Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Software Department

Developed Model Based on Text Analysis and Geo-Tweets for Detecting the Spread of Covid-19

A Dissertation
Submitted to the Council of the College of Information Technology, University of Babylon in Partial Fulfillment of the Requirements for the Doctor of Philosophy Degree in Information Technology / Software

By

Iqbal Abdul Baki Mohammed Mahdi

Supervised by

Professor Dr. Ahmed Saleem Abbas Jassim

2023 D.C. 1444 A.H.

بسم الله الرحمن الرحيم

أَلَمْ تَرَوْا أَنَّ اللَّهَ سَخَّرَ لَكُمْ مَا فِي السَّمَاوَاتِ وَمَا

فِي الْأَرْضِ وَأَسْبَغَ عَلَيْكُمْ نِعَمَهُ ظَاهِرَةً وَبَاطِنَةً ۗ

وَمِنَ النَّاسِ مَنْ يُجَادِلُ فِي اللَّهِ بِغَيْرِ عِلْمٍ وَلَا هُدًى وَلَا

كِتَابٍ مُنِيرٍ ﴿٢٠﴾

صدق الله العلي العظيم

(سورة لقمان)

Supervisor Certification

I certify that the dissertation entitled "***Developed Model Based on Text Analysis and Geo-Tweets for Detecting the Spread of Covid-19***" was prepared under my supervision at the Department of Software\ College of Information Technology\University of Babylon as partial fulfillment of the requirements of the degree of Doctor of Philosophy in Information Technology - Software.

Signature:
Supervisor Name: Prof. Dr. Ahmed Saleem Abbas
Date: / /2023

The Head of the Department Certification

In view of the available recommendations, I forward the dissertation entitled "***Developed Model Based on Text Analysis and Geo-Tweets for Detecting the Spread of Covid-19***" for debate by the examination committee.

Signature:
Name: Prof. Dr. Ahmed Saleem Abbas
Head of Software Department
Date: / /2023

Declaration

I hereby declare that this Dissertation, submitted to the University of Babylon - College of Information Technology -Department of Software in partial fulfillment of requirements for the degree of Doctorate of Philosophy in Information Technology-Software has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose sources are appropriately cited in the references.

Signature:
Name: Iqbal Abdul Baki Mohammed
Date: / / 2023

Acknowledgments

Gratitude to God, who helped me to accomplish this hard work. I always had the best belief in my Lord and that He is always supporting me. I went through difficult times and I was about to give up my project, Glory be to God. I suddenly found myself and my hope returned and I continued to work. To Him be thanks and praise be to Him.

I want to express my sincere gratitude and appreciation to my supervisor ***Professor Dr. Ahmed Saleem Abbas*** for his invaluable guidance, supervision, and efforts during my study. I thank him mainly for his patience and kindness.

Sincere appreciation to the members of the department of software and the Information technology collage staff for their help during accomplishing this work.

Special thanks to the examination committee for their constructive suggestions and perceptive comments.

Finally, a special thanks to my colleagues and my friends for their help and encouragement during finishing this work.

Iqbal Al-Saadi

Certification of the Examination Committee

We, the undersigned, certify that (**Iqbal Abdul Baki Mohammed**) candidate for the degree of Doctor of Philosophy in Information Technology - Software, has presented her dissertation of the following title (***Developed Model Based on Text Analysis and Geo-Tweets for Detecting the Spread of Covid-19***) as it appears on the title page and front cover of the dissertation that the said dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on: (12th Jan 2023).

Signature:
Name: Dr. Tarik Rasheed Ahmed
Title: Professor
Date: / / 2023
(Chairman)

Signature:
Name: Dr.Huda Naji Nawaf
Title: Professor
Date: / / 2023
(Member)

Signature:
Name: Dr. Ghaidaa Al-Sultany
Title: Professor
Date: / / 2023
(Member)

Signature:
Name: Dr. Suhad Ahmed Ali
Title: Professor
Date: / / 2023
(Member)

Signature:
Name: Dr. Rafid Sakhban
Title: Assistant Prof.
Date: / / 2023
(Member)

