DEPRESSION DETECTION USING MACHINE LEARNING

Udit Mohanty, Pratyasha Samal

Abstract- **Internet is considered as a major storehouse of information in today's world. No single work can be done without help. It has become one of the major ways of communication. Out of all communication methods available, social media is one of the most common forms. Social media has changed the world. The number of internet users worldwide in 2019 is 4.388 billion, up 9.1% year-on-year. The number of social media users worldwide in 2019 is 3.484 billion, up 9% year-on-year. People have a need to fit in to be visible praised in this new digital world. Social Media has many ill effects such as Cyber-bullying, Harassment and Depression. It is an almost undisputable truth that access to a cloak of anonymity and a large, large microphone brings the worst out in some people. Today, this is particularly true on social media. Online harassment and hate- messaging is a growing trend and a growing security concern for anyone on social media today. In fact, 40% of Internet users say they have personally experienced online harassment. Researchers have found that the longer you spend on social media, the more likely you are to be depressed. Hence, we have come up with this project in which we will be developing a system that helps in detecting a depressive and harassment text.**

Keywords- **Online harassment, Depression, Cyber-bullying, social media, Internet**

INTRODUCTION

Social Media has many ill effects such as Cyber-bullying, Harassment and Depression.

Evidence is mounting that there is a link between social media and depression. It is an almost un-disputable truth that access to a cloak of anonymity and a large, large microphone brings the worst out in some people.

Today, this is particularly true on social media. Online harassment and hate-messaging is a growing trend and a growing security concern for anyone on social media today.

In fact, 40% of Internet users say they have personally experienced online harassment.

Scope

Due to the increasing popularity of communications occurring in social networks, an inconsiderable part of young people's interpersonal interactions is performed online. New risks and threats emerge through the introduction of novel communication media, such as cyber bullying, -stalking and online grooming. Bullying, violence, and depression are the cornerstones of rising suicide worldwide. Therefore, this project seeks to build a method for determining whether or not a given post is suicidal or harassing.

Problem Statement

Due to lack of proper systems for detecting depression and harassment, people nowadays are harassed on social media. People expression depression are also not identified and hence they get lonelier and worse. The prime impediment of implementing a Depression and Harassment Detection on Social Media System is the social and psychological help of the individuals themselves.

Objectives

(1) The proposed system aims at providing automation for detecting depression and harassment on social media platforms.

(2) The system will assist the authorities with certain features added in tackling the problem of depression and harassment that is going on in social media through alert.

LITERATURE STUDY

Existing system

We have surveyed various papers and systems which have implemented or researched on text classification, natural language processing, depression or mood or harassment detection and assessment and discussed this below. This will help one gain a comprehensive idea of all the different types of system proposed or developed and can help one develop an efficient system based on previously proposed or developed systems.

Mood Path: A Depression

Mood path offers an assessment of mental health that can help you make the appeal for judgment to explore professional treatment. Unlike other assessments that ask you to reflect on the past two weeks, Mood path asks you questions about your emotional well-being at the moment over a series of 14 days. The app is designed to facilitate conversations with a professional, but more than 150 exercises and tools can also be found within the app to work on your mental health.

SURVEY

Sno	Name of the Paper	Name /Authors Of The Paper	Publication Date/Year	Areas Of Work	Findings
1	Facebook Social Media for Depression Detection in the Thai Community	Kantinee Katchapakirin, Konlakorn Wongpatikaseree, Panida Yomaboot, Yongyos Kaewpitakkun	2018	Psychiatry Counseling Psychology	Results from 35 Facebook users indicated that Facebook behaviours could predict depression level.
2	Deep learning for online harassment detection in tweets	Tolba Marwa, Ouadfel Salima, Meshoul Souham	2018	Broadcasting bully and harassment messages in different social media platforms	Models are considered namely Long short-term memory(LSTM) ,Bidirectional Long Short-Term Memory (BLSTM),Convolutional neural Network (CNN), and compared with other classification models. Obtained results are very encouraging.
3	Optimized Twitter Cyberbullying Detection based on Deep Learning	Monirah A. Al-Ajlan , Mourad Ykhlef	2018	Cyberbullying detection	Proposed optimized Twitter cyberbullying detection based on deep learning (OCDD), a novel approach to address the above challenges
4	X-A-BiLSTM : a Deep Learning Approach For Depression Detection in Imbalanced Data	Qing Cong ; Zhiyong Feng ; Fang Li ; Yang Xiang ; Guozheng Rao ; Cui Tao	2018	Accuracy of model	Techniques to reduce data imbalance and enhance classification capacity
5	Optimal Online Cyberbullying Detection	Daphney-Stavroula Zois ; Angeliki Kapodistria ; Mengfan Yao ; Charalampos Chelmis	2018	Cyberbullying alert	Proposed an algorithm to reduce the time to raise a cyberbullying alert by drastically reducing the number of feature evaluations necessary for a decision to be made

My Mobile Watch Dog

My Mobile Watchdog helps parents to track phone interaction from their children and help them make good electronic decisions. You will set up rules for your child with over 20 parental controls, get warnings when unusual behavior happens and see where your child is at all times. While available on Android and IOS smartphones, certain apps don't work on iPhones and there have been reports that some of the protections put up by their parents have been bypassed by teenagers. The page of My Mobile Watchdog enjoys a lot of public coverage but a lot of details is missing. The eight features described on the dashboard have very brief explanations and offer little information on what features are left out for iPhone and whether jailbreaking is needed.

Cyber Patrol Online Management

Cyber Patrol, which keeps children safe, comes for schools in stand-alone models for the home and network models. The network version either runs on local area networks or through proxy servers. Cyber Surveillance for schools is based on blocking Network access, and it goes by Microsoft's proxy service, Microsoft Internet Security and Optimization Service 2000, or Novell Border Boss. [1]

Facebook social media for Depression Detection in the Thai Community

Natural Language Processing (NLP) strategies to create an algorithm for Thai language depression diagnosis on Facebook where people use it as a platform for expressing thoughts, emotions and life events. Tests of 35 Facebook users showed Facebook habits could predict the degree of depression. [2]

Deep learning for online harassment detection in tweets

Classification challenge and explores the efficacy of deep learning in identifying online abuse in a broad human-labeled dataset designed specifically for the purpose of abuse analysis. Types are considered for this reason, including Long Short-Term Memory (LSTM), Bidirectional Long Short- Term Memory (BLSTM), Convolutional Neural Network (CNN), and contrasted with other types.[3]

Optimized Twitter Cyberbullying Detection based on Deep Learning

Cyberbullying is a crime in which a perpetrator bullies a victim who is being abused and disliked online. Some solutions to cyberbullying prevention were developed but they were mostly focused on textual and interface interfaces. The bulk of the literature work aimed at enhancing the identification by adding new apps. Though the extraction and selection processes of the app get tougher as the number of features increases. However, another downside to these enhancements is that it is easy to produce such features-for example, user age-. In this paper authors suggest automated identification of Twitter cyberbullying based on the very complex techniques of ML's deep-learning area.

Detection in Imbalanced Data

An increasing number of people suffering from mental health conditions resort to online resources (specialized websites, social media, etc.) to share their feelings. Early depression detection using social media data through deep learning models can help to change life trajectories and save lives. But the accuracy of these models was not satisfying due to the real-world imbalanced data distributions. To tackle this problem, we propose a deep learning model (X-A-BiLSTM) for depression detection in imbalanced social media data. To tackle this problem, authors are proposing a deep learning model (X- A-BiLSTM) to detect depression in imbalanced social media data. The X-A-BiLSTM architecture consists of two important components: the first is XGBoost, which is used to minimize data imbalance; and the second is a neural network Attention-BiLSTM, which enhances recognition functionality. The Reddit Self-Reported Depression Diagnosis (RSDD) dataset was chosen, which contained approximately 9,000 users who believed to have been diagnosed with depression ("diagnosed people" and approximately 107,000 matched control people. Results indicate that our methodology greatly outperforms the existing state-of-the-art models on the RSDD dataset.

Optimal Online Cyberbullying Detection

Cyberbullying has emerged as a serious societal and public health problem that demands accurate methods for the detection of cyber-bullying instances in an effort to mitigate the consequences. While techniques to automatically detect cyberbullying incidents have been developed, the scalability and timeliness of existing cyberbullying detection approaches have largely been ignored. Authors resolve the void by formulating cyberbullying prevention as a question of serial hypothesis testing. Depending on this hypothesis, authors are suggesting a novel algorithm designed to minimize the time taken to lift a cyberbullying warning by dramatically reducing the amount of feature tests available to make a decision. Authors demonstrate the effectiveness of our approach using a real-world dataset from Twitter, one of the top five networks with the highest percentage of users reporting cyberbullying instances. Authors show that our approach is highly scalable while not sacrificing accuracy for scalability.

Limitation of Existing System

A. Mood Path: A Depression

The System is not based on textual data that helps to identify mood and mental health but rather based on question-and-answer system. Also, the system also takes 14 days to form an analysis and get results.

