```

```
C:\Users\kanis\AppData\Local\Temp\ipykern
GroupBy.mean is deprecated. In a future v
columns which should be valid for the fun
  stval = data.groupby('stars').mean()
```

	cool	useful	funny	length
stars				
1	0.576769	1.604806	1.056075	826.515354
2	0.719525	1.563107	0.875944	842.256742
3	0.788501	1.306639	0.694730	758.498289
4	0.954623	1.395916	0.670448	712.923142
5	0.944261	1.381780	0.608631	624.999101

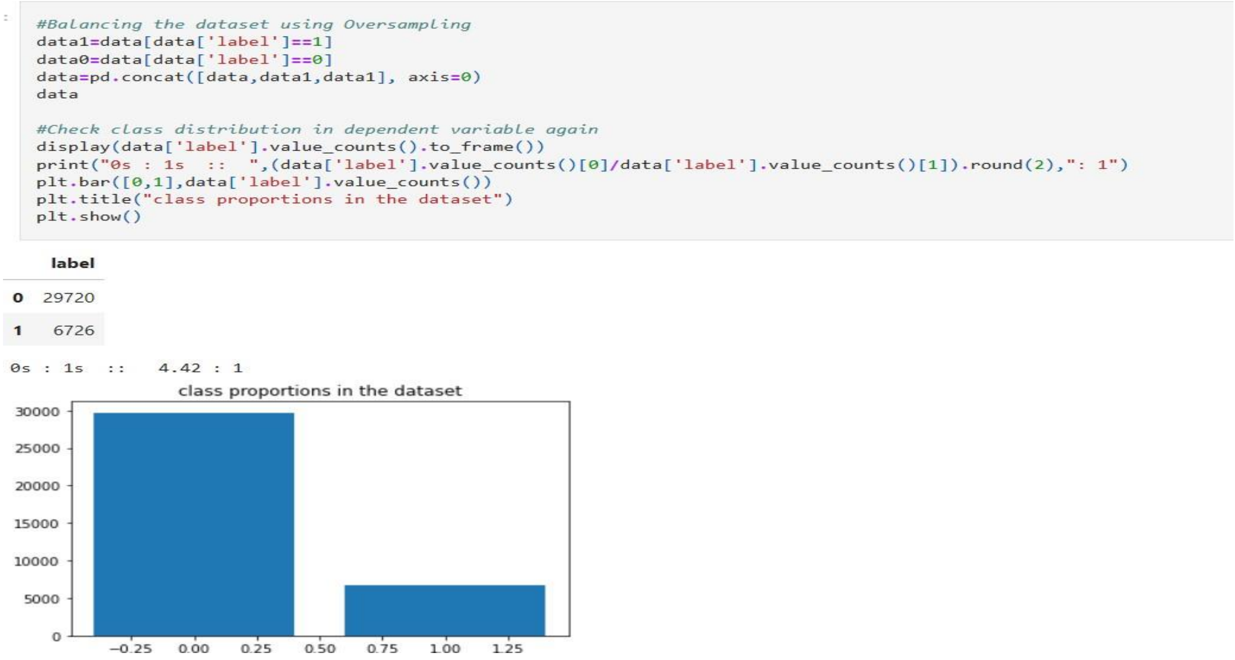
Figure 3.8: Mean Values of Vote

➢ Balancing the data and dividing it into ratings and stars

We will categorise the dataset and divide it into reviews and stars in this stage.

This is being done to aid in the more effective operation of the algorithms and obtain the best outcome from the dataset.

The following is coded as follows:



```
#Balancing the dataset using Oversampling
data1=data[data['label']==1]
data0=data[data['label']==0]
data=pd.concat([data,data1,data1], axis=0)
data

#Check class distribution in dependent variable again
display(data['label'].value_counts().to_frame())
print("0s : 1s  ::  ",(data['label'].value_counts()[0]/data['label'].value_counts()[1]).round(2),": 1")
plt.bar([0,1],data['label'].value_counts())
plt.title("class proportions in the dataset")
plt.show()
```

	label
0	29720
1	6726

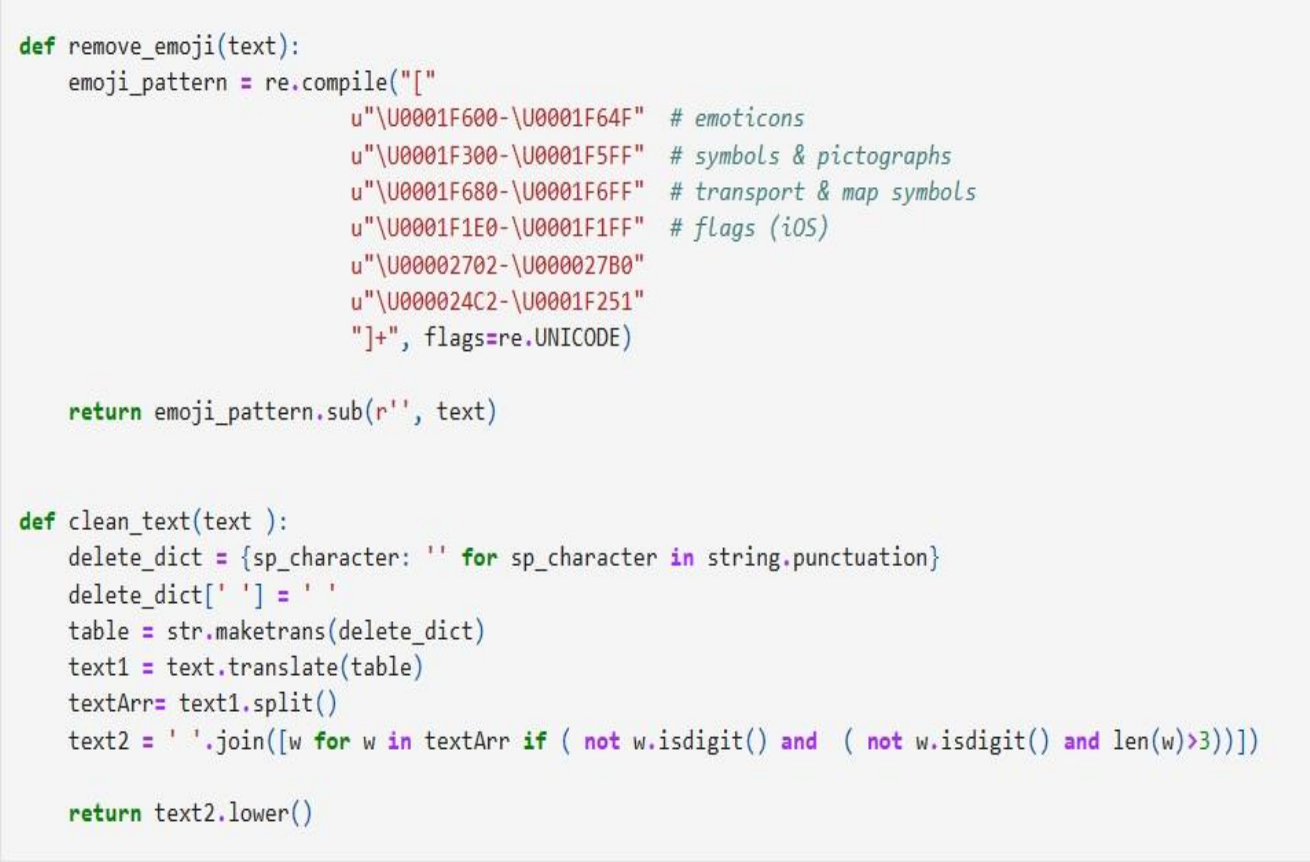
```
0s : 1s  ::    4.42 : 1
```

Figure 3.9: Balancing the datatset

Data Cleaning

It is the most crucial phase since the data that will be utilised to run the ML algorithms needs to be cleaned and free of stopwords like "the," "a," "an," "in," and similar words because they slow down processing.

Following is the code for the same:



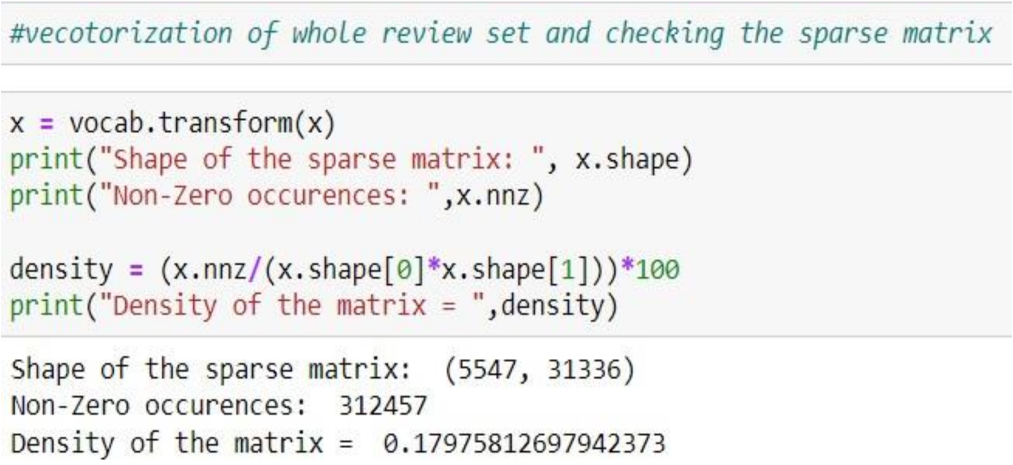
```
def remove_emoji(text):
    emoji_pattern = re.compile("["
                           u"\U0001F600-\U0001F64F"  # emoticons
                           u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                           u"\U0001F680-\U0001F6FF"  # transport & map symbols
                           u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                           u"\U00002702-\U000027B0"
                           u"\U000024C2-\U0001F251"
                           "]+", flags=re.UNICODE)

    return emoji_pattern.sub(r'', text)


def clean_text(text ):
    delete_dict = {sp_character: '' for sp_character in string.punctuation}
    delete_dict[' '] = ' '
    table = str.maketrans(delete_dict)
    text1 = text.translate(table)
    textArr= text1.split()
    text2 = ' '.join([w for w in textArr if ( not w.isdigit() and  ( not w.isdigit() and len(w)>3))])

    return text2.lower()
```

Figure 3.10: Data Cleaning

➢ Complete review set vectorization and sparse matrix verification

In this step, we vectorized the entire test set and checked the sparse matrix, which is defined as a matrix in which most of the elements are zero and it only contains data in a small number of positions. Additionally, the sparse matrix's memory usage is entirely made up of zeroes.

The code of the following is as shown below:



