Software Requirement Specification Document
Fake Tweets Detection

Mariam Khaled, Radwa Mostafa, Abdelrahman Gamal, John Gerges

February 5, 2020

1 Introduction

1.1 Purpose of this document

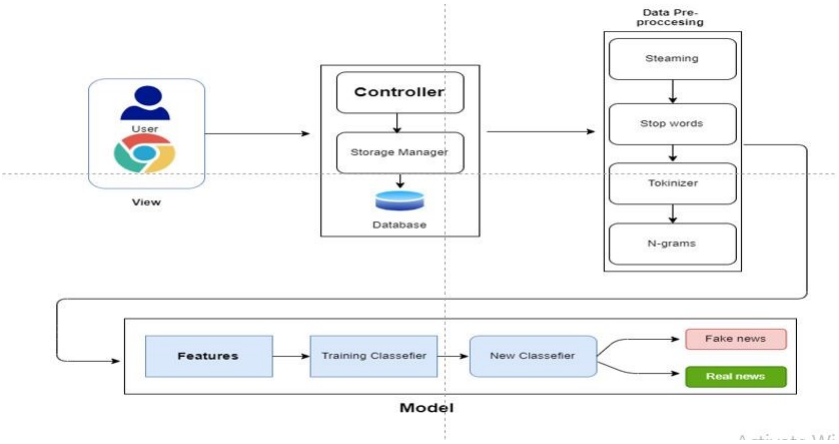
The main purpose of this Software Requirements Specification document is to outline the requirements for early detection of fake news: detect and separating between fake and real news/accounts on twitter. This is done with the aid of machine learning techniques and other features. This document will provide a detailed overview of our software product's parameters and goals and explain

purpose of our project. This software requirements specification (SRS) document defines how our stakeholder, team, and audience see the product and its functionality.

1.2 Scope of this document

This Software Requirements Specification (SRS) helps in detection of Fake news as it has an enormous social impact as it can mislead citizens into believing misinformation about a specific product/ person or situation and the spread of economic and political fake news can directly affect the stock market and government economics in general. Nevertheless, Tourism income can also be affected by fake news, as spreading fake news in countries can make tourists not interested in visiting certain places which costs losing much income.. This targets social media users that rely on social as their source of information.

1.3 Overview



Our System aims to detect fake tweets by using machine learning and a combination of algorithms The proposed diagram using MVC model (Model View Controller),The user's view a web based application ,entering a topic to search on then passing tweets related to this topic to the controller , this tweets needs pre-processing to be well defined data to work on like (stemming , stop words , toknizer, n-grams). After that the controller will send the organized tweets to work on additional features to differentiate between real and fake accounts on twitter and then neglect the fake ones and work on the real accounts to know the fake and real tweets they write, the final step is Training and testing this data on a combination of multi classifiers to produce the best accuracy in knowing the difference between fake and real tweets.

1.4 Business Context

Twitter fake news is a critical subject as twitter is one of the main news sources it can affect every single person life even if he/she is a normal person, company owner or even a country .As fake news can mislead and give false information for a person that take social media a source for his information in his daily life and trust it also can spread a rumor about a company's that can lead this company to lose alot of money as many of people will trust this news and can avoid this company products or a country's state that will lead to misunderstanding about wither it is a safe place to visit or not .This web based application will help them figure out and differentiate between fake and real tweets.

2 General Description

2.1 Product Functions

1) Application

- user enters keyword to search on

- system detect real and fake users and ignore fake ones
- system displays the tweets related to the keyword from the real users only
- User get to choose wither to display fake only , Real only or both together.
- system classify the real and fake news

2) Backend

- retrieve tweets from twitter API
- Pre-processing on data
- Applying sentiment analysis
- obtain user graph for all users
- ignore fake users
- train and test data using algorithms

2.2 Similar System Information

In the past few years some projects were developed to achieve the same goal as the proposed project but with different approaches from both the software aspect and hardware aspect, the following are some of the most relevant to the proposed system :

2.2.1 Identifying Tweets with Fake News:

In this paper they proposed an advanced framework to identify tweets with fake news using statistical analysis for Twitter users, reverse image searching, cross verification of fake news sources and data mining. They assumed that Natural Language processing(NLP) won't be enough to make context analysis as Tweets are usually short and does not follow even the simplest syntactic rules. In this paper they used SVM (as it's widely accepted as baseline classifier especially with binary classification problems) and J48 Decision Tree.