B. Youper

Youper is chatbot based system hence the system only gets information about the questions being asked and it is more of mood tracker rather than depression or harassment detection.

C.My Mobile WatchDog

Although blocking a specific website is simple and receiving an warning should your child attempt to access anything banned, the filtering capabilities of My Smartphone Watchdog aren't very sophisticated. You cannot filter domain categories; you need to block pages one by one, instead. The location features had been deceptive. Although blocking a certain website is simple and getting a warning should your child attempt to access anything banned, the filtering capabilities of My Smartphone Watchdog aren't very sophisticated. You cannot filter domain categories; you have to block pages one by one, instead. The location features are deceiving.

Cyber Patrol Online Management

- No remote management or notification features.
- Clever kids can beat time scheduler and IM filter.
- IM filter may block harmless words.

Facebook social media for Depression Detection in the Thai Community

Facebook has limited the data extraction process because they keep the user data private. This was the main limitation as the more the data the better is the results obtained on deep learning models.

Deep learning for online harassment detection in tweets

It is ruled out to be used in future due to the private information of large human labelled corpus.

Optimized Twitter Cyberbullying Detection based on Deep Learning

It is basically limited to the tweet-based information and it doesn’t extract feature from the tweet that are fed to them by classifier.

Optimal Online Cyberbullying Detection

Limited due to advanced features and it’s use due to the scalability and timeliness of the existing cyber-bullying detection approaches.

III. PROPOSED SYSTEMS

3.1 Analysis/Framework/ Algorithm

3.1.1 ANN

ANNs are computational models inspired by an animal's central nervous systems. It is capable of machine learning as well as pattern recognition. These presented as systems of interconnected "neurons" which can compute values from inputs. Stop by any EFT center, so as to make the payment.

A neural network is an oriented graph. It consists of nodes which in the biological analogy represent neurons, connected by arcs. It corresponds to dendrites and synapses. Each arc associated with a weight while at each node. Apply the values received as input by the node and define Activation function along the incoming arcs, adjusted by the weights of the arcs.

A neural network is a machine learning algorithm based on the model of a human neuron. The human brain consists of millions of neurons. It sends and process signals in the form of electrical and chemical signals. These neurons are connected with a special structure known as synapses. Synapses allow neurons to pass signals. From large numbers of simulated neurons neural networks forms.

An Artificial Neural Network is an information processing technique. It works like the way human brain processes information. ANN includes a large number of connected processing units that work together to process information. They also generate meaningful results from it. We can apply Neural network not only for classification. It can also apply for regression of continuous target attributes.

Neural networks find great application in data mining used in sectors. For example economics, forensics, etc and for pattern recognition. It can be also used for data classification in a large amount of data after careful training.

A neural network may contain the following 3 layers:

Input layer – The activity of the input units represents the raw information that can feed into the network.

- Hidden layer – To determine the activity of each hidden unit. The activities of the input units and the weights on the connections between the input and the hidden units. There may be one or more hidden layers.
- Output layer – The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

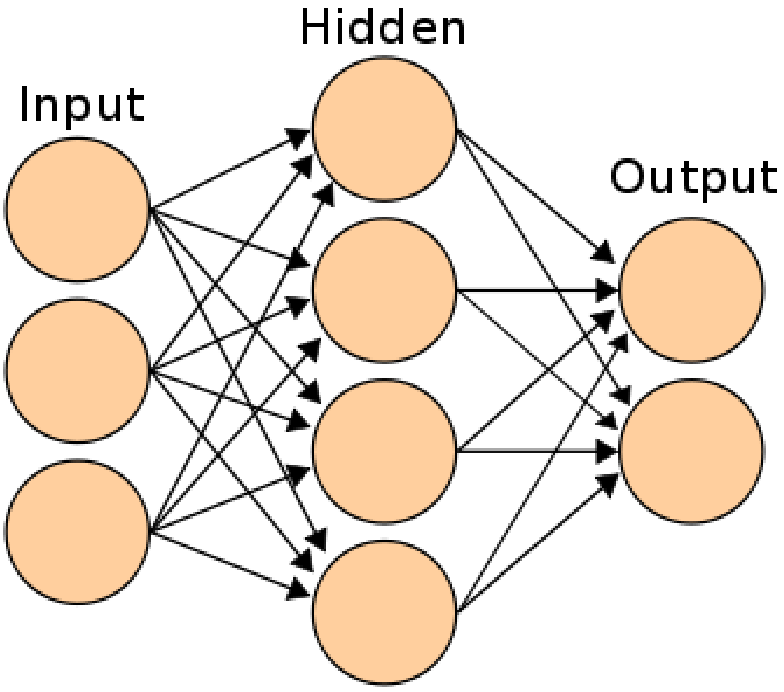


Figure 1 ANN ARCHIETECTURE

3.2 Details of Hardware & Softwares used

Table 3.2.1 Modules developed

Modules & Sub-systems	Hardware Requirements	Software / Libraries Requirements	Technology	Application
Data Collection	8GB, 8GB+RAM, Intel i3, i3+ processor	Python, Twint, Kaggle	Scrapping	Dataset formation

Data Preparatio	8GB, 8GB+ RAM, Intel i3, i3+ processor	Python, Pandas, Re, Tensorflow, Keras, Matplotlib, Seaborn	Data Analysis and visualization	Data cleaning and splitting for model
Model Developmen	8GB, 8GB+RAM, Intel i3 , i3+ processor	Python, Tensorflow, Keras, Sklearn, Matplotlib	Machine Learning, Deep Learning	Model building
Front-End and Back-End	4 GB, 4 GB+RAM, Intel i3 , i3+ processor	Flask, Pickle, Pandas, HTML/CSS, Bootstrap.	Web Development	Website

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015.

3.2.2 NumPy

NumPy is a library for the Python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

3.2.3 Pandas

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three- clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

3.2.4 Twint

Twint is an advanced Twitter scraping tool written in Python that allows for scraping Tweets from Twitter profiles without using Twitter's API. Twint utilizes Twitter's search operators to let you scrape Tweets from specific users, scrape Tweets relating to certain topics, hashtags & trends, or sort out sensitive information from Tweets like e-mail and phone numbers. I find this very useful, and you can get really creative with it too. Twint also makes special queries to Twitter allowing you to also scrape a Twitter user's followers, Tweets a user has liked, and who they follow without any authentication, API, Selenium, or browser emulation.

Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network.

3.2.5 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter, has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012, and further joined by Thomas Caswell. Matplotlib 2.0.x supports Python versions 2.7 through 3.6. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6. Matplotlib has pledged to not support Python 2 past 2020 by signing the Python 3 Statement.

3.2.6 Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

3.2.7 Sklearn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

3.2.8 Flask web framework

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program. Some popular applications that use the Flask framework include Pinterest and LinkedIn.

Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is the sixth- most-starred project on GitHub, with more than 135,000 stars, behind free Code Camp and marginally behind Vue.js framework. According to Alexa Rank, Bootstrap is in the top-2000 in US while vuejs.org is in top-7000 in US.