Signature:
Name: Dr. Ahmed Saleem Abass
Title: Professor
Date: / / 2023
(Member and Supervisor)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: Dr. Hussein Atiyah Lafta
Title: Professor
Date: / / 2023
(Dean of Collage of Information Technology)

Abstract

Twitter the mine of gold, it provides incredible valuable information for every working sector, mainly the health sector. Its reputation grown faster as the virus of COVID-19 spreading universally and quickly. Previously, the geographical information provided by Twitter was cuffed. While, during the disease emergence the appearance of Twitter datasets have been released with heavily broad scope, mainly provided with geographical information. This information is needed for detecting the spread of the virus and to identify infected areas. For this dissertation, the demand for a geospatial dataset is crucial. Many datasets have been launched for the public, hydrating them are unsuitable for the purpose of this research because of the lack of some important information.

The collected dataset used several controlled keys (logical statements) to reach to the desired tweets that enabled the authors to find the exact infected cases (self-reported users) among millions of tweets posted every hour regarding this crises, and analyzing their exact location. These tweets were splitted into four groups according to the type of geographical information being provided with the tweet. Each type where exploited and examined to figure out the spatial location for each individual. These are taken from the place dictionary, the location field, the matched location and place information, and finally the text tweet content to be inferred. The authors built a small global gazetteer to investigate the spatial entities and to inspect the exact location for the infected user firstly and his/her social network relations secondly. This will lead to look for spatial close followers and friends of the infected user and identify those who live in the same area of the infected user, then, retrieving their tweets within a specified period of time and extracting their spatial information.

Isolating those who are in the same proxy environment by applying clustering method the k-means algorithm to the longitude and latitude coordinate point for all of them. Those who gain the same label cluster will be isolated as one of the proxy relations. Their tweets collected to be investigated about getting the disease by applying a supervised classification method using machine learning, but before this step the collected dataset have been labeled manually on some special criteria into three classes. The multilayer perceptron algorithm (MLP) has been applied to the text tweets to discover three types and to identify the infected follower. This neural model has achieved an accuracy of 0.88. The positive cases have been collected and showing their distribution on the map of New York City as a case study. And providing the number of infectious cases within that area.

This research presents a complete spatial dataset of infected cases being collected for one year, and a labeled dataset of three classes to show the state of Covid-19 infectious disease. The classification process was able to identify the self-reported infected user other than spam and declarative tweets.

Table of Content

Chapter Three: The Proposed System

Chapter Four: Implementation and Results

Chapter Five: Conclusion and Future Works

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ABBREVIATIONS	DESCRIBTION
API	Application Program Interface
BOW	Bag of Words
CAP	Complete Automation Probability
CDC	Centers for Disease Control and Prevention
CHAID	chi-square automatic interaction detector
CM	Confusion Matrix
CRISP-DM	Cross Industry Standard Process for Data Mining
CSV	Comma Separated Values
DL	Deep Learning
FT	Fast Text
FN	False Negative
FP	False Positive
GPS	Global Positioning System
HTML	Hyper-Text Markup Language
JSON	JavaScript Object Notation
ML	Machine Learning
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
NN	Neural Network
OSM	Open Street Map
OSN	Online Social Network
REST	Representational State Transfer
RT	Retweet
TP	True Positive
TN	True Negative
SNS	Social Network Sites
SVM	Support Vector Machine
TF-IDF	Term Frequency - Inverse Document Frequency
URL	Uniform Resource Locator

List of Publications

1. Iqbal A.B. Mohammed, Ahmed S. Abbas. "Twitter APIs for Collecting Data of Influenza Viruses, A Systematic Review", *2021 International Conference on Communication & Information Technology (ICICT)*, 2021

2. Iqbal AMuffiaki Mohammed, Ahmed Saleem Abbas. "A Geolocated Dataset of COVID-19 Pandemic: Tweets with Location Information", *2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA)*, 2021

Chapter 1

General Introduction

1.1 Overview

COVID-19 has brushed the whole world, this event made people discuss facts about their health, express their fears about catching the pandemic, and share sentiments with their friends and followers. Twitter was and still the best social media platform that became an active source of information, better than traditional news channels and emergency response agencies. Twitter can break the news faster. Millions of subscribers worldwide preferred tweeting on this messaging platform during the spread of the disease for its robustness [1] [2]. One of the reports issued by Twitter refer to the number of active users worldwide of 187 million at January 2021 [3] [4].