```
#vecotorization of whole review set and checking the sparse matrix

x = vocab.transform(x)
print("Shape of the sparse matrix: ", x.shape)
print("Non-Zero occurences: ",x.nnz)

density = (x.nnz/(x.shape[0]*x.shape[1]))*100
print("Density of the matrix = ",density)
```

```
Shape of the sparse matrix:  (5547, 31336)
Non-Zero occurences:  312457
Density of the matrix =  0.17975812697942373
```

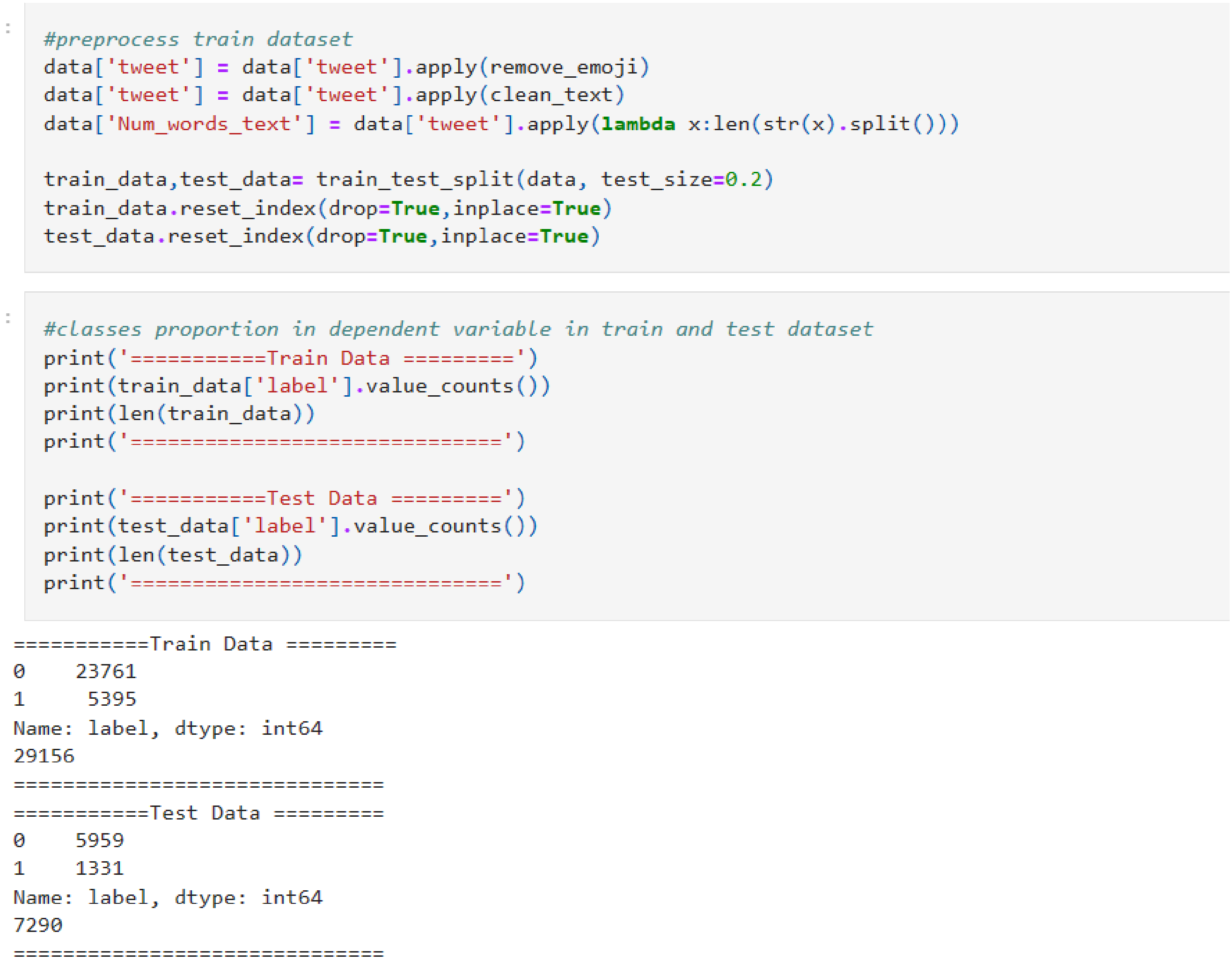
Figure 3.11: Vectorization and Checking of Sparse Matrix

- **Splitting the dataset into testing and training set**

If machine learning is to assure effective model computation, then it is crucial to train and design an algorithm to help anticipate the data. Typically, the data that experts gather is divided up into datasets that are then used for training and testing. Training, validation, and test sets are frequently included in these sets. To assess the overall effectiveness of ML algorithms, the approach may be used to generate informed judgements on de-identified data. The findings enable comparing the efficiency of ML algorithms to the complexity of predictive modelling, and it is a quick and simple approach to accomplish so. Even though the process is simple to use and compile, there are times when it is not worth using it right away, such as when the database is unstable when additional preparation is required, the database is tiny, or both. Initially, mtxlel was introduced to the training data after the model was created using a supervised learning approach. The training data is used by the current model or the model we created to produce the result, and based on the outcome, we can decide if the model was successful in foretelling the prices or not.

The procedure yields two each of the following values: X train, X test, Y train, and Y test. These values are then saved in an array, and the stored value will be employed in his analysis for amusement. Python's sklearn package is what we utilise for this.

The code for the following is shown below:



```
#preprocess train dataset
data['tweet'] = data['tweet'].apply(remove_emoji)
data['tweet'] = data['tweet'].apply(clean_text)
data['Num_words_text'] = data['tweet'].apply(lambda x:len(str(x).split()))

train_data,test_data= train_test_split(data, test_size=0.2)
train_data.reset_index(drop=True,inplace=True)
test_data.reset_index(drop=True,inplace=True)
```

```
#classes proportion in dependent variable in train and test dataset
print('===========Train Data =========')
print(train_data['label'].value_counts())
print(len(train_data))
print('==============================')

print('===========Test Data =========')
print(test_data['label'].value_counts())
print(len(test_data))
print('==============================')
```