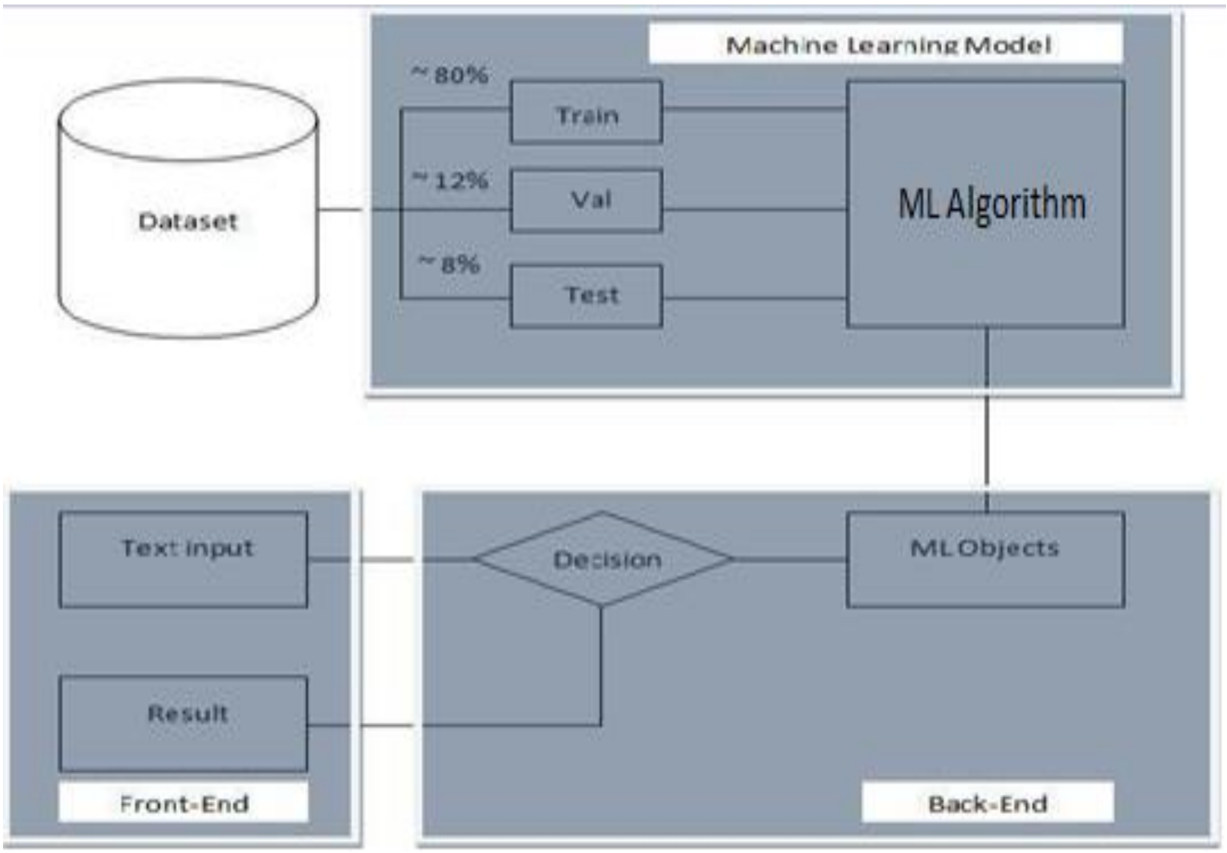


Figure. 2 Proposed System

1. The basic design we plan on is to give a website to the user where the user can input the textual post or comment or message in an input form.

2. Once the user submits the form the system will run the prediction and detect whether the input is depressive, harassment or neither.

3. The user is then prompted by the system the result of the prediction on a page. The user can go back to the landing page and try different inputs (textual post).

4. The basic design we plan on is to give a website to the user where the user can input the textual post or comment or message in an input form.

5. Once the user submits the form the system will run the prediction and detect whether the input is depressive, harassment or neither.

6. The user is then prompted by the system the result of the prediction on a page. The user can go back to the landing page and try different inputs (textual post).

3.3 Methodology

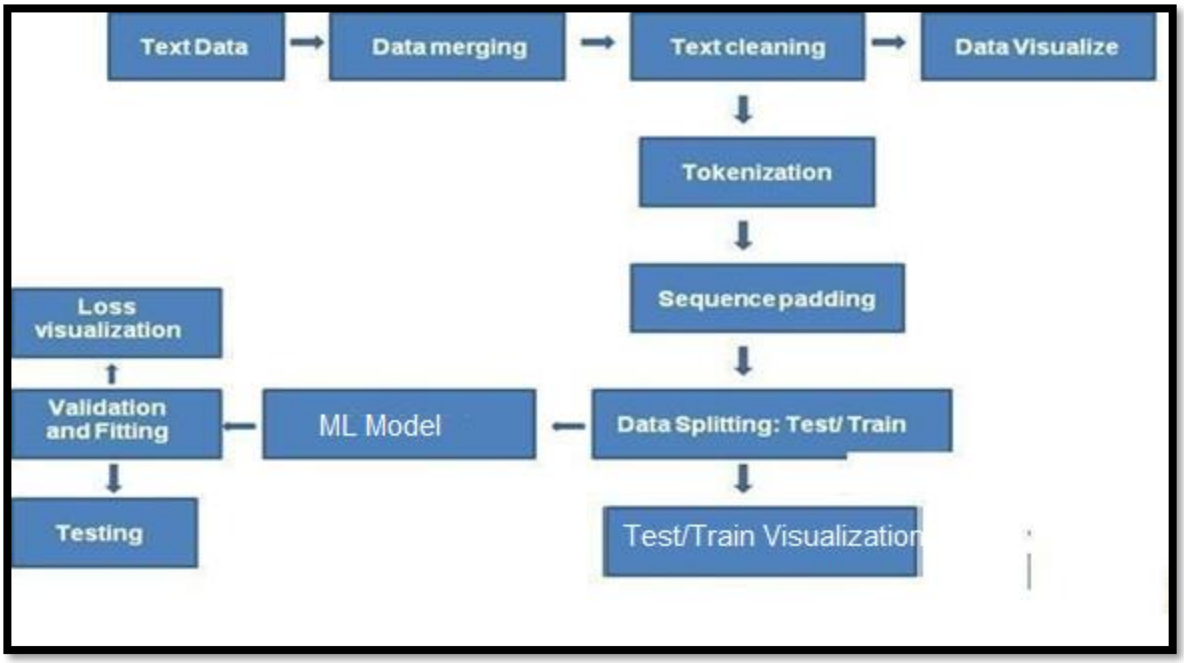


Figure 3 Flowchart of ML Model

Our model basically focuses on detecting depressive and harassment related textual post's that are shared on social media. A problem with modeling text is that it is messy, and techniques like machine learning algorithms prefer well defined fixed- length inputs and outputs. Machine learning algorithms cannot work with raw text directly; the text must be converted into numbers. Specifically, vectors of numbers. This model comprises of techniques like Bag of Words. The bag-of-words model is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

Hence, we first read the data that is available to us in the csv files. We load data from those multiple csv files. Then we perform data merging to create a single dataset on which we then perform cleaning. We then visualize the dataset distribution and perform tokenization to represent text data for the model in form of numbers. Then we perform sequence padding and then we split the data into 3 sets. We then visualize the distribution of these 3 sets and check if it's similar to the original distribution. Then we train the ANN model and visualize the train and validation result to detect underfit and overfit if they occur. Then we run the test set and check the

results.

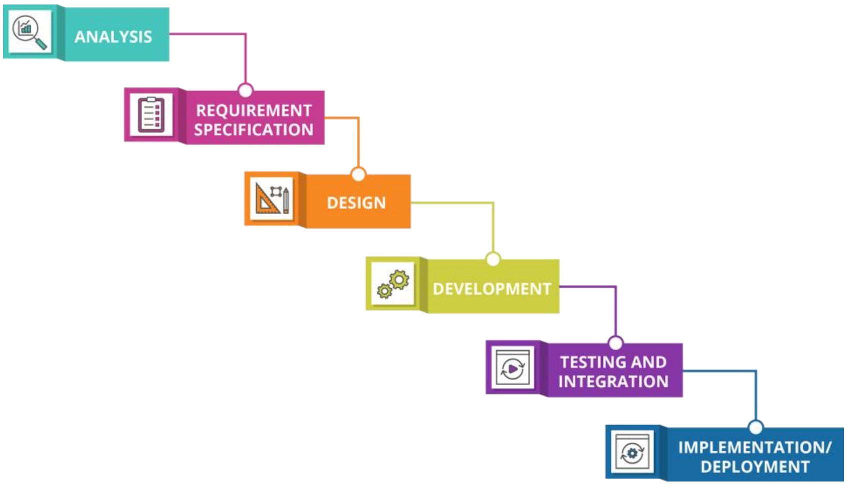


Figure 4 Flow of Waterfall Model

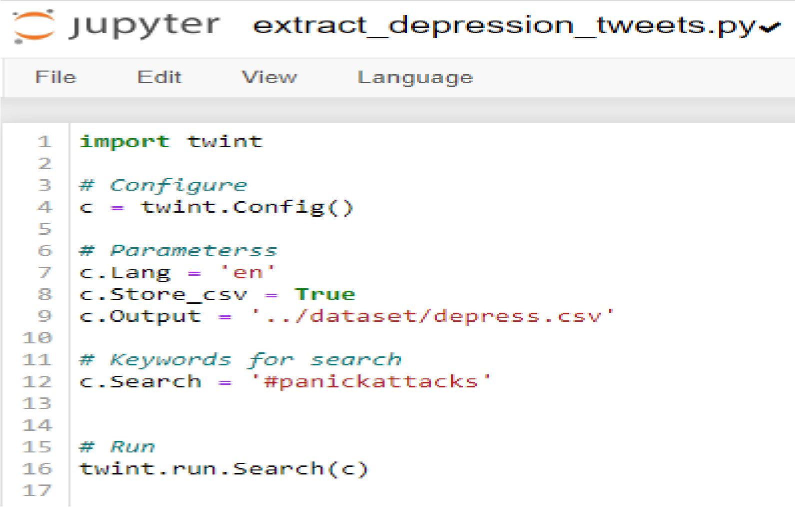
The waterfall model is a breakup of mission tasks into linear sequential stages, where each step depends on the previous one's deliverables and leads to a task specialization. In other aspects of computer design the approach is common. This appears to be among the most iterative and scalable methods in software development, as change moves overwhelmingly in one direction ("downwards" like a waterfall) through the various and essential phases of creation, introduction, analysis, planning, building, testing, delivery and maintenance. The waterfall architecture paradigm emerged in the auto motive and building industries; where the tightly organized physical structures meant that improvements in nature had been prohibitively costly in the production process even earlier. There were no recognized substitutes for knowledge- based artistic research when first implemented for software development.