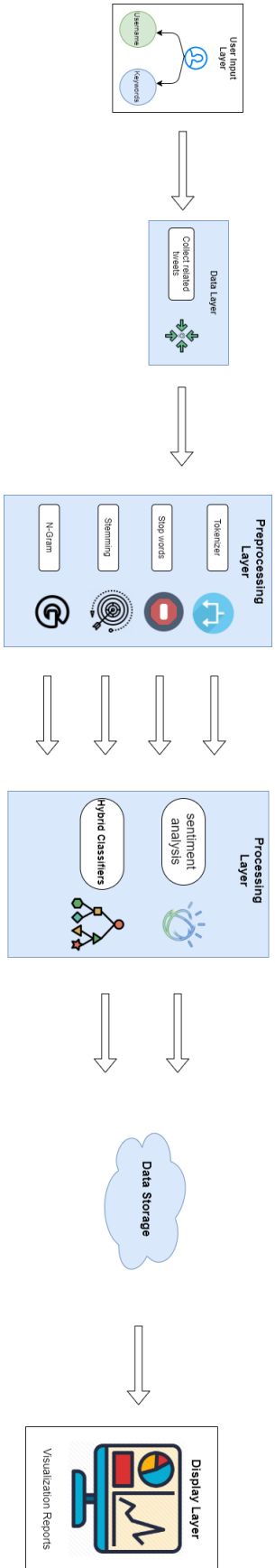
2.2.2 Detecting Fake News in Social Media Networks

The purpose of the work is to come up with a solution that can be utilized by users to detect and filter out sites containing false and misleading information. We use simple and carefully selected features of the title and post to accurately identify fake posts. The experimental results show using logistic classifier.

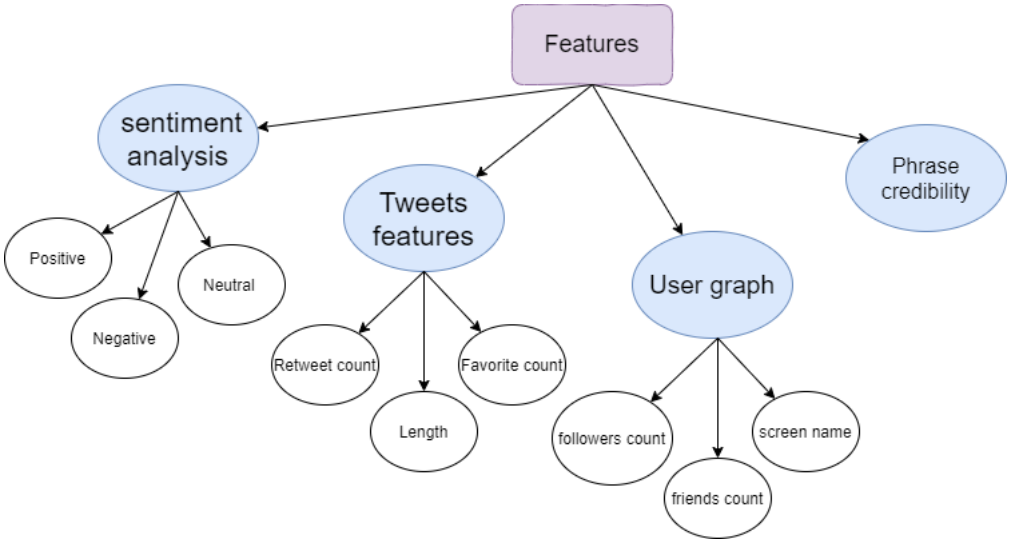
2.2.3 Deep Learning Algorithms for Detecting Fake News in Online Text

In this paper they built a classifier that can predict whether a piece of news is fake or not based only its content, so they added a deep learning perspective by RNN technique models (vanilla ,GRU) and LSTMs that showed the difference and analysis of results by applying them to dataset that we used called LAIR and found out that the GRU is the best followed by LSTM and finally comes vanilla .

2.3 Product Context



2.3.1 Features



2.4 User Characteristics

User needs to have basic knowledge in using social media and websites as our Application will be user friendly and easy to use by any person.

2.5 User Problem Statement

Human ability to differentiate between real and fake news is 54% so this app will help people finding out the fake news on twitter.