The lockdown made people resort to the Twitter platform to check on others' health conditions. They issue nearly 280 characters for each post, they exchange retweets with their followers and friends, also mentions, replies, and quotes [1]. All these facilities made the relationship between any user and his social network stronger, which can give the implication that they might live very close to each other. Spatial information can be provided by Twitter with several attributes. It is also possible to analyze the content of the tweet to infer the spatial location of the user. As the content of the tweet is dissimilar and uneven, no one can figure out what is in the users' mind to be posted [5] [6][[7]. Gathering all this information with the tweet content and with reasonable implementation, this data can be analyzed and processed to obtain valuable information to recognize the infected area with the disease.

A tweet can diffuse through Twitter network using the follower/friend connections. A user can retweet a health related tweet belong to other users then tracking this retweet which spread through a

directed graph of follower/friend inside Twitter network. Analyzing this graph to recognize the nodes of this diffusion to make it possible to follow the infection of a certain disease [8] [9]. This procedure has been used to reach the followers and friends of a certain user to discover the spread of disease via the retweet graph. For this model and unlike all the previous researches, the procedure being used is to reach the follower/friend information via the metadata of the infected user, after implementing several steps and make use of the APIs of the Twitter platform. The type of this model is a Data Science project. This field is a new aspect and it is more than fun to work on.

The intention of this model is to achieve highest credibility in identifying the infected area with exact number of infectious cases and to provide honest information concerning disease spreading. This may be achieved via inspecting tweets of a self_reported infected person first, then investigating his/her exact location address. Retrieving his/her social network the followers/friends and identifying their spatial information to see how close they are to the infected person and isolate those who live in the same area. Finally, checking their tweets to figure out whether they caught the disease or not, within the same period that the user being infected.

1.2 Problem Statement

Twitters' policy in using its spatial information was restricted and this action led to preventing researchers in this field from delving into it. Before the emergence of Covid_19 and its spread as a deadly virus, research using Twitter spatial information was almost non-existent, except for research on natural disasters that identify a known country in advance during scraping data. Three problems may encounter within this field of research; they are:

1- Making use of Twitter spatial information provided is very rare as such information is hardly available within the tweet due to many reasons.
2- When locating an infected user, the social network of his friends and followers is needed to identify who caught the infection and may affected the others or vice versa. This action needs to locate them all and calculate which one has got the shortest distance from the sick user.
3- Locating the infected area is the target of this research. This depends on the number of infection cases within this area.

for this, the following questions need to be answered :-

- How to manipulate the unavailability of datasets with spatial information?
- How to locate infected areas efficiently?
- How to identify infected people, the friends and followers?
- How to label the dataset?

1.3 Challenges

There are many challenges related to the Twitter platform and also related to the classification process. These challenges are:

1) Twitter refused to offer the approval of a development account to scrap the dataset. While this work needs to pull data several times using different APIs. This rejection affects the progress of working on this research.
2) The lack of spatial information that should be found within the metadata of a tweet. As most of the users do not activate their GPS or fill the location field of the user profile with an accurate information. For this most of the datasets stated that only 1% could have this information.

3) It is useless to track the available Covid_19 datasets to use them in this work for several reasons the keywords used do not match with what is required, also the presence of confusion because Twitter allows only ID downloads. In most research, they download the user's ID and not the Tweet ID itself, which complicates the retrieval process.
4) In case of using unsupervised or semi-supervised learning it won't be effective as it is limited due to the lack of rule-based structural grouping. These techniques are still at early stage struggling for efficiency. Unlike the supervised learning requires labeling massive social datasets.
5) The absence of labeled dataset affected the process of classification. This challenge obliged the author to accomplish the task of labeling manually.