IV. MODULES OR EMPERICAL STUDY

Training Classification Model

A. Preparing Dataset

Twint is an innovative Python-written Twitter scraping application that helps you to retrieve Tweets from Twitter accounts by utilizing Twitter's API. We used twint library to retrieve the depression and harassment dataset based on keywords and hashtags that reflect depression and harassment. Other than twint we have used sentiment 140 dataset and toxic comment dataset to collect positive, negative and toxic comments to add to our dataset. The labels we have annoted are 0 for non-harassment and non-depression in the respective models and 1 for depression or harassment in the respective models. We have done sampling of these Kaggle datasets and our fetched datasets to achieve a distribution.

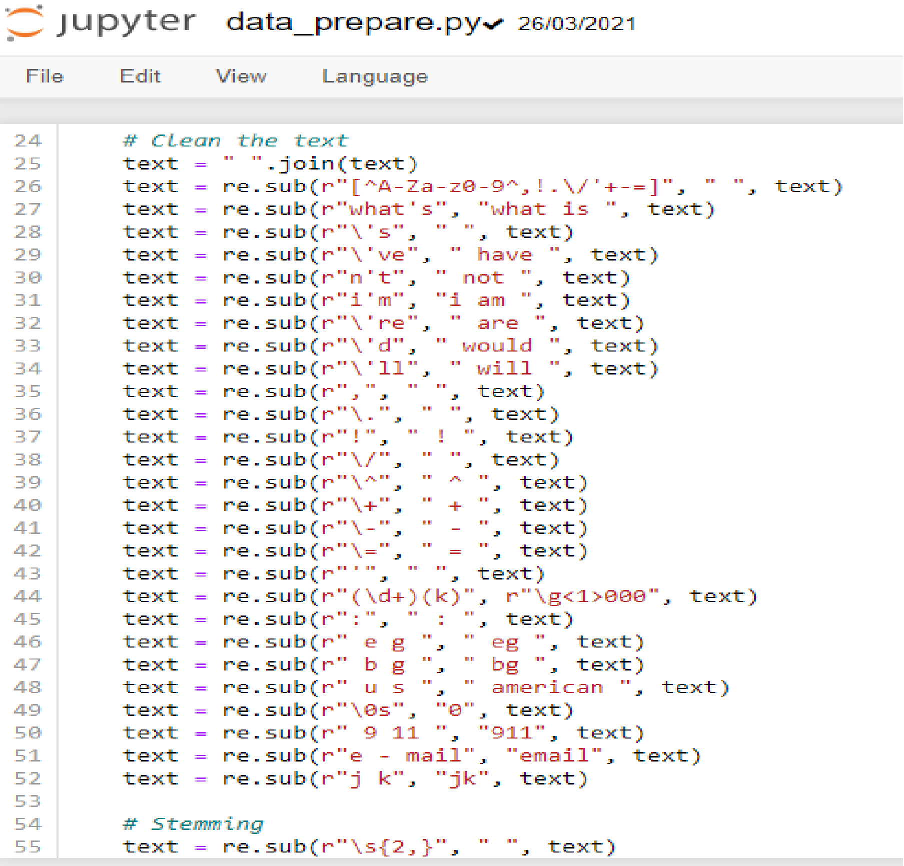


```
import twint

# Configure
c = twint.Config()

# Parameterss
c.Lang = 'en'
c.Store_csv = True
c.Output = '../dataset/depress.csv'

# Keywords for search
c.Search = '#panickattacks'


# Run
twint.run.Search(c)

```

Figure 5 Data Extraction

B. Data Cleaning

Data cleaning is an essential tool for Data and NLP research. Text cleaning involves eliminating stopwords (words that don't make much of a difference to the model) and these are usually pronouns, conjuctions etc. In addition to that code cleaning in NLP involves deleting unique characters such as @, # and trailing. Stemming is translating those terms that are identical to a specific root word so model will apply to all those cases where similar words are used, function better, and therefore minimize difficulty.

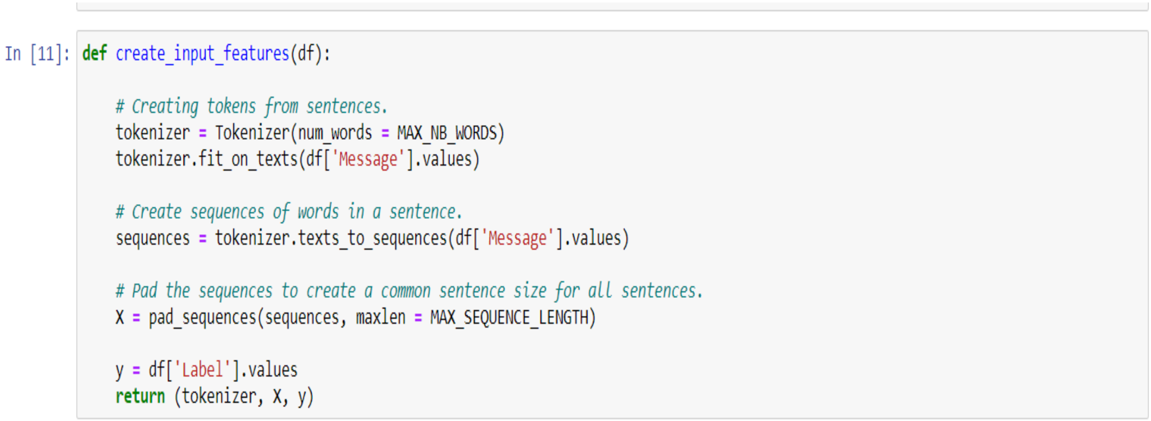


```
# Clean the text
text = " ".join(text)
text = re.sub(r"[^A-Za-z0-9^,!.\/'+-=]", " ", text)
text = re.sub(r"what's", "what is ", text)
text = re.sub(r"\'s", " ", text)
text = re.sub(r"\'ve", " have ", text)
text = re.sub(r"n't", " not ", text)
text = re.sub(r"i'm", "i am ", text)
text = re.sub(r"\'re", " are ", text)
text = re.sub(r"\'d", " would ", text)
text = re.sub(r"\'ll", " will ", text)
text = re.sub(r",", " ", text)
text = re.sub(r"\.", " ", text)
text = re.sub(r"!", " ! ", text)
text = re.sub(r"\/", " ", text)
text = re.sub(r"\^", " ^ ", text)
text = re.sub(r"\+", " + ", text)
text = re.sub(r"\-", " - ", text)
text = re.sub(r"\=", " = ", text)
text = re.sub(r"'", " ", text)
text = re.sub(r"(\d+)(k)", r"\g<1>000", text)
text = re.sub(r":", " : ", text)
text = re.sub(r" e g ", " eg ", text)
text = re.sub(r" b g ", " bg ", text)
text = re.sub(r" u s ", " american ", text)
text = re.sub(r"\0s", "0", text)
text = re.sub(r" 9 11 ", "911", text)
text = re.sub(r"e - mail", "email", text)
text = re.sub(r"j k", "jk", text)

# Stemming
text = re.sub(r"\s{2,}", " ", text)
```

Figure 6 Data Cleaning

C. Tokenization

Tokenization is a more sophisticated form of information management that can suit and exchange different text documents. It could be the right option for large projects. The Tokenizer must be built to fit into either raw text or encoded text documents in an integer. Next is sequencing and sequence padding where we transform the tokens into sequences used for a sequential pattern. These sequences that vary in duration, so we pad certain sequences to get all the statements or sequences to a standard length.



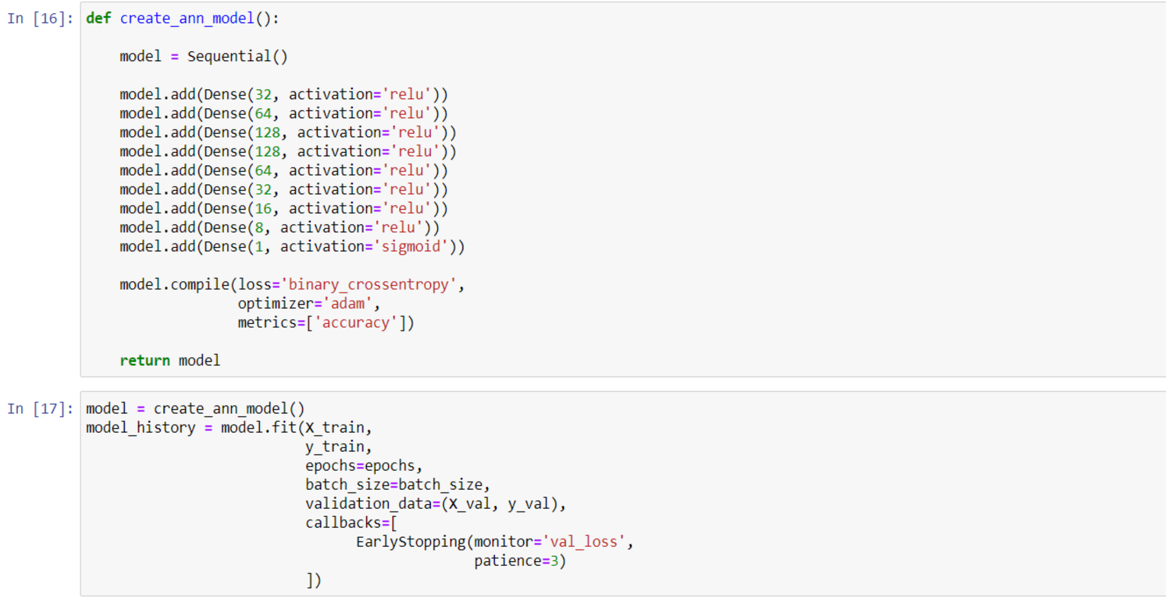
```
In [11]: def create_input_features(df):

             # Creating tokens from sentences.
             tokenizer = Tokenizer(num_words = MAX_NB_WORDS)
             tokenizer.fit_on_texts(df['Message'].values)

             # Create sequences of words in a sentence.
             sequences = tokenizer.texts_to_sequences(df['Message'].values)

             # Pad the sequences to create a common sentence size for all sentences.
             X = pad_sequences(sequences, maxlen = MAX_SEQUENCE_LENGTH)

             y = df['Label'].values
             return (tokenizer, X, y)
```