2.6 User Objectives

- Differentiate between fake and real news.
- High accuracy in detecting.
- User able to enter some keywords related to the topic and the app search for the tweets by this keywords
- User be able to select the start date and end date to search in.

2.7 General Constraints

- User needs to enter a readable word to search on.
- User must enter the word in the right language.

3 Functional Requirements

3.1 FR1

Title	Retrive-Tweets
Descrebtion	This fuction collect tweets related to the user's keyword
Input	Keywords from get-Keyword function
Action	Search twitter API for tweets
Output	Collection of tweets
PRECONDITION	get-Keyword function dosen't return null
POST-CONDITION	Pre-processing for output
DEPENDENCIES	FR17

3.2 FR2

Title	Retrive-User-Info
Descrebtion	This fuction collect User information
Input	Username from get-username function
Action	Get information from twitter API
Output	User information
PRECONDITION	get-username function dosen't return null
POST-CONDITION	Non
DEPENDENCIES	FR16

3.3 FR3

Title	Pre-processing
Descrebtion	This fuction make data ready to be used by another function
Input	Data retrived (Tweets)
Action	Remove stop words , Tokinization …..
Output	Pre-processed data
PRECONDITION	Real-world **data**
POST-CONDITION	Classfecation for data
DEPENDENCIES	FR1

3.4 FR4

Title	Vectorizer
Descrebtion	This fuction transform data to readable form for classefier
Input	Data retrived (Tweets/user info)
Action	Transform string to pattern
Output	Vectorized data
PRECONDITION	Data contain strings
POST-CONDITION	
DEPENDENCIES	FR3

3.5 FR5

Title	Classify
Descrebtion	This fuction Classify the data and diffrentiate between itself
Input	Data(tweets)
Action	Classefing data
Output	Classefied data / accuracy
PRECONDITION	Train/Test/Split
POST-CONDITION	
DEPENDENCIES	FR4

3.6 FR6

Title	Display-User
Descrebtion	This fuction display if the user is fake / Real
Input	Username
Action	Display
Output	Fake/Real account
PRECONDITION	Check for valid username
POST-CONDITION	
DEPENDENCIES	FR2

3.7 FR7

Title	Display-tweets
Descrebtion	This fuction display if the tweet is fake / Real
Input	Classefied tweets
Action	Display
Output	Fake/Real tweets
PRECONDITION	Select fake, real or both
POST-CONDITION	
DEPENDENCIES	FR5

3.8 FR8

Title	Predict (SVM)
Descrebtion	This function predicts the response for input sample(s).
Input	Array of shape (n_samples, n_features) (X)
Action	Compute log probabilities of possible outcomes for samples in X
Output	Array of shape (n_samples, n_classes)(T)
PRECONDITION	Fit function
POST-CONDITION	Score function
DEPENDENCIES	svm

3.9 FR9

Title	fit (SVM)
Descrebtion	This function indicating whether the fitted values should be computed and included in the model or not
Input	Training vector(X)/Target vector(Y)/Array of weights
Action	Fit the model according to the given training data
Output	An instance of the estimator
PRECONDITION	Kernel function
POST-CONDITION	Predict function
DEPENDENCIES	svm

3.10 FR10

Title	Kernel (SVM)
Descrebtion	This function specified for the decision function
Input	Two matrices of shape (n_samples_1, n_features), (n_samples_2, n_features)
Action	Transforms an input data space into the required form
Output	Matrix of shape (n_samples_1, n_samples_2)
PRECONDITION	SVM function
POST-CONDITION	Fit function
DEPENDENCIES	svm

3.11 FR11

Title	Select-best-result
Descrebtion	This function display data with best accuracy to achieve reliability
Input	Data with their accuracy
Action	Select data with high accuracy
Output	Display only selected data
PRECONDITION	Accuracy function doesn't return null
POST-CONDITION	FR6/FR7
DEPENDENCIES	Accuracy function

3.12 FR12

Title	Accuracy
Descrebtion	This function calculate the percentage of efficiency of classifier
Input	Trained prediction/test prediction
Action	(TP+TN)/(TP+TN+FP+FN)
Output	Percentage
PRECONDITION	To have prediction of trained data and prediction of test data
POST-CONDITION	
DEPENDENCIES	FR5