1.4 The Contribution of the Dissertation

- This model was able to provide a special dataset collected from Twitter platform with 36602 unique records. All the tweets of this dataset belong to real users stating their infection with corona virus. This dataset has got no spam tweets at all.
- Analyzing the spatial information taken from the metadata of the tweet provided four spatial types to locate the infected user. For each type a dataset was built providing important attributes that assisted the progress of this model.
- With more than 96000 records a new labeled dataset has been built with three classes that show the state of the text tweet whether the user is infected with the disease or not this took the positive and negative labels, while the neutral label has been given to any tweet

that is taking about the disease in general. This labeled dataset would be of great importance for future researches predicting disease outbreak.

1.5 The Aim of the Dissertation

- Building such a model will provide a disease distribution tracker by collecting data from the Twitter Platform to increase the facility to recognize infectious outbreaks and assists global disease dissemination analysis at an exceptional point like the outbreak of the Coronavirus.
- This model will be able to predict the location of the infected user together with the followers-friends' connection in such a way that it can be considered international disease surveillance providing exact quarantine zones in any place tracked.
- This model will be able to predict the infected follower when applying classification method to the text tweets.

1.6 Related Works

Recently and not very far intensive efforts committed to the sector of healthcare. Many researchers developed techniques for the sake of saving human lives. Different approaches for surveillance, tracking, detecting, monitoring, and making predictions before outbreaks of diseases were deeply investigated for the sake of human kinds survival.

Olegas NIAKŠU (2015) [10]: this research proposed an expansion to the CRISP-DM, called CRISP-MED-DM, which reports detailed questions of data mining in medication. The medical implementation scope with its normal encounters is represented with the CRISP-DM mention pattern, intending improvements in the CRISP-DM mention pattern. Moreover, the pattern to estimate agreement with the CRISP-MED-DM

was recommended. The pattern permits estimating and matching to which level distinctive data mining plans are ensuring the process model of CRISP-MED-DM.

Kimberly Chulis (2016) [11]: in this dissertation, the researcher collected tweets to contrast the processes by which a large collection of unstructured disease-related tweets could be converted into structured data to be further analyzed. Three separate data reduction methods were contrasted to identify a method to generate analysis files, maximizing classification precision and data retention. Each of the disease files were then run through a CHAID (chi-square automatic interaction detector) analysis to demonstrate how user behavior insights vary by disease. The investigations yielded insights that offer a modern scope inside the latent Twitter data has as an important healthcare data supplier as well as the distinctions concerned in operating with the data. This dissertation worked on multi-classification using statistics only.

Manan Shah (2016) [12]: this research used a unique pipeline to produce a real-time, correct representation of infectious disease dissemination using Twitter data. He used NLP with ML for processing mass media data. His work gained a correlation coefficient between the Twitter disease distribution of work and the CDC data of about 0.983. This improved the result that made his work the best among all the previous approaches. His plan was on utilizing a disease-linked tweet dataset to ensure how diseases are spreading between people having accounts on Twitter.

Teodoro A. Macaraeg Jr (2017) [13]: this research focuses on the prediction of 21 community diseases: dengue fever, TB, respiratory diseases, typhoid fever, skin diseases, measles, hepatitis, malaria, meningitis, STD, kidney diseases, cancer, diarrhea, hypertension, asthma,

heart diseases, blood-related diseases, diabetes, flu, chickenpox, and ulcer. Demographic, socio-economic, and basic medical profiles were collected randomly from the residents of different barangays in Caloocan City in the Philippines. A total of 5,519 records were collected. Data mining decision tree algorithm is used to predict the occurrence of every disease. Rapidminer 7.3 was used to generate and evaluate the model for accuracy in predicting the occurrence of each disease.

Ryusei Matsumoto, et al (2018) [14]: the proposed framework depicting the epidemics of infectious diseases and employed realistic analysis both to disease recognition and location estimation for correct depiction. They examined their techniques for several infectious diseases and stated that their technique worked well on different diseases. The proposed framework isolated tweets having the disease name and estimates the location of the user. The framework operated a realistic analysis for both, the infection incident, and user locations for further correct approximation. Finally, a map is created for the infectious disease utilizing the outcomes of the location approximation. The framework used very few tweets to predict the location of the user who lives in Japan only. They used only one classification algorithm which is the SVM that takes one of two cases either +1 or -1. Finally, the comparison was conducted only between the accuracy of each disease.