Figure 7 Tokenization

D. ANN

A neural network may contain the following 3 layers:

- Input layer – The activity of the input units represents the raw information that can feed into the network.
- Hidden layer – To determine the activity of each hidden unit. The activities of the input units and the weights on the connections between the input and the hidden units. There may be one or more hidden layers.
- Output layer – The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.



```
In [16]: def create_ann_model():

             model = Sequential()

             model.add(Dense(32, activation='relu'))
             model.add(Dense(64, activation='relu'))
             model.add(Dense(128, activation='relu'))
             model.add(Dense(128, activation='relu'))
             model.add(Dense(64, activation='relu'))
             model.add(Dense(32, activation='relu'))
             model.add(Dense(16, activation='relu'))
             model.add(Dense(8, activation='relu'))
             model.add(Dense(1, activation='sigmoid'))

             model.compile(loss='binary_crossentropy',
                           optimizer='adam',
                           metrics=['accuracy'])

             return model

In [17]: model = create_ann_model()
         model_history = model.fit(X_train,
                                   y_train,
                                   epochs=epochs,
                                   batch_size=batch_size,
                                   validation_data=(X_val, y_val),
                                   callbacks=[
                                         EarlyStopping(monitor='val_loss',
                                                       patience=3)
                                   ])
```

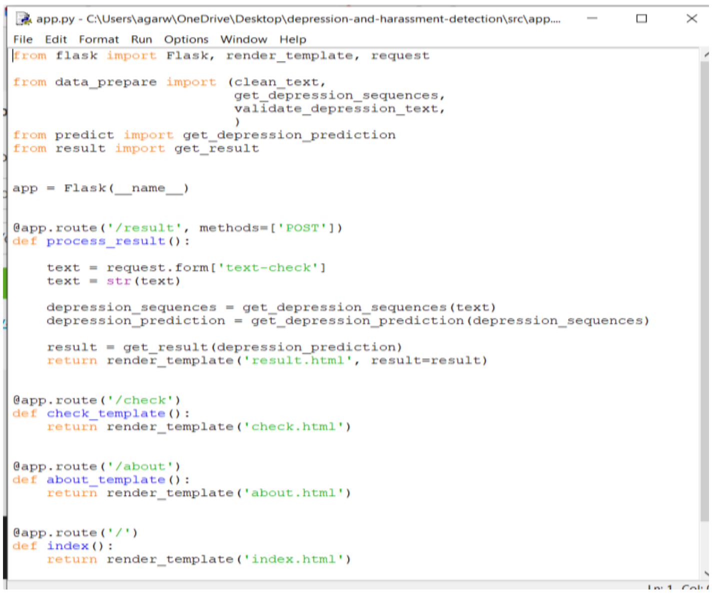
Figure 8 ANN Model

Back-end Implementation

Web Server

We used Flask Web Framework to develop our Hyper Text Transfer Protocol (HTTP) server. This server is used to provide the following services:

- To provide a website for the user to access.
- Accept text from the users to predict


app.py - C:\Users\agarw\OneDrive\Desktop\depression-and-harassment-detection\src\app....
File Edit Format Run Options Window Help

```
from flask import Flask, render_template, request

from data_prepare import (clean_text,
                          get_depression_sequences,
                          validate_depression_text,
                          )
from predict import get_depression_prediction
from result import get_result


app = Flask(__name__)


@app.route('/result', methods=['POST'])
def process_result():

    text = request.form['text-check']
    text = str(text)

    depression_sequences = get_depression_sequences(text)
    depression_prediction = get_depression_prediction(depression_sequences)

    result = get_result(depression_prediction)
    return render_template('result.html', result=result)


@app.route('/check')
def check_template():
    return render_template('check.html')


@app.route('/about')
def about_template():
    return render_template('about.html')


@app.route('/')
def index():
    return render_template('index.html')
```

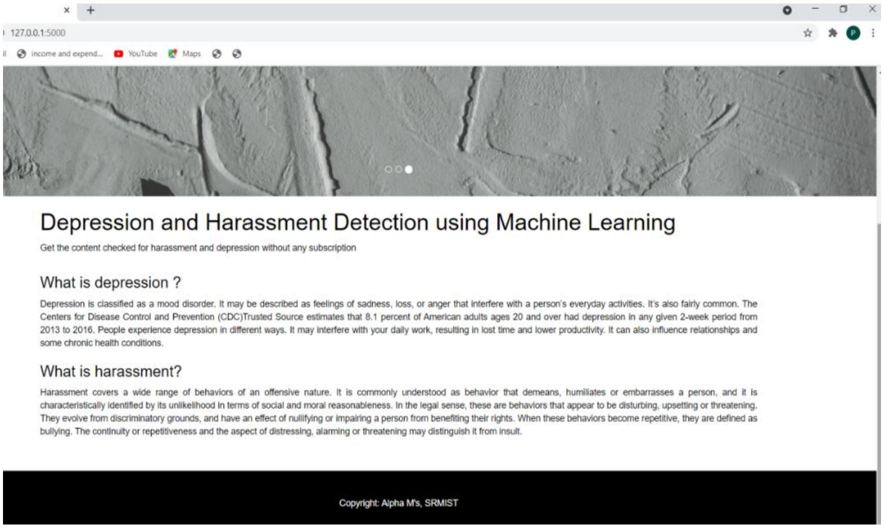
Figure 9 Back End

4.2 Front-end Implementation

The front end is a Web App which consists of the following modules:

A. Home Page

The home page acts as the landing page where the user comes when visiting our site. It has some information about what is depression and what is harassment.



Depression and Harassment Detection using Machine Learning

Get the content checked for harassment and depression without any subscription

What is depression ?

Depression is classified as a mood disorder. It may be described as feelings of sadness, loss, or anger that interfere with a person's everyday activities. It's also fairly common. The Centers for Disease Control and Prevention (CDC)Trusted Source estimates that 8.1 percent of American adults ages 20 and over had depression in any given 2-week period from 2013 to 2016. People experience depression in different ways. It may interfere with your daily work, resulting in lost time and lower productivity. It can also influence relationships and some chronic health conditions.

What is harassment?

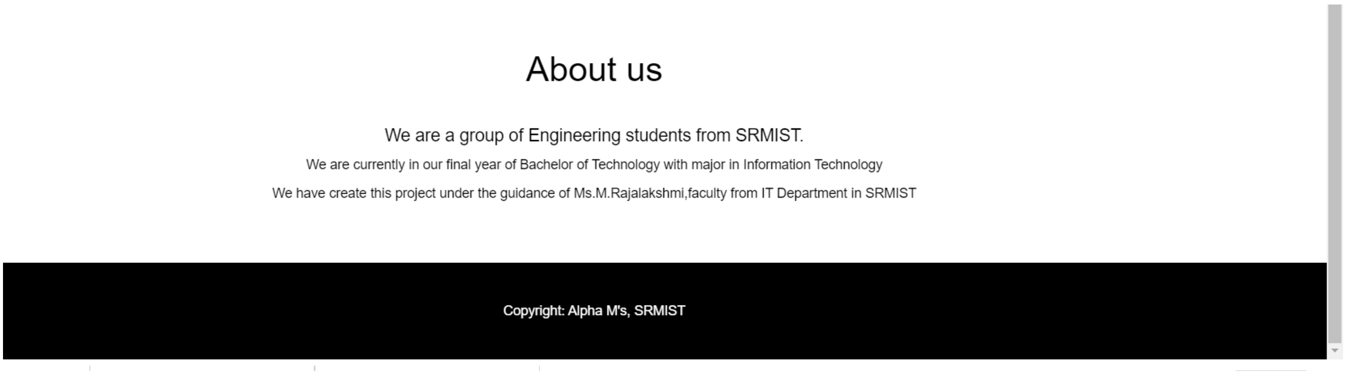
Harassment covers a wide range of behaviors of an offensive nature. It is commonly understood as behavior that demeans, humiliates or embarrasses a person, and it is characteristically identified by its unlikelihood in terms of social and moral reasonableness. In the legal sense, these are behaviors that appear to be disturbing, upsetting or threatening. They evolve from discriminatory grounds, and have an effect of nullifying or impairing a person from benefiting their rights. When these behaviors become repetitive, they are defined as bullying. The continuity or repetitiveness and the aspect of distressing, alarming or threatening may distinguish it from insult.