```
===========Train Data =========
0    23761
1     5395
Name: label, dtype: int64
29156
==============================
===========Test Data =========
0    5959
1    1331
Name: label, dtype: int64
7290
==============================
```

Figure 3.12: Splitting the Dataset in Testing and Training set

Models Used:

We used six machine learning algorithms in our project that are as follows:

1. Recurrent neural network (RNN)
2. Long short-term memory (LSTM)
3. Decision Tree Classifier
4. Support Vector Machines
5. Gradient Boosting Classifier
6. K-Nearest Neighbour Classifier

Recurrent neural network (RNN)

A sort of artificial neural network called a recurrent neural network (RNN) uses sequential data or timeseries data. They are implemented into well-known programmes like Siri, voice search, and Google Translate. These deep learning algorithms are frequently employed for ordinal or temporal issues, such as language translation, natural language processing (nlp), speech recognition, picture captioning, and nlp. Recurrent neural networks (RNNs) learn from training data similarly to feedforward and convolutional neural networks (CNNs). They stand out because to their "memory," which allows them to affect the current input and output by using data from previous inputs. Recurrent neural networks' outputs depend on the previous items in the sequence, in contrast to standard deep neural networks' assumption that inputs and outputs are mutually exclusive.

Unidirectional recurrent neural networks are unable to anticipate future occurrences, even if they would be useful in predicting how a series would turn out.
Let's use a phrase that people frequently use to describe being sick—"feeling under the weather"—to help us better understand RNNs. It must be stated in that particular order for the idiom to make sense. Therefore, in order to predict the following word in the sequence, recurrent networks need to take into account the order in which each word appears in the idiom.

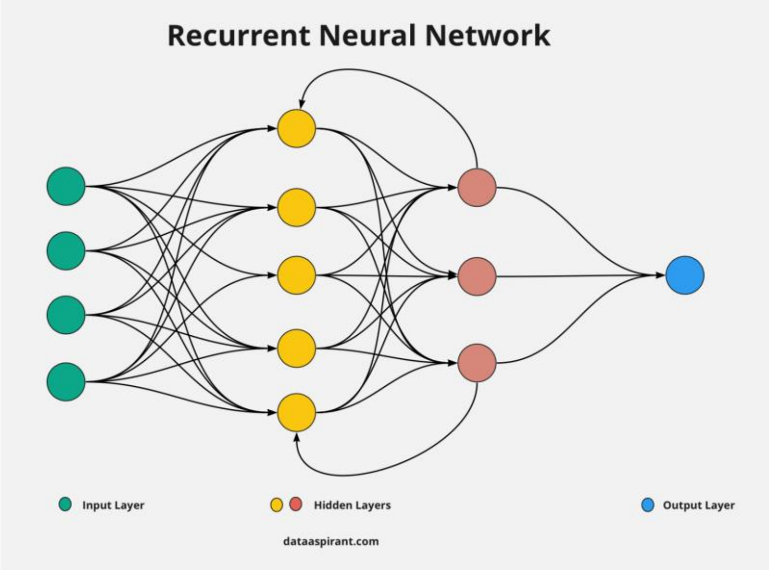


Figure 3.13: Flowchart of Recurrent Neural Network

- **Recurrent neural network benefits include:**

1. Over time, an RNN remembers every single bit of information. Only the ability to remember previous inputs makes it useful for time series prediction. Long Short Term Memory is the term for this.

2. To increase the effective pixel neighbourhood, convolutional layers and recurrent neural networks are combined.

- **Recurrent neural network disadvantages:**

1. Problems with gradient disappearing and exploding.
2. It is exceedingly tough to train an RNN.
3. If tanh or relu are used as the activation function, it cannot parse very lengthy sequences.

- **Recurrent neural network applications**

1. Text generation and language modelling
2. Speaking Recognition
3. Automatic Translation
4. Face identification and image recognition
5. Forecasting time series

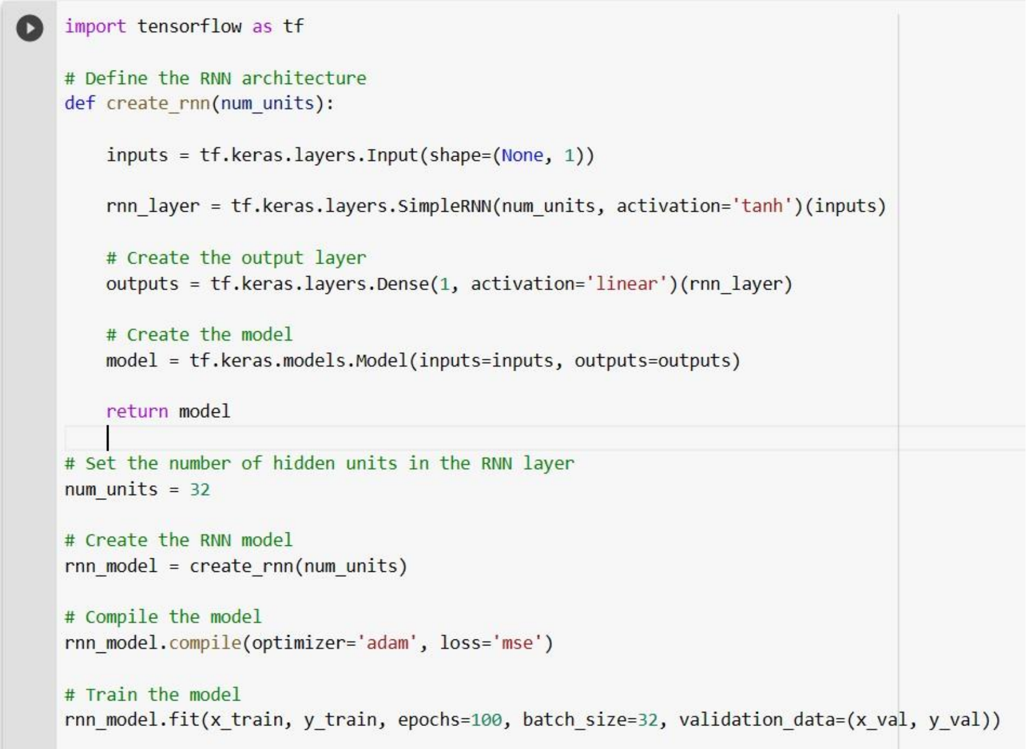
RNNs could act erratically. Analysis in these situations might make use of dynamical systems theory.

They are recursive neural networks, specifically ones with a linear chain-like structure. Recurrent neural networks work on the linear progression of time, fusing the representation for the previous time step and a hidden representation into the representation for the current time step, as opposed to recursive neural networks, which operate on any hierarchical structure by fusing child representations into parent representations.

RNNs can, for instance, take the form of a nonlinear autoregressive exogenous model (NARX), a nonlinear finite impulse response filter, or an infinite impulse response filter.[22]

When choosing a learning algorithm and scientific field (such as RNN, GAN, RL, CNN, etc.), a learning algorithm recommendation framework may be helpful.

The code we used is as follows:



```
import tensorflow as tf

# Define the RNN architecture
def create_rnn(num_units):

    inputs = tf.keras.layers.Input(shape=(None, 1))

    rnn_layer = tf.keras.layers.SimpleRNN(num_units, activation='tanh')(inputs)