3.13 FR13

Title	Save-data-in-database
Descrebtion	This function store data in database
Input	Classified data/preprocessed data/vectorized data
Action	Insert in database
Output	Stored data
PRECONDITION	FR5/FR4/FR3 doesn't return null
POST-CONDITION	
DEPENDENCIES	FR5/FR4/FR3

3.14 FR14

Title	Get-Category
Descrebtion	This function get-categories
Input	Labels data
Action	Path labels to the classifier
Output	Labelled data to help the train of data
PRECONDITION	Data be labelled
POST-CONDITION	Path data to FR5
DEPENDENCIES	

3.15 FR15

Title	Get-attribute
Descrebtion	This function get attributes
Input	Vectorized data(Pattern)
Action	Path patterns to the classifier
Output	Data to train on
PRECONDITION	FR4 doesn't return null
POST-CONDITION	Path data to FR5
DEPENDENCIES	FR4

3.16 FR16

Title	Get-Username
Descrebtion	This function get username from user
Input	Valid username
Action	Path username to the search-users-from_api function
Output	Search-users-from_api function
PRECONDITION	User entered a username
POST-CONDITION	Search-users-from_api function
DEPENDENCIES	

3.17 FR17

Title	Get-Keyword
Descrebtion	This function get keyword from user
Input	Valid string
Action	Path keyword to the search -tweets-from_api function
Output	Search-tweets-from_api function
PRECONDITION	User entered a string
POST-CONDITION	Search-tweets-from_api function
DEPENDENCIES	

3.18 FR18

Title	Tokenizer
Descrebtion	This fuction split the data into smaller form
Input	Data(tweets)
Action	Turn data into tokens
Output	Tokenized data
PRECONDITION	Valied data
POST-CONDITION	
DEPENDENCIES	FR1

3.19 FR19

Title	Stemming
Descrebtion	This fuction remove inflected words from data
Input	Data(tweets)
Action	Erase inflected words
Output	Data (tweets) without stemming
PRECONDITION	Valied data
POST-CONDITION	
DEPENDENCIES	FR1

3.20 FR20

Title	Stop-words
Descrebtion	This fuction remove all the stoping words from the data
Input	Data(tweets)
Action	Erase stoping-words
Output	Data (tweets) without stop words
PRECONDITION	Valied data
POST-CONDITION	
DEPENDENCIES	FR1

3.21 FR21

Title	Display-Real
Descrebtion	This fuction display all real tweets only
Input	Input from user (boolean)
Action	Select all real tweets
Output	Display real tweets
PRECONDITION	
POST-CONDITION	
DEPENDENCIES	FR1

3.22 FR22

Title	Display-Fake
Descrebtion	This fuction display all fake tweets only
Input	Input from user (boolean)
Action	Select all fake tweets
Output	Display fake tweets
PRECONDITION	
POST-CONDITION	
DEPENDENCIES	FR1