Liang, et al (2019) [15]: this work aimed to explore the dissemination form of Ebola reports taken from Twitter and discover dominant users concerning Ebola tweets. They rebuilt Ebola associated retweeting routs based on Twitter subject material and the follower-followed connections. The analysis of social network was implemented to explore the retweeting form. This work classified types of users and found four categories (influential user, hidden influential user, disseminator, and

ordinary user), also discussed the distribution system, their classifier depends on following and retweeting form. They found that 91% of the retweets from the original tweets were straightly issued. Besides, 47.5% of the retweeting routs had a distance of 1 of the original tweets. They figured out that both influential users and hidden influential users generated more retweets than disseminators and ordinary users. While disseminators and ordinary users were further confident about the virus system for disseminating data afar from their direct followers through influential and hidden influential users.

Ali Alessa and Miad Faezipour (2019) [16]: this research suggested an effectual and correct pipeline using the social networks' data to monitor disease outbreaks and provide early warnings. This pipeline consisted of a classification system to discover flu tweets, it includes three main sections: text classification, mapping, and linear regression for weekly flu rate predictions. They used sentiment analysis for text classification module. Then, they predefined keyword occurrences. Furthermore, they used different classifiers, like FastText (FT) and six typical machine learning algorithms, were examined to figure out the further effectual and correct one for the intended pipeline. This research used several datasets each one for a special purpose, one of them was used as a training dataset as it was labeled manually. These datasets were processed using Map Reduce.

Adel Alshehri (2019) [17]: in this dissertation, the researcher collected tweets for disasters. He presented a complete framework consisting of three sections, clustering-classification-ranking. The first section, he built identical machine learning models to automatically isolate and find subjects within the text and to determine hidden models displayed by a dataset. The second section implemented both binary and multi-class classification model for Twitter data to classify every tweet as related or

unrelated. Further, he classified related tweets into four kinds of public employment these are: informing details, stating negative employment, stating positive employment, and asking for information. The last section, he proposed a binary classification system to classify tweets into either less importance or high importance tweets. He presented an assessment of the efficiency of discovering events utilizing various types taken from Twitter just like, tweet text content, TF-IDF, Linguistic, sentiment, psychometric, temporal, and spatial. This dissertation used the ensemble method to solve the problem of not using a multi-class algorithm by labeling the dataset manually and passing it to three binary classifiers, then the ensemble method did the work of classifying.

Yousefinaghani, et al (2019) [18]: this research, the author developed a Twitter-based data analysis system was developed to automatically follow avian influenza outbreaks in a real-time method. The system was employed to identify irritating tweets and changing news on Twitter. They detected the unrelated tweets, and find those talking about outbreaks in different countries. They tested the latent of Twitter data to denote the date, severity, and virus type from official reports. Furthermore, they examined the use of filtering unrelated tweets that could truly influence the implementation of the system. The system was examined utilizing a real-world outbreak-reporting source. The result was 75% of real-world outbreak tweets of avian influenza were captured by Twitter.

The kind of framework and procedure of this dissertation is quite complicated and conflicts due to the unstructured data. Most of the previous works used a treated dataset or started their work with a labeled one. It is quite obvious that some systems are designed for their dataset and only a few of them could be considered as a general system that can be applied to any other datasets.

The idea of this model is similar to the work of [14]. It is also looking for disease infections and identifying infected users after locating their places. They used several infectious disease and they isolated tweets having the name of these diseases and then estimated the location of the user who live in Japan only. Unlike this model it is working on one disease that spread globally, and looking for an infected area with many infections more than looking of one infection case.

1.7 Dissertation Organization

There are five chapters in the dissertation. Each chapter begins with a brief introductory overview of the subject. The remaining parts of each chapter are as follows:

- Chapter two: presents some general information on spatial information analysis, matching toponyms, using clustering to isolate the spatial closest followers to the user, and applying a multi-class perceptron algorithm to classify followers' tweets.
- Chapter three: Illustrates the proposed model to identify the social network for each user spatially. Many techniques have been used to reach the address of the infected followers of a certain area.
- Chapter four: presents and discusses the experimental results of tracing the social network of each infected user to identify the infected area.
- Chapter five: describes the dissertation-derived conclusions and provides recommendations for future work.