    # Create the output layer
    outputs = tf.keras.layers.Dense(1, activation='linear')(rnn_layer)

    # Create the model
    model = tf.keras.models.Model(inputs=inputs, outputs=outputs)

    return model

# Set the number of hidden units in the RNN layer
num_units = 32

# Create the RNN model
rnn_model = create_rnn(num_units)

# Compile the model
rnn_model.compile(optimizer='adam', loss='mse')

# Train the model
rnn_model.fit(x_train, y_train, epochs=100, batch_size=32, validation_data=(x_val, y_val))
```

Figure 3.14: Code for Recurrent Neural Networks

Long short-term memory (LSTM)

Deep learning and artificial intelligence both employ the long short-term memory (LSTM)[1] artificial neural network. The LSTM features feedback connections, in contrast to conventional feed forward neural networks. Such a recurrent neural network (RNN) may analyse complete data sequences as well as single data points, such as pictures, voice, or video.

As of this feature, LSTM networks are excellent for data processing and prediction. For instance, LSTM may be used for applications like networked, unsegmented handwriting recognition, voice recognition, machine translation, speech activity detection, robot control, video games, healthcare, and more.[12]

An input gate, an output gate, a forget gate, and a cell make up a typical LSTM unit[14].[15] The three gates control how information enters and leaves the cell, and the cell may remember values for any length of time.

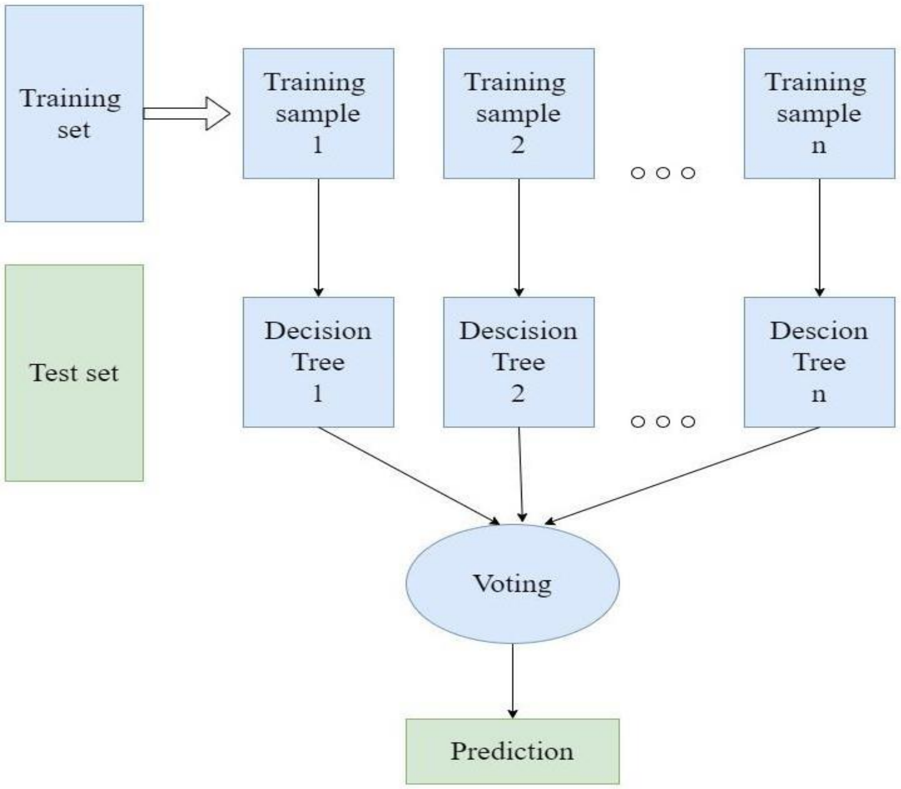


Figure 3.15: Flowchart of LSTM

- **Advantages:**

1. LSTM can complete tasks involving classification and regression.

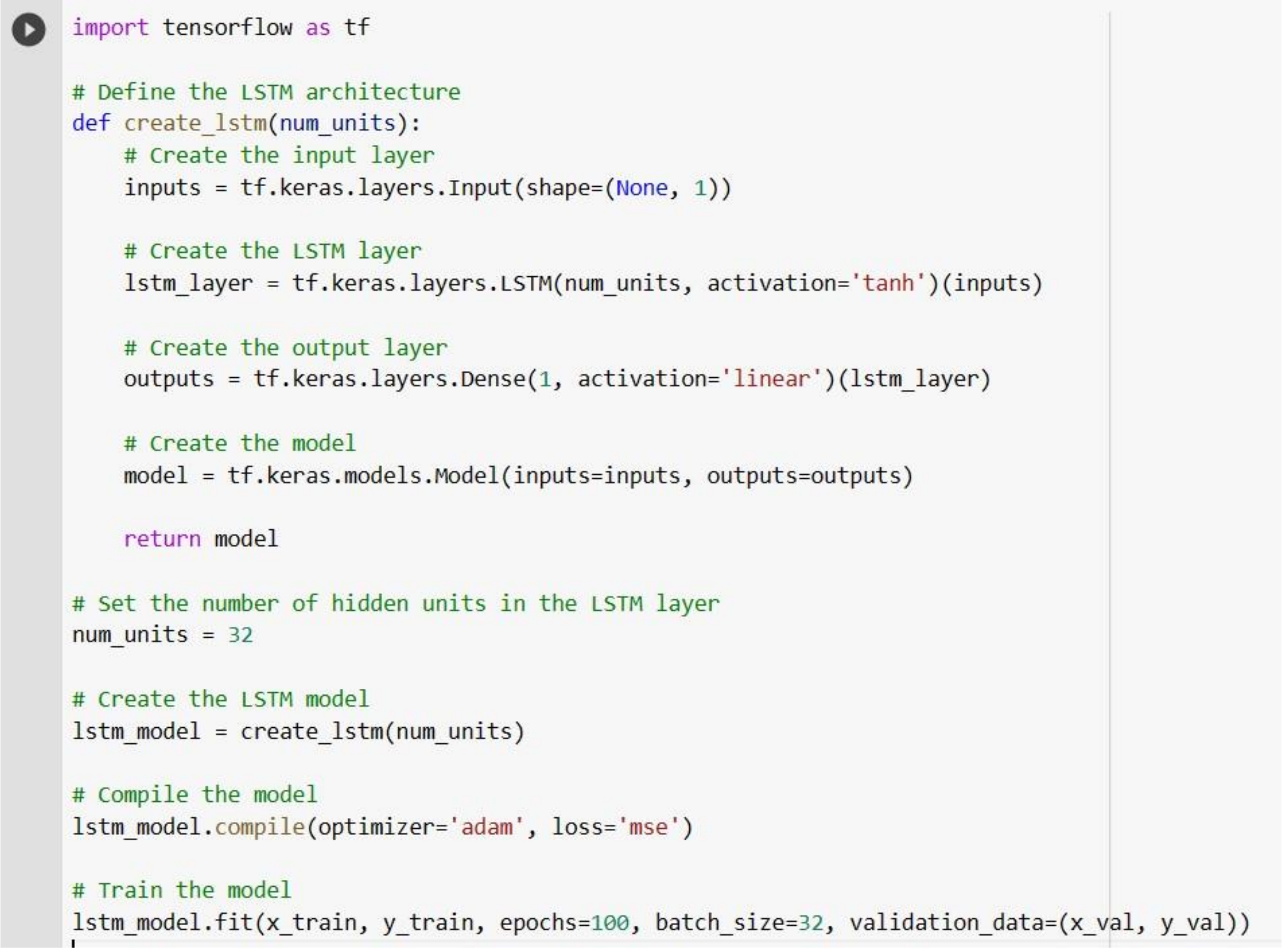
2. It has the ability to handle big datasets with many of dimensions.

3. It improves the model's accuracy and avoids the overfitting problem.

- **Disadvantages:**

1. It can be used for classification and regression tasks, although regression tasks are not appropriate for it.

2. We used the algorithm and the code for the following is as shown below:



```
import tensorflow as tf

# Define the LSTM architecture
def create_lstm(num_units):
    # Create the input layer
    inputs = tf.keras.layers.Input(shape=(None, 1))

    # Create the LSTM layer
    lstm_layer = tf.keras.layers.LSTM(num_units, activation='tanh')(inputs)

    # Create the output layer
    outputs = tf.keras.layers.Dense(1, activation='linear')(lstm_layer)

    # Create the model
    model = tf.keras.models.Model(inputs=inputs, outputs=outputs)

    return model

# Set the number of hidden units in the LSTM layer
num_units = 32

# Create the LSTM model
lstm_model = create_lstm(num_units)

# Compile the model
lstm_model.compile(optimizer='adam', loss='mse')

# Train the model
lstm_model.fit(x_train, y_train, epochs=100, batch_size=32, validation_data=(x_val, y_val))
```

Figure 3.16: Code for LSTM

Decision Tree Classifier

A decision tree is a supervised learning technique that may be used to address classification and regression issues, but is most frequently employed to address them. It is a tree-structured classifier, where internal nodes represent the dataset's characteristics, branches the rules for classifying objects, and individual leaf nodes the classification outcome. In a decision tree, a decision node and a leaf node are the only nodes. A choice is made by a decision node, which has many branches, but a decision's outcome is represented by a leaf node, which has no additional branches.

The transmitted data set's functionalities are utilised to run tests or reach choices. It is a graphical depiction that displays every scenario for resolving a dilemma or choosing a course of action based on particular criteria. It is known as a decision tree because it begins at the root node and grows into a tree-like structure through additional branches.

The CART algorithm, which stands for Classification and Regression Tree Algorithm, is used to construct the tree.

A decision tree only poses a question and subsequently subdivides the tree into subtrees based on the response (yes/no).

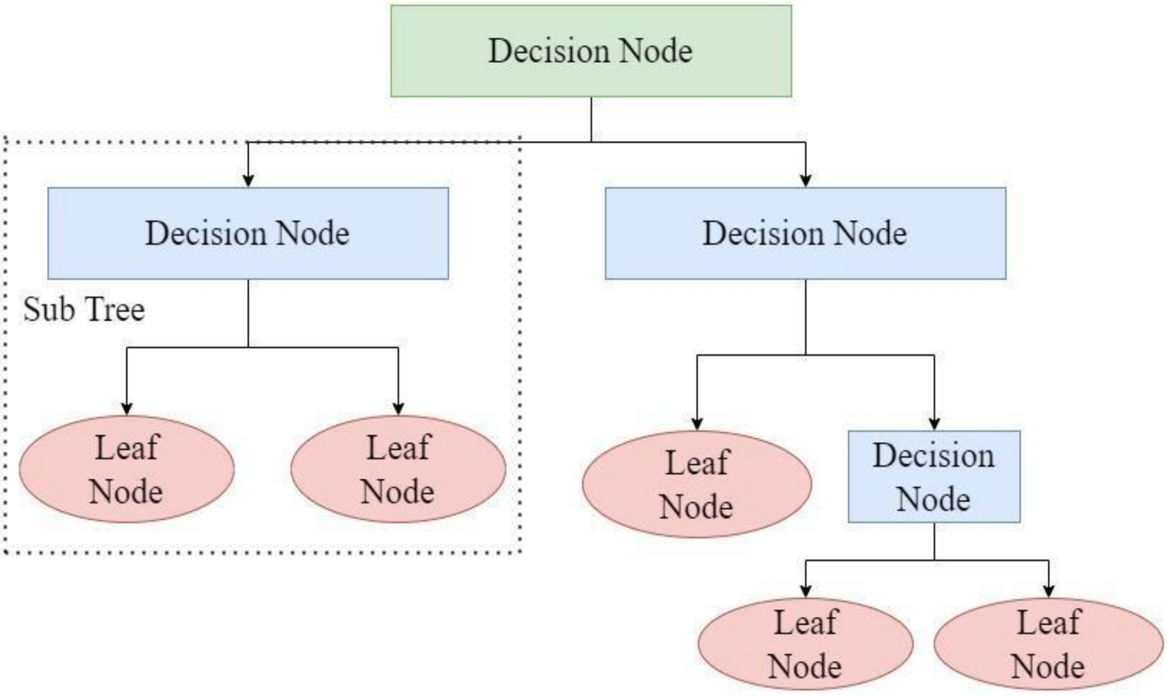


Figure 3.17: Flowchart of Decision Tree Classifier

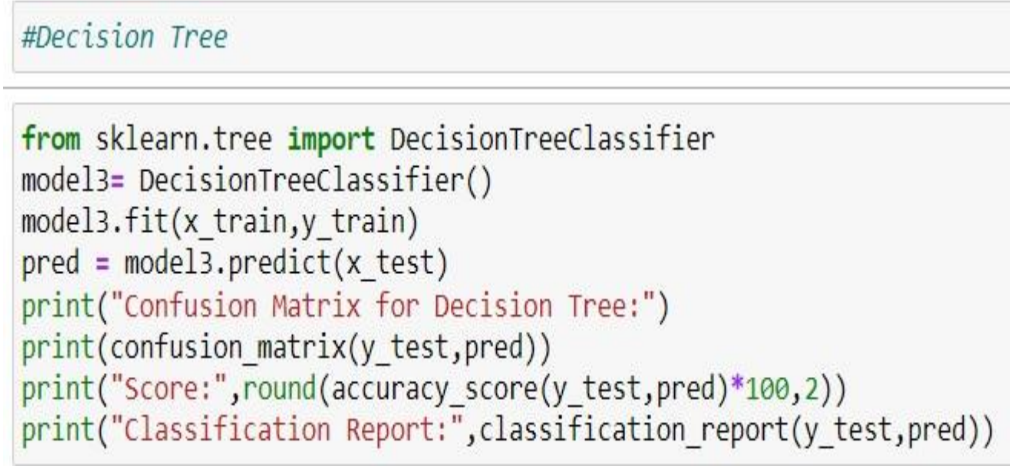
- **Advantages:**

1. It is easy to comprehend since it is based on how people normally make decisions.
2. It is highly helpful for resolving issues with decisions.
3. We find it useful to consider every scenario that might result from an issue.
4. Compared to other methods, it requires less data cleansing.

➢ **Disadvantages:**

1. It has several levels, making it complicated.
2. The Random Forest Algorithm may be used to address the Overfitting problem, which can occur at any time.
3. With more labels, the algorithm's computational complexity might rise.

The Code we used in our system is shown below:



```
#Decision Tree
```

```
from sklearn.tree import DecisionTreeClassifier
model3= DecisionTreeClassifier()
model3.fit(x_train,y_train)
pred = model3.predict(x_test)
print("Confusion Matrix for Decision Tree:")
print(confusion_matrix(y_test,pred))
print("Score:",round(accuracy_score(y_test,pred)*100,2))
print("Classification Report:",classification_report(y_test,pred))
```

Figure 3.18: Code for Decision Tree Classifier

Support Vector Machines

To address classification and regression issues, Support Vector Machine (SVM), one of the most popular supervised learning techniques, is utilised. The solution of classification issues is frequently employed in machine learning.

The SVM method's goal is to establish an ideal decision boundary or line that can categorise the n-dimensional space and enable rapid classification of fresh data points in the future. Also referred to as the hyperplane, this ideal decision boundary.

By choosing extreme points and vectors, SVM generates the hyperplane. SVM, the name of the algorithm, is the acronym for the support vectors connected with these extreme instances.

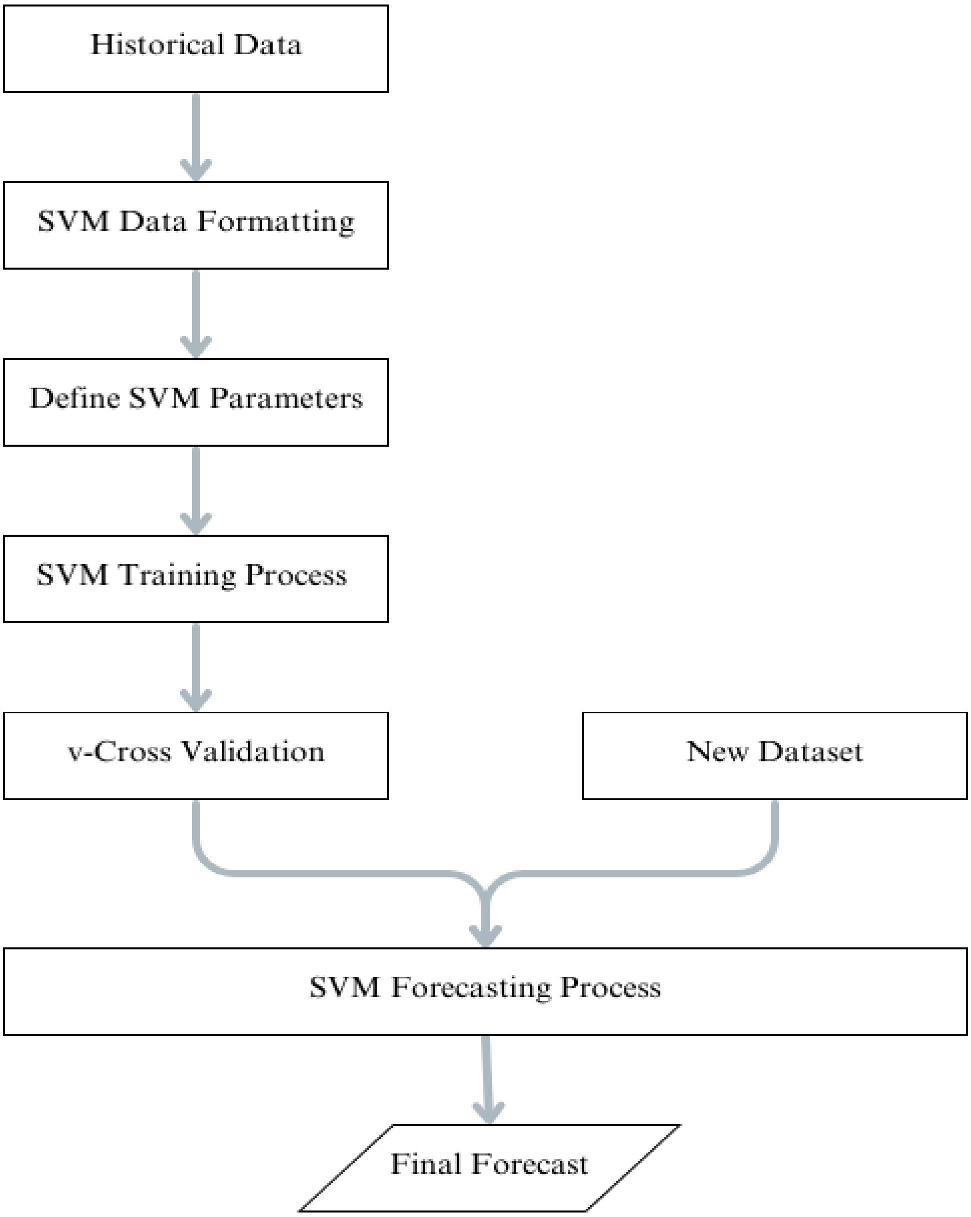


Figure 3.19: Flowchart of Support Vector Machine

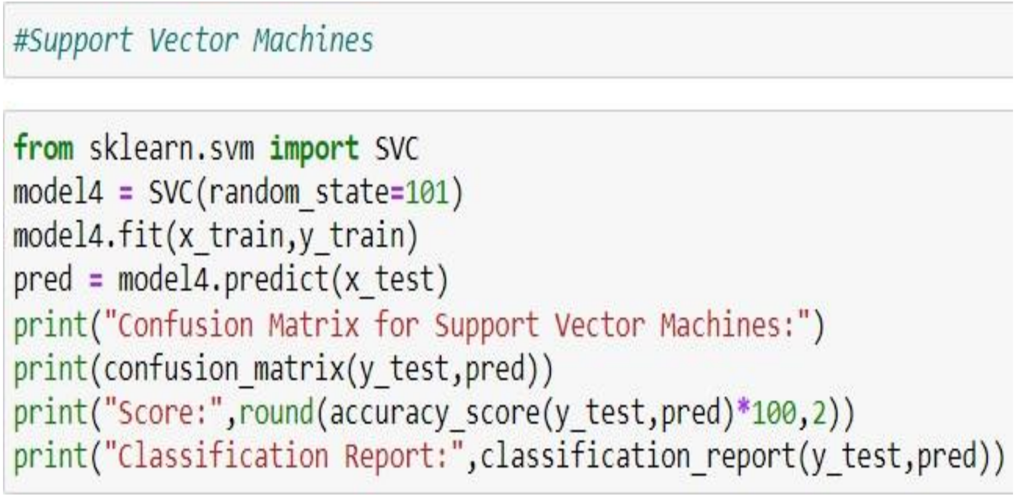
- **Advantages:**

1. In instances with high dimensions, it is particularly effective.

2. As only a portion of the training points, known as support vectors, are used in the decision, it is memory efficient.

3. Different kernel functions can be used for decision functions, and a custom kernel can also be specified.

The code we implemented is as follows:



```
#Support Vector Machines
```

```
from sklearn.svm import SVC
model4 = SVC(random_state=101)
model4.fit(x_train,y_train)
pred = model4.predict(x_test)
print("Confusion Matrix for Support Vector Machines:")
print(confusion_matrix(y_test,pred))
print("Score:",round(accuracy_score(y_test,pred)*100,2))
print("Classification Report:",classification_report(y_test,pred))
```

Figure 3.20: Code for Support Vector Machines

Gradient Boosting Classifier

For classification problems, a well-liked machine learning approach is gradient boosting classifier. It is based on the idea of ensemble learning, which combines several weak learners into one powerful learner, and is a member of the family of boosting algorithms.

The weak learners in gradient boosting are decision trees. In order to implement the method, decision trees are successively added to the ensemble, each one seeking to fix the mistakes produced by the preceding one. A certain number of trees must be added or the necessary degree of accuracy is reached before the procedure is complete.

The word "gradient" in the name alludes to the fact that the approach employs gradient descent optimisation to reduce the loss function while training the model. The gradient descent algorithm changes the model parameters in the direction of the loss function's steepest descent in order to minimise it. The loss function assesses the difference between the predicted and actual values of the target variable.

Overall, Gradient Boosting Classifier is a strong and popular technique for classification problems, noted for its capacity to manage complicated datasets with high-dimensional feature spaces and achieve high accuracy.

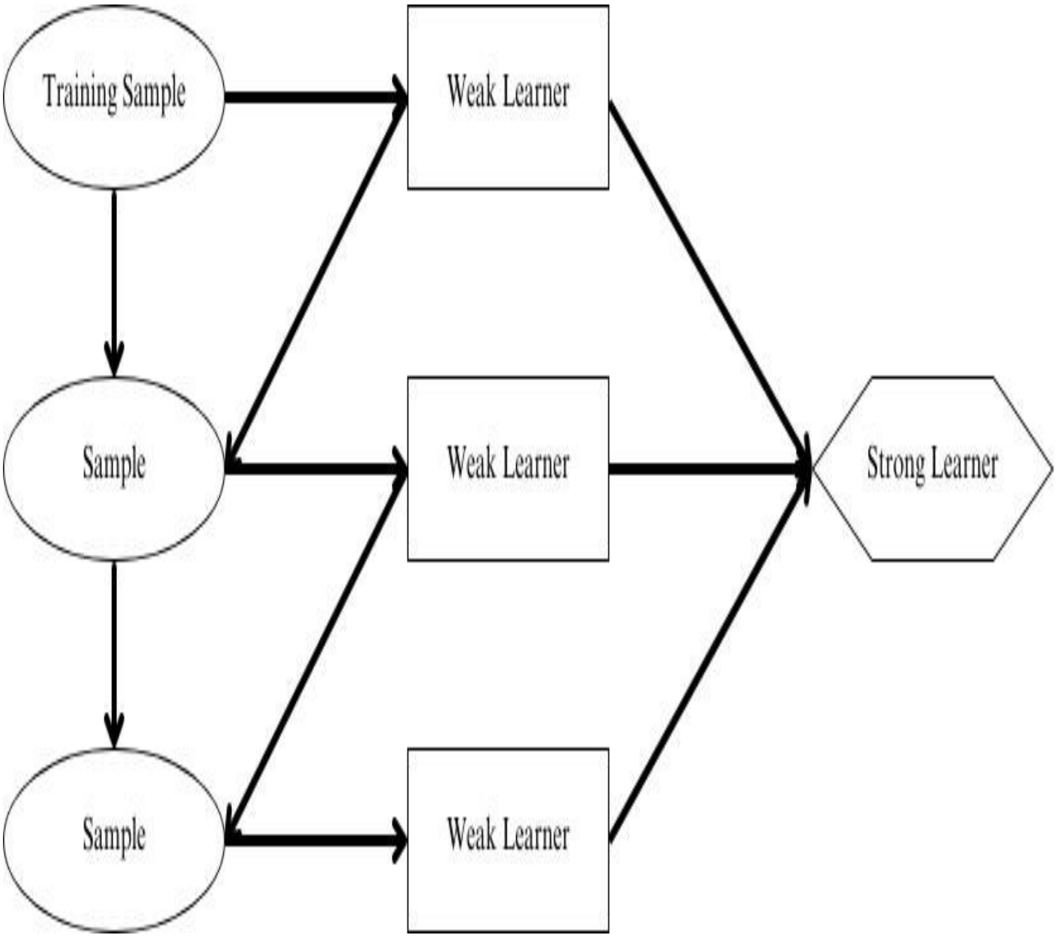


Figure 3.21: Flowchart of Gradient Boosting Classifier

Chapter 4

Results and Analysis

Results

Different neural-network based hybrid deep learning models, in conjunction with three distinct word embeddings, must be applied in order to examine the problem from a variety of perspec- tives and achieve the best possible performance on the work. The results of these different deep learning models are shown in Table 4.1,4.2 and 4.3.

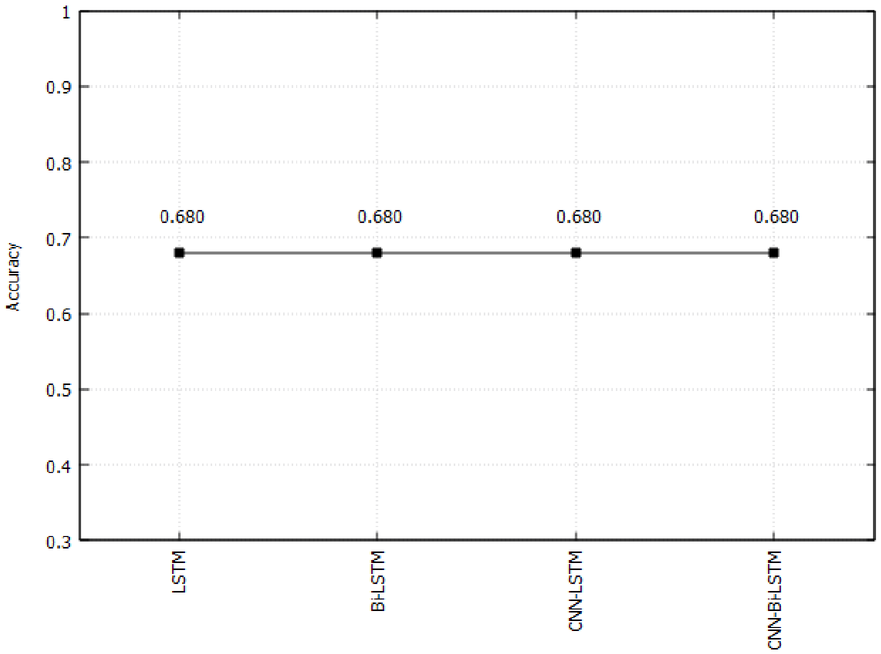


Figure 4.1: Performance of different models using glove word-embedding.

The accuracy of these deep learning models using word embeddings i.e glove,word2vec and fastext is given in Figure 4.1,4.2,and 4.3. It was observed that CNN-Bi-LSTM using word2vec method performs well in terms of precision. Regarding recall, Bi-LSTM with glove outper-

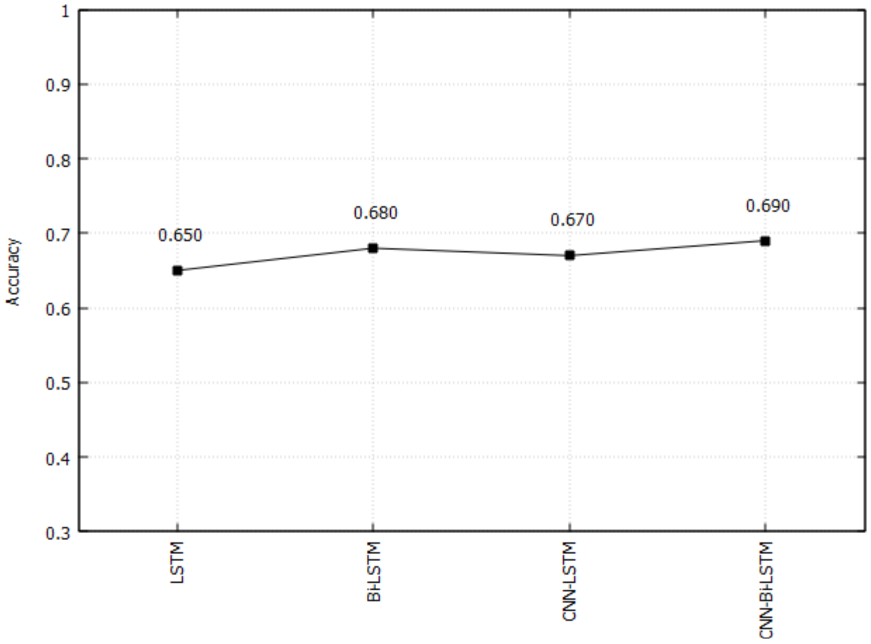


Figure 4.2: Performance of different models using word2vec word-embedding.

forms other models with the highest recall of 0.770. The performs of the model CNN-Bi-LSTM regarding accuracy, F1-score and ROC-AUC metrics is seen highest with a value of 0.720, 0.690 and 0.760 respectively . Overall, it appears that CNN-Bi-LSTM with fastext as a word embedding seems to be the most promising among the various deep learning techniques, with strong performance in the majority of evaluation criteria.

Table 4.1: Performance of different models using glove word-embedding.

S.No.	**Model**	**Acc.**	**Prec.**	**Recall**	**ROC-AUC Score**	**F1-Score**
1	LSTM	0.680	0.650	0.720	0.750	0.690
2	Bi-LSTM	0.680	0.650	0.770	0.750	0.700
3	CNN-LSTM	0.680	0.680	0.670	0.750	0.680
4	CNN-Bi-LSTM	0.680	0.650	0.750	0.750	0.710

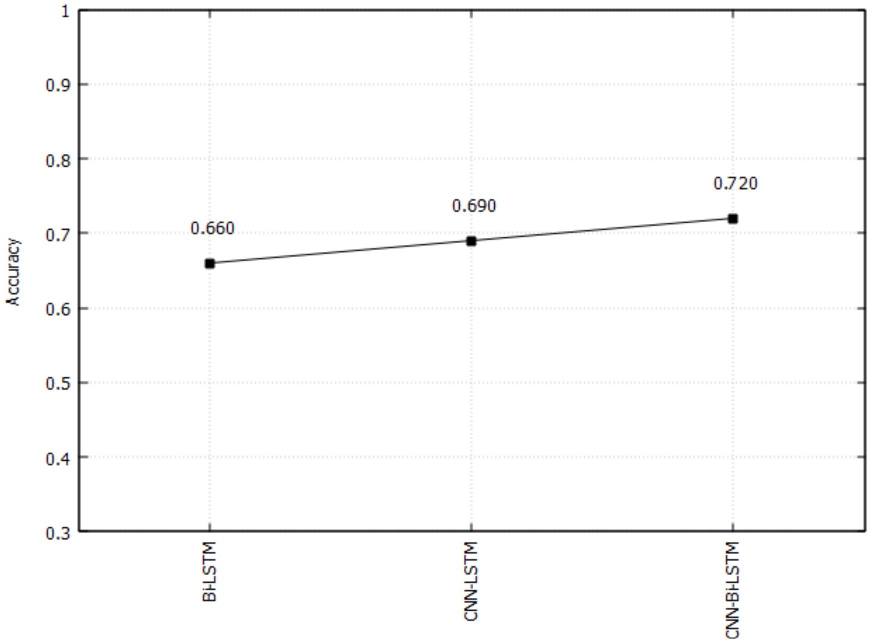


Figure 4.3: Performance of different models using fastext word-embedding.

Table 4.2: Performance of different models using Word2vec word-embedding.

S.No.	**Model**	**Acc.**	**Prec.**	**Recall**	**ROC-AUC Score**	**F1-Score**
1	LSTM	0.650	0.670	0.650	0.720	0.660
2	Bi-LSTM	0.680	0.680	0.680	0.750	0.680
3	CNN-LSTM	0.670	0.710	0.630	0.740	0.670
4	CNN-Bi-LSTM	0.690	0.720	0.630	0.740	0.670

Table 4.3: Performance of different models on Fastext word-embedding

S.No.	**Model**	**Acc.**	**Prec.**	**Recall**	**ROC-AUC Score**	**F1-Score**
1	LSTM	–	–	–	–	–
2	Bi-LSTM	0.660	0.680	0.670	0.740	0.690
3	CNN-LSTM	0.690	0.710	0.680	0.750	0.670
4	**CNN-Bi-LSTM**	**0.720**	**0.690**	**0.690**	**0.760**	**0.690**

Comparison with other Researchers

At this point in our project, we have probably read a fair number of research articles regarding hate speech detection. Many studies have been carried out on this topic. we have compared some of those studies with our proposed work i.e in 2018 [7] the author used SVM and random forest models on a bohra-2018 dataset to get the accuracy of (0.710) and (0.667).In the year 2019 [11] the author used a hierarchical LSTM model with attention based on phonemic sub-words on two different datasets i.e bohra-2018 and HEOT to get the accuracy of (0.666) and (0.630) respectively.

In this study, different neural network-based models and hybrid models using different word embeddings from which CNN-Bilateral LSTM using fastext as word embedding outperformed other models and gave us the accuracy of 0.720, precision of 0.690, recall of 0.690 , and F1 - score of 0.720 on a dataset which was made by merging 3 different datasets i.e bohra-2018,hasoc-2021 and kumar-2018 containing a total of 21 thousands rows.

Table 4.4: Comparison of proposed work with other researchers.

S.No.	Year	Model	Dtatset	Acc.	Prec.	Recall	F1-score
1	2018 [7]	SVM	4575 code-mixed tweets-HS	0.717	–	–	0.620
2	2018 [7]	Random Forest	4575 code-mixed tweets-HS	0.667	–	–	–
3	2019 [11]	Hierarchical LSTM model	HS(Bohra el al)-3800 tweets	0.666	–	0.451	0.487
4	2019 [11]	Hierarchical LSTM model	3679(Hindi manually created) -HEOT	0.630	–	–	0.520