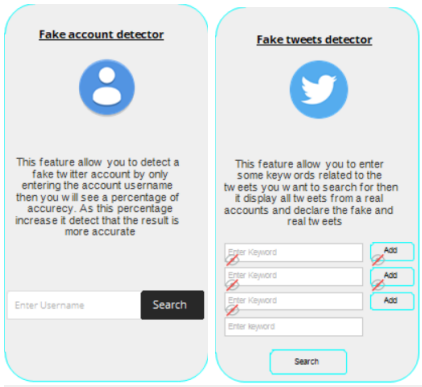
3.23 FR23

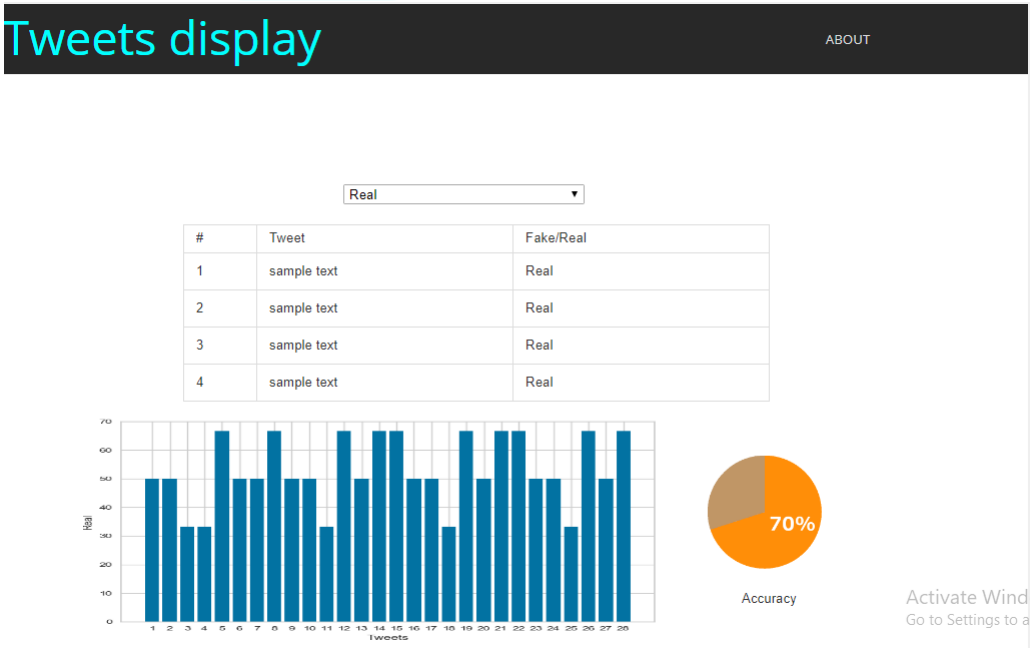
Title	weight
Description	This function specify a weight for every function
Input	3 predicted function output
Action	Show final prediction
Output	final prediction
PRECONDITION	Output for the 3 features
POST-CONDITION	Input data into statistics function
DEPENDENCIES	

4 Interface Requirements

4.1 User Interfaces

4.1.1 GUI





4.1.2 API

- IBM Watson: IBM Watson [1] is a question-answering computer system that offers a suite of tools for natural language processing. The speech-totext and language translator API modules are used within this system.
- NLP: The Stanford NLP [2] library offers language understanding tools. This library is used within the system for part-of-speech tagging.

- Natural Language ToolKit: NLTK [3] is a Python library with a suite of NLP modules. The remove-stop-words and HTML5 parsing modules are used within this system

5 Performance Requirements

Our project required large free memory to save the data and it must be fast to retrieve and write data while running.

6 Design Constraints

Any device that include a browser and must have the connection with the internet to use the website.

6.1 Web Accessibility

The system is designed as a web application accessible through an Internet browser, as such, an Internet connection is required for the various modules that make up the overall system to interact and be fully functional.

7 Other non-functional attributes

7.1 Performance and Speed

Our project works on a large twitter data that can take huge time to detect the fakes news so it's important for our project to be fast and have high performance.