Table 1-1 Summary of Related Works

No.	Refrence	Technique	Dataset	Improvement	Evaluation Measurement
1.	Olegas NIAKŠU (2015)	CRISP-MED-DM Data mining	Any dataset about medicine	Proposed an expansion to the CRISP-DM called CRISP-MED-DM	system
2.	Kimberly Chulis (2016)	Clasification using Statistics & CRISP-DM	The Commercial Firehose API offered by the Tweet Archivist service	Making a comparison between NLP and statistics to analyze tweets about diseases. They found that the CRISP-DM is better than NLP because of the length of the tweet.	1-CHAID (CHAID is a tool used to discover the relationship between variables) 2- Cross Industry Standard Process for Data Mining (CRISP-DM)
3.	Manan Shah (2016)	Classification and clustering, SEIR simulation	Stanfored Spinner 100,000,000 Tweets	A correlation with the CDC ILI distribution (r = 0.983) for disease distribution	Correlation
4.	Teodoro A. Macaraeg Jr (2017)	Dission Trees	A questionnaire used to build a model for predicting the disease with 5,519 record	The prediction gained high accuracy for each disease	Rapidminer 7.3
5.	Ryusei Matsumoto, et al (2018)	SVM	Collected very few tweets	The framework isolated tweets having the disease name and estimates the location of the user	accuracy
6.	Liang, et al (2019)	Classification, Diffusion	Collected from GNIP 18,949,515	Find a relation between Twitter and Ebola disease	Percentage number of the four classes found
7.	Ali Alessa and Miad Faezipour (2019)	Classification, Mapping, linear Regression, Fast Text, Map Reduce	Different datasets one of them labeled manually	Tracking disease outbreak and give warnings.	accuracy
8.	Adel Alshehri (2019)	Clustering, Classification, Ranking	Purchased 4000 tweets	developed a binary and multi-class classification model of Twitter data and to classify relevant tweets into four types of community engagement	F1 score
9.	Yousefinaghani, et al (2019)	NLP, anomaly detection	Collected tweets 209,000	used the context of tweets to find geographical references of events	Accuracy, F1

Chapter 2

Theoretical Fundamentals

2.1 Introduction

Social networking sites (SNS) like Twitter, became very popular to post news, crises, and even express emotions. Tweets are gathered by interested researchers to use it for many purposes may be caused by an event. Events similar to disasters, emergency response, and disease outbreaks. The demand for Twitter data has increased since the outbreak of the Corona epidemic. Twitter data is very important as it affords spatial information that shows the extent of the disease around the world.

The health community is an important product. Healthcare providers should be updated about the health community and disease diffusion and outbreaks touching their societies to make proper decisions at the exact time. Most of the research were conducted to help public health to present superior facilities in a competent mode and at the entire time [19].

2.2 Twitter Platform

Twitter is a significant supplier of data for distinctive scientific researchers. Twitter platform yields both current and historical data. Tweets can be generated from the Twitter platform. Nowadays millions of users issue tweets every day and make these tweets available for researchers and professionals by using special APIs (Application Programming Interfaces) either free of charge or with payment. To access Twitter data, users need to reach these APIs, however, most researchers have no idea how to make this accessible and how to deal with this kind of short text data [20].

Twitter data has multiple purposes and many users made use of it. Some of them use it for business purposes, while others use it for academic research. The extracted tweets have got information called the metadata. This metadata is for both, the user and the tweet. Whenever retrieving

tweets many attributes appeared with the text tweet, they hold many information specially the spatial information.

Twitter has become important because it maintains an API, it is used to state and compose Twitter data. Thus, any user can use it to issue tweets, read profiles, and track the followers' data, but mainly a big size of tweets on certain subjects in particular locations and languages could also be reached properly [21]. Twitter presents several APIs for scraping data from its massive real-time source. The goal of offering open APIs is to encourage outer invention. presenting information distantly through such open APIs allow researchers and developers to not only collect data easily, but also to create invention applications, platforms, and visual interfaces without the need to discover rough information [22].