Copyright: Alpha M's, SRMIST

Figure 10 Home Page

B. About Page

The about page contains information about the team that developed the system.



About us

We are a group of Engineering students from SRMIST.

We are currently in our final year of Bachelor of Technology with major in Information Technology

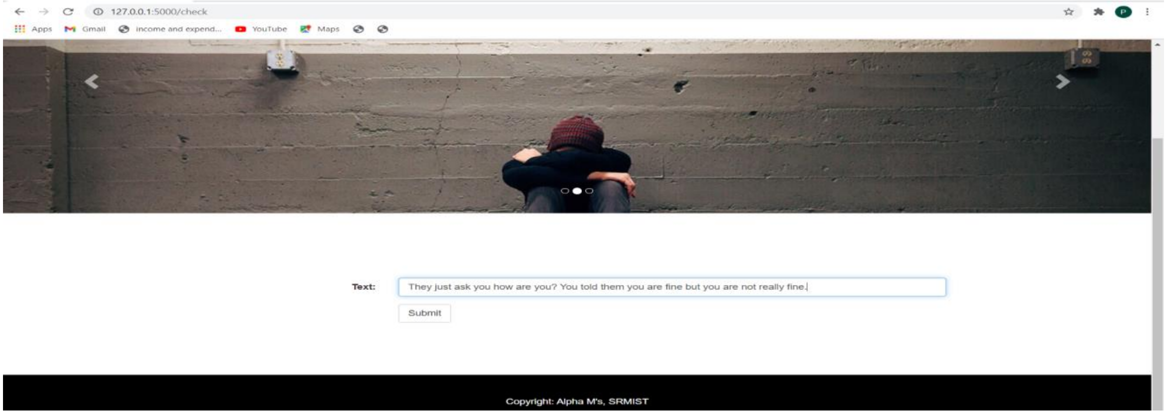
We have create this project under the guidance of Ms.M.Rajalakshmi,faculty from IT Department in SRMIST

Copyright: Alpha M's, SRMIST

Figure 11 About Page

C. Check Page

The check page is the main core part of the front-end. It is the page where the user will enter the text on which detection/ classification is to be performed. It contains the form thought which the user will enter data.



Text: They just ask you how are you? You told them you are fine but you are not really fine.

Copyright: Alpha M's, SRMIST

Figure 12 Check Page

Testing

Table 5.1 Development of Test cases (Functional)

Test Case ID	Objective	Steps / Description	Input	Expected Output	Actual Output	Result	Remark
IC01	To test the preprocessing of data	Pass the text to Cleaning and numbers Tokenization And Padding	Social media text	Numberized vector	Vector of	Pass	Successfully prepared data
IC02	To test for Balanced distribution of dataset	Pass dataset to seaborn counter plot	Data set	Balanced dataset plot	Balance dataset	Pass	Successfully test for data balance
IC03	To test the train/ test/ validation distribution	Pass the Train/ test/ Validation to count plot	Split dataset	3 balanced count plot	Balance data for all 3 sets.	Pass	Successfully test for similar distribution in split
IC04	To test the condition for overfit/ under fit	Pass loss of train and validation for plot	Train & validate loss road	Plot with loss values	Loss Values are close	Pass	No overfit No underfit
IC05	To test the detection accuracy on Test set classifier	Pass the Test set to	Test set	Very close Accuracy to 98%	Accuracy 98.0%	Pass	The model is generalized
AP01	To test the pages of website	User will Enter the local host url	Localhost url of webpage	Landing page and	Land Pages and all other pages	Pass	All page loaded

Result and Analysis

We have used the model to predict different textual data to check whether the text belongs to depression or harassment. The result of training the models are :

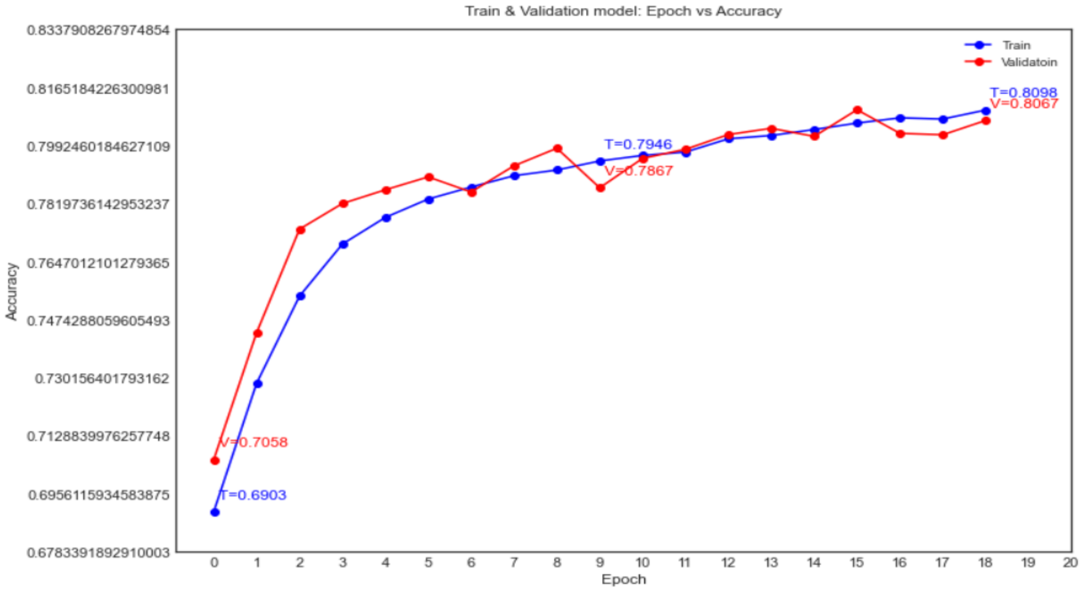


Figure 13 Accuracy Graph

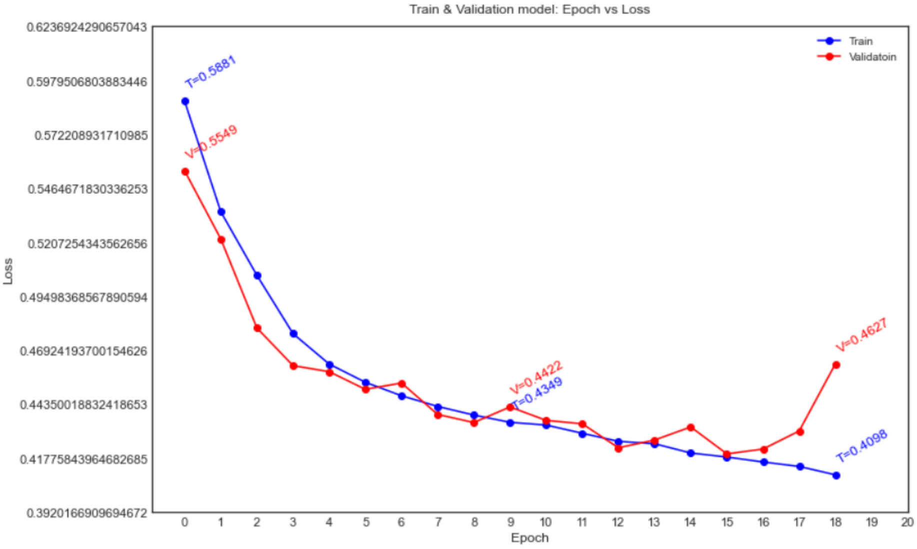
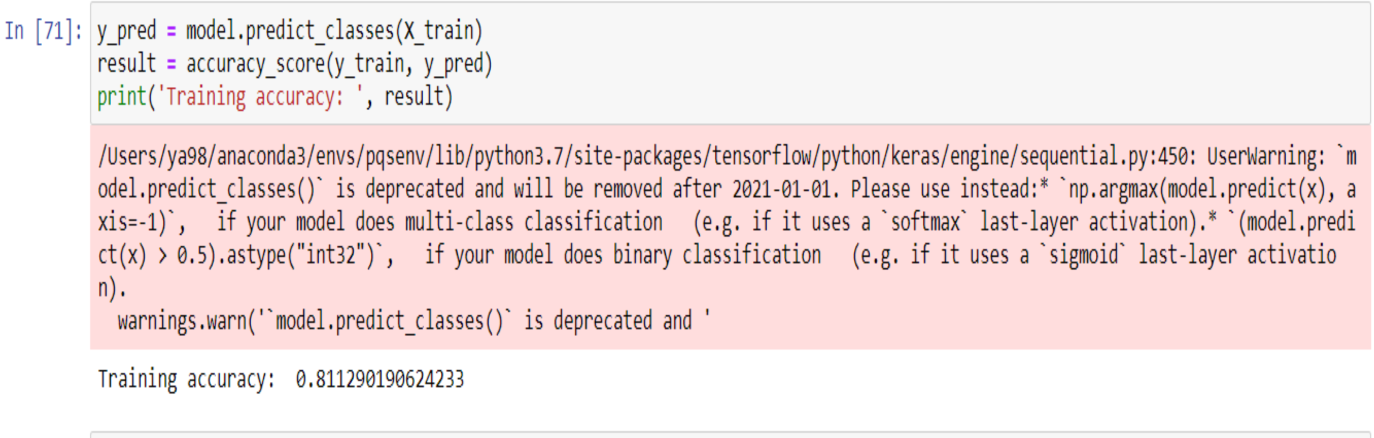