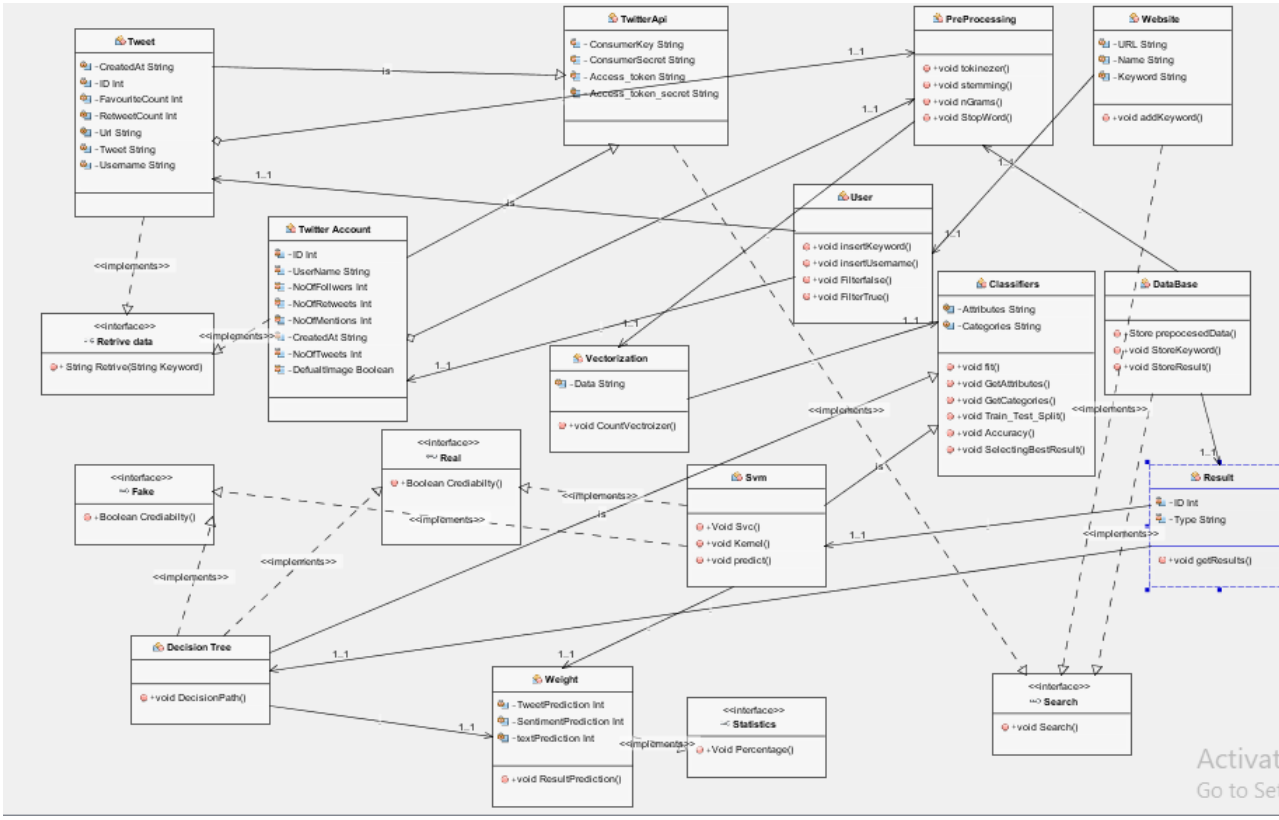
7.2 Reliability

Our Application is built on reliability as it is the most important feature . we need to produce the highest accuracy when detecting fake news and try to make it error free to be a reliable source for end users.

7.3 Usability

Also, this system is easy to be memorized by the user due to the small number of tasks the user will do.

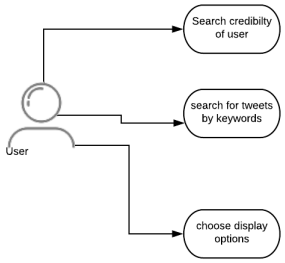
8 Class Diagram



8.0.1 Constraints:

Lists any restrictions upon the general state or behavior of instances of this class.

9 Operational Scenarios



Scenario one:

user can search about a certain user by entering a specific username and detect if it's a fake or real user.

Scenario two:

Moreover he can also search by entering a keyword and it retrieve related tweets and he can also choose how to display those tweets.

10 Preliminary Schedule Adjusted

Project phase:	Start Date	End Date
Announce proposal for students	1 july 2019	15 july 2019
Register to projects	-	till end of july
Lecture writing paper skills	Second week of september 2019	
Proposal Evaluation	First week of October 2019	
SRS Evaluation	Second week of December 2019	
SDD Evalution	Third week of February 2020	
Prototype Evalution	3 days after midterm exams	
Technical Evaluation	1st week of May 2019	
Final Thesis	Last 10 days in June 2020	
Cermony	24 June 2020	

11 Appendices

11.1 Definitions, Acronyms, Abbreviations

- NLP :Natural language processing is a subfield of computer science, information
- API:An application programming interface (API) is a set of routines, protocols, and tools for building software applications
- GUI:Graphical user interface
- NLTK:Natural Language Toolkit [3] (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP)
- Python An interpreted high-level programming language for generalpurpose programming.

12 References

[1]Aldwairi, Monther, and Ali Alwahedi. "Detecting Fake News in Social Media Networks." Procedia Computer Science 141 (2018): 215–22. https://doi.org/10.1016/j.procs.2018.10.171.

[2]Aphiwongsophon, Supanya, and Prabhas Chongstitvatana. "Detecting Fake News with Machine Learning Method." 2018 15th International Conference

on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2018. https://doi.org/10.1109/ecticon.2018.8620051.

[3]Aldwairi, Monther, and Ali Alwahedi. "Detecting Fake News in Social Media Networks." Procedia Computer Science 141 (2018): 215–22. https://doi.org/10.1016/j.procs.2018.10.171.