5	**This study**	**CNN-Bi-LSTM**	**Created our datset by merging 3 different datasets** **Bohra-2018** **Hasoc-2021-eng-hin** **Kumar-2018**	**0.720**	**0.690**	**0.690**	**0.690**

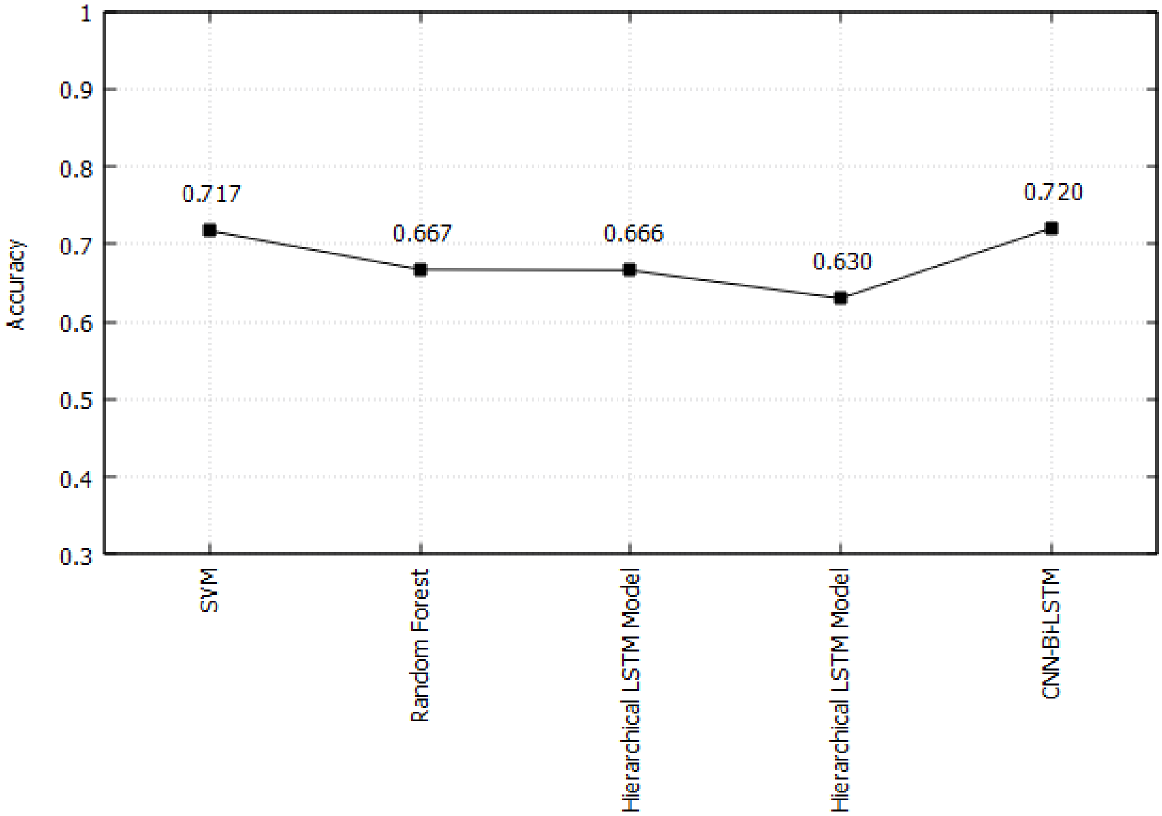


Figure 4.4: Comparison of proposed work with other researchers.

Chapter 5

Conclusion and Future Work

Conclusion

In this work, the most important step was to convert multilingual data to a uniform code. Our dataset contained English, Hindi (Devanagari), and Hinglish tweets. We tried to convert these tweets into English tweets by using translation, dictionaries, etc.We have also handled emojis,hashtags, etc. Apart from this, we have assessed several deep learning models for the task of hate speech detection. The present study made use of Word2Vec, Glove, and FastText word embeddings and models like long short term memory, Bidirectional long short term memory, etc to arrive at the model that is being proposed. In our experiment, CNN-BiLSTM outperforms other models in terms of accuracy (0.72) and roc-auc-score (0.76).

Future Work

We have achieved an accuracy of 72 percent, but this can be increased by converting Hinglish tweets to English in a more precise manner.We can use some global dictionary. In future work, we will try to make a robust system to convert Hinglish data to English.

References

[1] Council of europe. *Available from: https://www.coe.int/en/web/portal/home.*

[2] *Available from: https://backlinko.com/instagram-users.*, 2020.

[3] India social media statistics 2021. *The Global Statistics. Available from: https://www.theglobalstatistics.com/india-social-media-statistics/.*, 2021 Dec.

[4] Safa Alsafari, Samira Sadaoui, and Malek Mouhoub. Hate and offensive speech detection on arabic social media. *Online Social Networks and Media*, 19:100096, 2020.

[5] Jatin Sharma Monojit Choudhury Bali, Kalika and Yogarshi Vyas. "“i am borrowing ya mixing?” an analysis of english-hindi code mixing in facebook.”. *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126, 2014.

[6] Shubham Bharti, Arun Kumar Yadav, Mohit Kumar, and Divakar Yadav. Cyberbullying detection from tweets using deep learning. *Kybernetes*, 2021.

[7] Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivas - tava. A dataset of hindi-english code-mixed social media text for hate speech detection. In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pages 36–41, 2018.

[8] Singh V Akhtar SS Shrivastava M. Bohra A, Vijay D. A dataset of hindi-english code-mixed social media text for hate speech detection. *In: Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pages 36–41, 2018.

[9] S. Burch. Youtube deletes 500 million comments in fight against ‘hate speech’. thewrap. *https://www.thewrap.com/youtube-deletes-500-millioncomments-in- fight-against-hate-speech/*, 2019, September 3.

[10] J. Cement. Social media: active usage penetration in selected countries 2020. *Retrieved from https://www.statista.com/statistics/282846/regular-socialnetworking-usage-penetration-worldwide-by-country/*, 2020, February 14.

[11] Shivang Chopra, Ramit Sawhney, Puneet Mathur, and Rajiv Ratn Shah. Hindi-english hate speech detection: Author profiling, debiasing, and practical perspectives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 386–393, 2020.

[12] Michele Di Capua, Emanuel Di Nardo, and Alfredo Petrosino. Unsupervised cyber bullying detection in social networks. In *2016 23rd International conference on pattern recognition (ICPR),* pages 432–437. IEEE, 2016.

[13] Wikimedia Foundation. Hate speech. *Available from: https://en.wikipedia.org/w/index.php?title=Hate$_s$peecholdid* = 1059042962*.,* 2021.

[14] McNamara L. Gelber, K. Evidencing the harms of hate speech. social identities. *Annals of the Romanian Society for Cell Biology,* 22(3):324–341, 2016.

[15] Satyajit Kamble and Aditya Joshi. Hate speech detection from code-mixed hindi-english tweets using deep learning models. *arXiv preprint arXiv:1811.05145,* 2018.

[16] Ritesh Kumar, Aishwarya N Reganti, Akshit Bhatia, and Tushar Maheshwari. Aggression-annotated corpus of hindi-english code-mixed data. *arXiv preprint arXiv:1803.09402,* 2018.

[17] Suraj Kumar, Dipesh Kumar, Lalit Kumar, Kapil Kumar, Sandeep Kumar Maurya, Mohit Kumar, Divakar Yadav, et al. Fake news detection using hybrid deep learning method. 2022.

[18] Tao Li, Lei Lin, Minsoo Choi, Kaiming Fu, Siyuan Gong, and Jian Wang. Youtube av 50k: an annotated corpus for comments in autonomous vehicles. In *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP),* pages 1–5. IEEE, 2018.

[19] Shahi GK Madhu H Satapara S Majumder P et al. Mandl T, Modha S. Hate speech and offensive content identification in english and indo-aryan languages. *Annals of the Romanian Society for Cell Biology*, page arXiv preprint arXiv:211209301., 2021.

[20] Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26, 2018.

[21] I. Mehta. Twitter sees 900% increase in hate speech towards china because coronavirus. *The Next Web. https://thenextweb.com/world/2020/03/27/twittersees-900-increase-in-hate-speech-towards-china-because-coronavirus/*, 2020, March 27.

[22] Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. Misogyny detection in twitter: a multilingual and cross-domain study. *Information Processing & Management*, 57(6):102360, 2020.

[23] Endang Wahyu Pamungkas and Viviana Patti. Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop*, pages 363–370, 2019.

[24] Kumar Ravi and Vadlamani Ravi. Sentiment classification of hinglish text. In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)* , pages 641–645. IEEE, 2016.