Figure 14 Epoch vs Loss

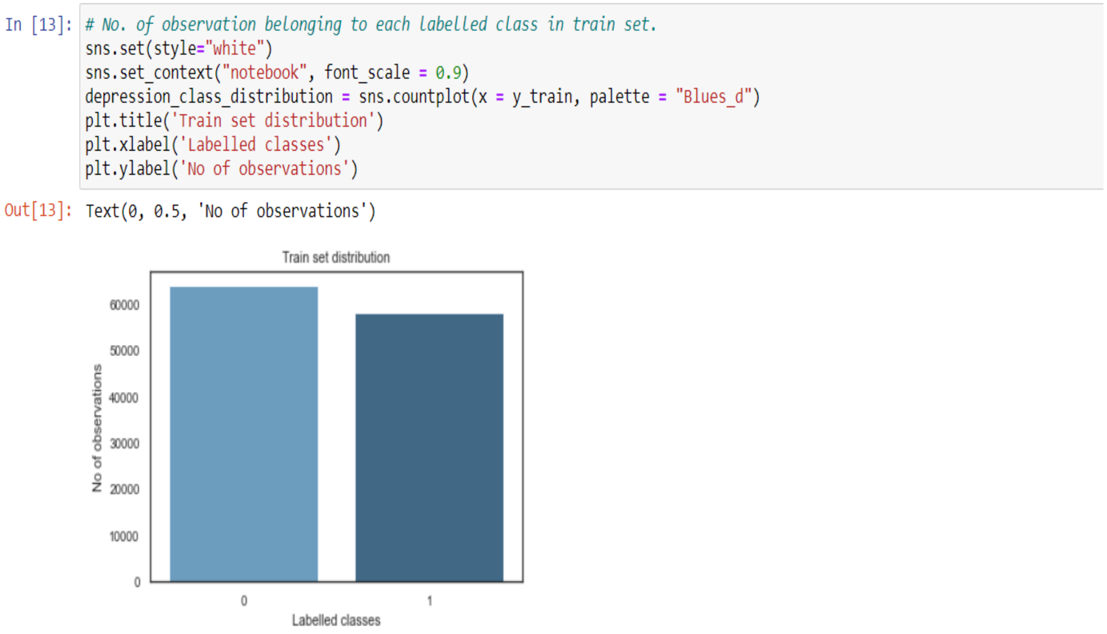
© 2022, IJSREM | www.ijsrem.com DOI: 10.55041/IJSREM16475

Training set: Set of data used to train the model. At each epoch our model will be trained over and over again on this same data in our training set and it will continue to learn about the features of this data with a hope that this data could be used to deploy our model and have it accurately predict the new data that it has never seen before.

Training Accuracy: 81.12%



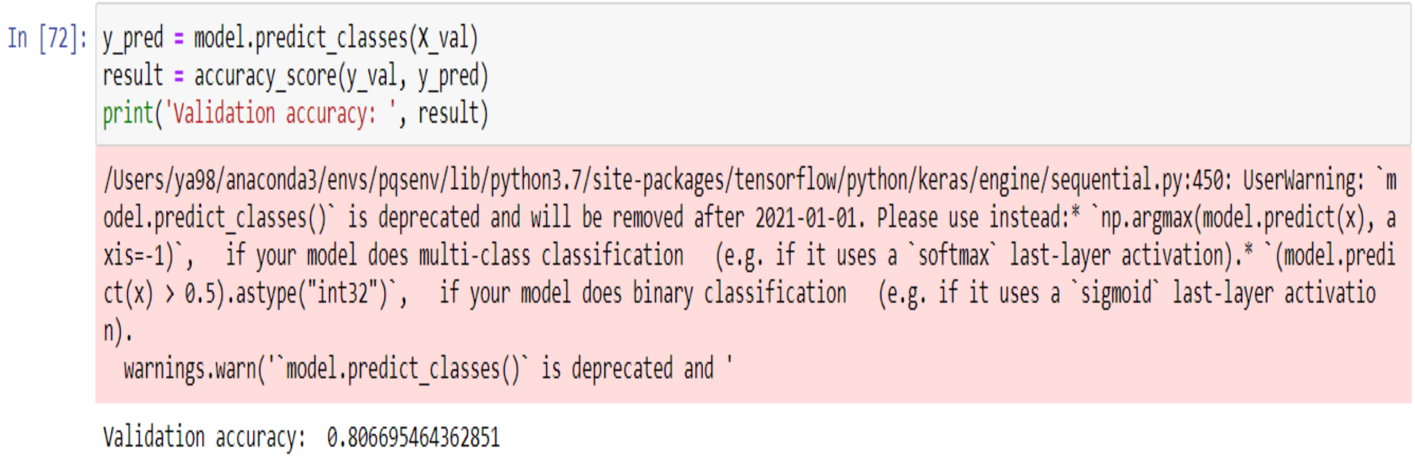
```
In [71]: y_pred = model.predict_classes(X_train)
         result = accuracy_score(y_train, y_pred)
         print('Training accuracy: ', result)

/Users/ya98/anaconda3/envs/pqsenv/lib/python3.7/site-packages/tensorflow/python/keras/engine/sequential.py:450: UserWarning: `m
odel.predict_classes()` is deprecated and will be removed after 2021-01-01. Please use instead:* `np.argmax(model.predict(x), a
xis=-1)`,   if your model does multi-class classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.predi
ct(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it uses a `sigmoid` last-layer activatio
n).
  warnings.warn('`model.predict_classes()` is deprecated and '

Training accuracy:  0.811290190624233
```



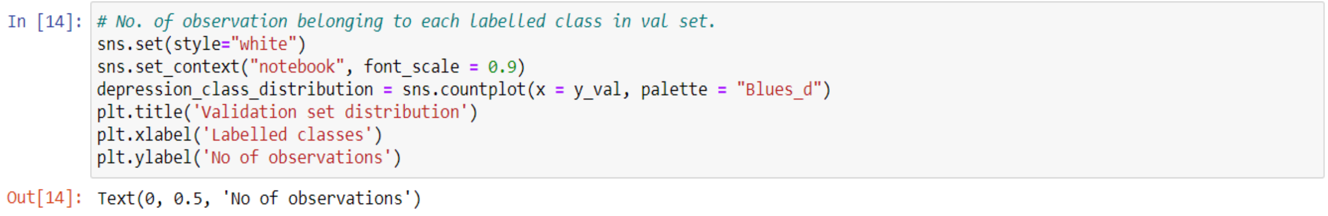
```
In [13]: # No. of observation belonging to each labelled class in train set.
         sns.set(style="white")
         sns.set_context("notebook", font_scale = 0.9)
         depression_class_distribution = sns.countplot(x = y_train, palette = "Blues_d")
         plt.title('Train set distribution')
         plt.xlabel('Labelled classes')
         plt.ylabel('No of observations')

Out[13]: Text(0, 0.5, 'No of observations')
```

Validation set: It is a data set separate from the training set that is used to validate our model during training. This process helps to give information that may assist us with adjusting our hyper parameters. It will be simultaneously done with the training set. The model will be classifying each input from validation set as well based on what it has learned from the training set.