[25] Hugo Rosa, David Matos, Ricardo Ribeiro, Luisa Coheur, and João P Carvalho. A “deeper” look at detecting cyberbullying in social networks. In *2018 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2018.

[26] T Tulasi Sasidhar, B Premjith, and KP Soman. Emotion detection in hinglish (hindi+ english) code-mixed social media text. *Procedia Computer Science*, 171:1346–1352, 2020.

[27] Sonali Rajesh Shah and Abhishek Kaushik. Sentiment analysis on indian indigenous languages: a review on multilingual opinion mining. *arXiv preprint arXiv:1911.12848*, 2019.

[28] Pranaydeep Singh and Els Lefever. Sentiment analysis for hinglish code-mixed tweets by means of cross-lingual word embeddings. In *LREC 2020–4th Workshop on Computational Approaches to Code Switching*, pages 45–51. European Language Resources Association (ELRA), 2020.

[29] Özaslan T Pfrommer B Kumar V Daniilidis K. Zhu AZ, Thakur D. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters.*, 3(3):2032–9, 2018.

6 may 2023

ORIGINALITY REPORT

16%	11%	11%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	link.springer.com Internet Source	2%
2	arun kumar yadav, Suraj Kumar, Dipesh Kumar, Lalit Kumar, Kapil Kumar, Sandeep Kumar Maurya, Mohit Kumar, Divakar Yadav. "Fake News Detection using Hybrid Deep Learning Method", Institute of Electrical and Electronics Engineers (IEEE), 2022 Publication	1%
3	Submitted to nith Student Paper	1%
4	Submitted to National Institute of Technology, Rourkela Student Paper	1%
5	"Advances in Computational Collective Intelligence", Springer Science and Business Media LLC, 2022 Publication	1%
6	Submitted to Management & Science University Student Paper	1%

7	www.researchgate.net Internet Source	1%
8	Submitted to Aligarh Muslim University, Aligarh Student Paper	1%
9	docplayer.net Internet Source	1%
10	en.wikipedia.org Internet Source	<1%
11	Arun Kumar Yadav, Mohit Kumar, Abhishek Kumar, Shivani, Kusum, Divakar Yadav. "Hate speech recognition in multilingual text: hinglish documents", International Journal of Information Technology, 2023 Publication	<1%
12	K Sreelakshmi, B Premjith, K.P. Soman. "Detection of Hate Speech Text in Hindi-English Code-mixed Data", Procedia Computer Science, 2020 Publication	<1%
13	conference.ase.ro Internet Source	<1%
14	Mangala Shetty, Deekshitha, Manisha Bhat, Manisha Devadiga. "Detection of Alzheimer's Disease Using Machine Learning", 2022	<1%

	International Conference on Artificial Intelligence and Data Engineering (AIDE), 2022 Publication	
15	arxiv.org Internet Source	<1 %
16	vidwan.inflibnet.ac.in Internet Source	<1 %
17	Syed Rameel Ahmad, Deborah Harris, Ibrahim Sahibzada. "Understanding Legal Documents: Classification of Rhetorical Role of Sentences Using Deep Learning and Natural Language Processing", 2020 IEEE 14th International Conference on Semantic Computing (ICSC), 2020 Publication	<1 %
18	Submitted to University of Southampton Student Paper	<1 %
19	www.mdpi.com Internet Source	<1 %
20	dokumen.pub Internet Source	<1 %
21	diva-portal.org Internet Source	<1 %
22	www.coursehero.com Internet Source	<1 %
	www.gnedenko.net	

23	Internet Source	<1%
24	Submitted to National Research University Higher School of Economics Student Paper	<1%
25	Submitted to Liverpool Hope Student Paper	<1%
26	Submitted to University of Strathclyde Student Paper	<1%
27	Submitted to University of South Africa Student Paper	<1%
28	www.datasciencecentral.com Internet Source	<1%
29	Submitted to Coventry University Student Paper	<1%
30	machinelearningmastery.com Internet Source	<1%
31	patents.google.com Internet Source	<1%
32	www.frontiersin.org Internet Source	<1%
33	www.irjmets.com Internet Source	<1%
34	yvm2020.authorea.com Internet Source	

		<1%
35	Abdulqahar Mukhtar Abubakar, Deepa Gupta, Suja Palaniswamy. "Explainable Emotion Recognition from Tweets using Deep Learning and Word Embedding Models", 2022 IEEE 19th India Council International Conference (INDICON), 2022 Publication	<1%
36	Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, Jianfeng Gao. "Deep Learning--based Text Classification", ACM Computing Surveys, 2021 Publication	<1%
37	Submitted to University of Sydney Student Paper	<1%
38	aclweb.org Internet Source	<1%
39	www.jiit.ac.in Internet Source	<1%
40	www.slideshare.net Internet Source	<1%
41	"Computational Intelligence", Springer Science and Business Media LLC, 2023 Publication	<1%

42	Shankar Biradar, Sunil Saumya, Arun chauhan. "Fighting hate speech from bilingual hinglish speaker's perspective, a transformer- and translation-based approach.", Social Network Analysis and Mining, 2022 Publication	<1%
43	drops.dagstuhl.de Internet Source	<1%
44	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
45	era.ed.ac.uk Internet Source	<1%
46	pdfcoffee.com Internet Source	<1%
47	peer.asee.org Internet Source	<1%
48	repositorio.uam.es Internet Source	<1%
49	repositorium.sdum.uminho.pt Internet Source	<1%
50	www.ijitee.org Internet Source	<1%
51	www.lrec-conf.org Internet Source	<1%

52	www.preprints.org Internet Source	<1 %
53	"Innovations in Computer Science and Engineering", Springer Science and Business Media LLC, 2021 Publication	<1 %
54	Femi Emmanuel Ayo, Olusegun Folorunso, Friday Thomas Ibharalu, Idowu Ademola Osinuga. "Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions", Computer Science Review, 2020 Publication	<1 %
55	Abhishek Chopra, Deepak Kumar Sharma, Aashna Jha, Uttam Ghosh. "A Framework for Online Hate Speech Detection on Code Mixed Hindi-English Text and Hindi Text in Devanagari", ACM Transactions on Asian and Low-Resource Language Information Processing, 2022 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		