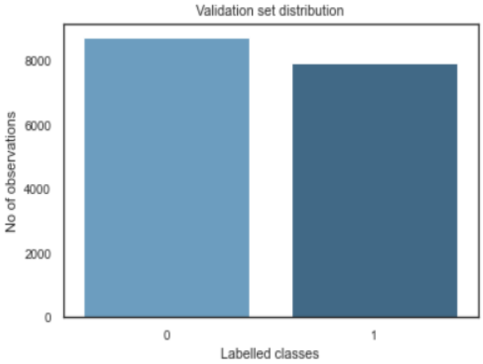
Validation Accuracy: 80.66%



```
In [72]: y_pred = model.predict_classes(X_val)
         result = accuracy_score(y_val, y_pred)
         print('Validation accuracy: ', result)

/Users/ya98/anaconda3/envs/pqsenv/lib/python3.7/site-packages/tensorflow/python/keras/engine/sequential.py:450: UserWarning: `m
odel.predict_classes()` is deprecated and will be removed after 2021-01-01. Please use instead:* `np.argmax(model.predict(x), a
xis=-1)`,   if your model does multi-class classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.predi
ct(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it uses a `sigmoid` last-layer activatio
n).
  warnings.warn('`model.predict_classes()` is deprecated and '

Validation accuracy:  0.806695464362851
```

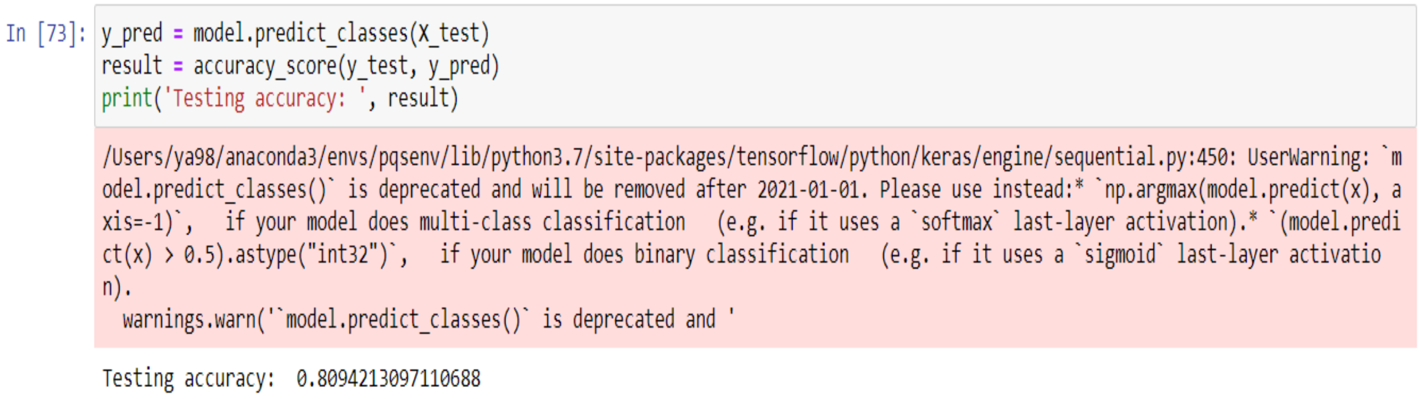


```
In [14]: # No. of observation belonging to each labelled class in val set.
         sns.set(style="white")
         sns.set_context("notebook", font_scale = 0.9)
         depression_class_distribution = sns.countplot(x = y_val, palette = "Blues_d")
         plt.title('Validation set distribution')
         plt.xlabel('Labelled classes')
         plt.ylabel('No of observations')

Out[14]: Text(0, 0.5, 'No of observations')
```

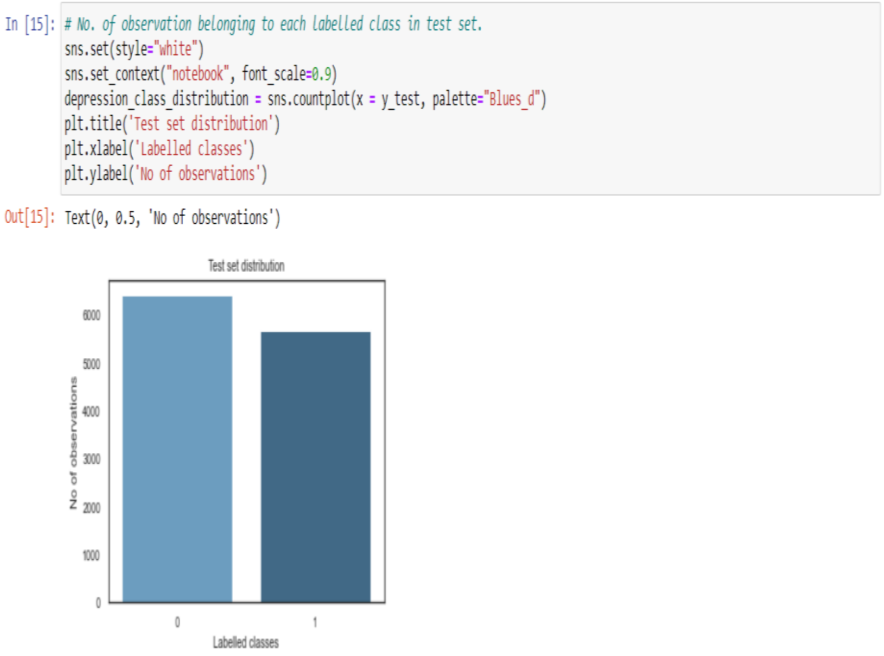


Testing set: Set of data used to set the model after the model has already been trained. This test set is separate from both training and validation set. After our model has trained and validated using these test set and validation set, we will then use our model to predict the output of the data in the test set.

Testing Accuracy: 80.94%



```
In [73]: y_pred = model.predict_classes(X_test)
         result = accuracy_score(y_test, y_pred)
         print('Testing accuracy: ', result)

/Users/ya98/anaconda3/envs/pqsenv/lib/python3.7/site-packages/tensorflow/python/keras/engine/sequential.py:450: UserWarning: `m
odel.predict_classes()` is deprecated and will be removed after 2021-01-01. Please use instead:* `np.argmax(model.predict(x), a
xis=-1)`,   if your model does multi-class classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.predi
ct(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it uses a `sigmoid` last-layer activatio
n).
  warnings.warn('`model.predict_classes()` is deprecated and '

Testing accuracy:  0.8094213097110688
```



```
In [15]: # No. of observation belonging to each labelled class in test set.
         sns.set(style="white")
         sns.set_context("notebook", font_scale=0.9)
         depression_class_distribution = sns.countplot(x = y_test, palette="Blues_d")
         plt.title('Test set distribution')
         plt.xlabel('Labelled classes')
         plt.ylabel('No of observations')

Out[15]: Text(0, 0.5, 'No of observations')
```

Conclusion

We have completed challenges listed in depression and harassment detection on textual data on implementing, developing and testing the system on various inputs. The system performance is found to be significant in determining the type of text entered whether it belongs to class depression or class harassment or neither. We were able to show by implementing the system on real time data and under realtime circumstances that technology can help solve this problem and give an early indication of depression and harassment.

FUTURE ENHANCEMENT

- Due to the increasing popularity of communications occurring in social networks, an inconsiderable part of people's interpersonal interactions are performed online.
- New risks and threats emerge through the introduction of communication media, such as cyber bullying, stalking and online grooming.
- Bullying, violence, and depression are the cornerstones of rising suicide worldwide.
- Therefore, this project seeks to build a method for determining whether or not a given post is suicidal or harassing.

REFERENCES

1. Kantinee Katchapakirin, Konlakorn Wongpatikaseree, Panida Yomaboot, Yongyos Kaewpitakkun, " Facebook Social Media for Depression Detection in the Thai Community", 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE).
2. Monirah A. Al-Ajlan , Mourad Ykhlef, "Optimized Twitter Cyberbullying Detection based on Deep Learning", 2018 21st Saudi Computer Society National Computer Conference (NCC).
3. Tolba Marwa, Ouadfel Salima, Meshoul Souham, "Deep learning for online harassment detection in tweets", 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS).
4. Qing Cong ; Zhiyong Feng ; Fang Li ; Yang Xiang ; Guozheng Rao ; Cui Tao "X-A-BiLSTM : a Deep Learning Approach For Depression Detection in Imbalanced Data." 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)
5. Daphney-Stavroula Zois ; Angeliki Kapodistria ; Mengfan Yao ; Charalampos Chelmis "Optimal Online Cyberbullying Detection", 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

6. ANN Model: https://www.sciencedirect.com/topics/engineering/artificial-neural-network

7. World Health Organization. https://www.who.int/mental_health/prevention/ suicide/suicideprevent/en/, last accessed 2019/08/19

8. Flow of Waterfall Model [Online]. Available:https://en.wikipedia.org/wiki/Waterfall_model.

9. Sentiment140 Dataset [Online]. Available: https://www.kaggle.com/kazanova/sentiment140

10. Toxic Comment Classification Dataset [Online].Available: https://www.kaggle.com/c/jigsaw- toxic-comment-classification-challenge

11. Seabrook, E.M., Kern, M.L., Fulcher, B.D., and Rickard, N.S.: Predicting depression from language-based emotion dynamics: longitudinal analysis of Facebook and Twitter status updates. Journal of Medical Internet Research 20 (5), e168 (2018).

12. Stankevich, M., Isakov, V., Devyatkin, D., and Smirnov, I.: Feature Engineering for Depression Detection in Social Media. In ICPRAM, 426–431 (2018).

13. Stankevich, M., Isakov, V., Devyatkin, D., and Smirnov, I.: Feature Engineering for Depression Detection in Social Media. In ICPRAM, 426–431 (2018).

14. Mowery, D., Bryan, C., & Conway, M. Feature studies to inform the classification of depressive symptoms from Twitter data for population health. Preprint at arXiv:1701.08229 (2017)

15. Nadeem, M. Identifying depression on Twitter. Preprint at arXiv:1607.07384 (2016).

16. Harman, G., & Dredze, M. H. Measuring post traumatic stress disorder in Twitter. Proceedings of ICWSM, (2014)

1.

2.