2.2.1 Access to Twitter's Application Program Interfaces

On the whole, API is a group of commands constructed for developers to join with some form of technology and states how some software sections should cooperate with each other [23]. Before attempting to scrape data from Twitter, there is a need to prepare Twitters' developer account. This can be reached through the ordinary user's account on Twitter which will lead the user to enter the developer application site. Then several private keys are required together with the developer account. Using this account will maintain the continuity of interaction with the APIs of Twitter [24].

There are three types of APIs used for collecting data from the Twitter platform: REST API (Representational State Transfer), the Search API, and Streaming API [25].

• The REST API: various types of Twitter data can be extended using this collection API. The demonstrated data from the graphic interface have been allowed to be transferred by this API. These data can be the

users' profiles, their followers and the people they follow (following), their tweets, trending topics, and so on. It is specified that REST API is the most convenient one for investigating user profiles and their networks. It is also known as the historical APIs as it brings only past tweets and not live tweets. The REST APIs are using a pull strategy for retrieving data [25].

• The Search API: can search for some type of standards. This can be usernames, user IDs, keywords, locations, named cities, etc. the Search API is the entrance to an existent tweet recovered from the history of tweets. This API is somehow faint in retrieving accurate data in relation with the keywords used for tracking tweets. In most cases it gets relevant data [25].

• The Streaming API: this API offer a real-time stream of tweets by creating a stable connection with Twitter's servers. Data can be categorized by ten distinct kinds of factors. The most ordinary ones are keywords, hashtags, users, and locations. This is the API that is most appropriate to collecting information about a topic continuously over a long period (months or years) [26]. The Streaming API is using the push strategy when a request is made, an endless stream of updates is available with no extra input needed from the user.

2.2.2 Java Script Object Notation

The JSON is a lightweight interchangeable file for storing and transforming data and it is an open standard text file format. It is always used when there is a transformation of data to a web page from a server. The code for reading and generating JSON file data can be done using any programming language [27].

JSON file does not depend on any language but uses agreements that are ordinary to programmers of the C family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. The JSON is

built on unique shapes and it is a set of name or value pairs. In several languages, this is noticed as an object, record, structure, dictionary, hash table, keyed list, or associative array. It is an ordered list of values. Virtually all modern programming languages support them in one form or another [23]. The collected data from Twitter can be saved in JSON file format. JSON file could be manipulated either using the same file or transferring the data to a CSV file format or a text Notepad file which is quite similar to the JSON file.

2.2.3 Open Authentication

OAuth is an open standard used to access protected information by the Tweepy library (an open-source enables Python to communicate with a Twitter platform and use its API) as a means to access Twitter API or, with JAVA the Twitter 4J library is used. OAuth passwords are highly protected and these passwords should not be shared with anyone. This will make OAuth more secure and confident.

The passwords provided for any developer account should be added as shown in figure (2.1):-

```
consumer_key = "wXXXXXXXXXXXXXXXXXXXXXX1"
consumer_secret = "qXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXh"
access_token = "9XXXXXXXX-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXi"
access_token_secret = "kXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXT"
```

Figure (2.1): Initializing Credentials

These credentials must be valid and are used for creating the API object, thus it will be the basis of any application to be built. It should look as shown in figure (2.2) [28].

```
# Creating the authentication object
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
# Setting your access token and secret
auth.set_access_token(access_token, access_token_secret)
# Creating the API object while passing in auth information
api = tweepy.API(auth)
```

Figure (2.2): Using Credentials to Create Authentication

Now Twitter APIs can be reached via this authenticated request.

2.2.4 Getting User's Information

When collecting text tweets using one of Twitter's APIs, many information can be derived together and at the same time, describing the profile of the user which contains rich information about the user like the user's real name, location, network activity of the user, profile creation date, number of tweets written by the user, number of followers and friends, etc. These are called the metadata of the user profile [29].

Every text tweet has extra information that also can be requested, called the tweet metadata. Any published tweet has much information like the date of issuing the tweet, the place of issue, the type of the tweet (tweet, retweet, reply), and so on. Figure (2.3) show the details of the tweet.

The topic of Twitter data extraction is of great importance. It includes several trends that interact and affect the process of extracting these data. At the same time, this process suffers from the challenges and obstacles arising from it, as they constitute the strongest part of it.

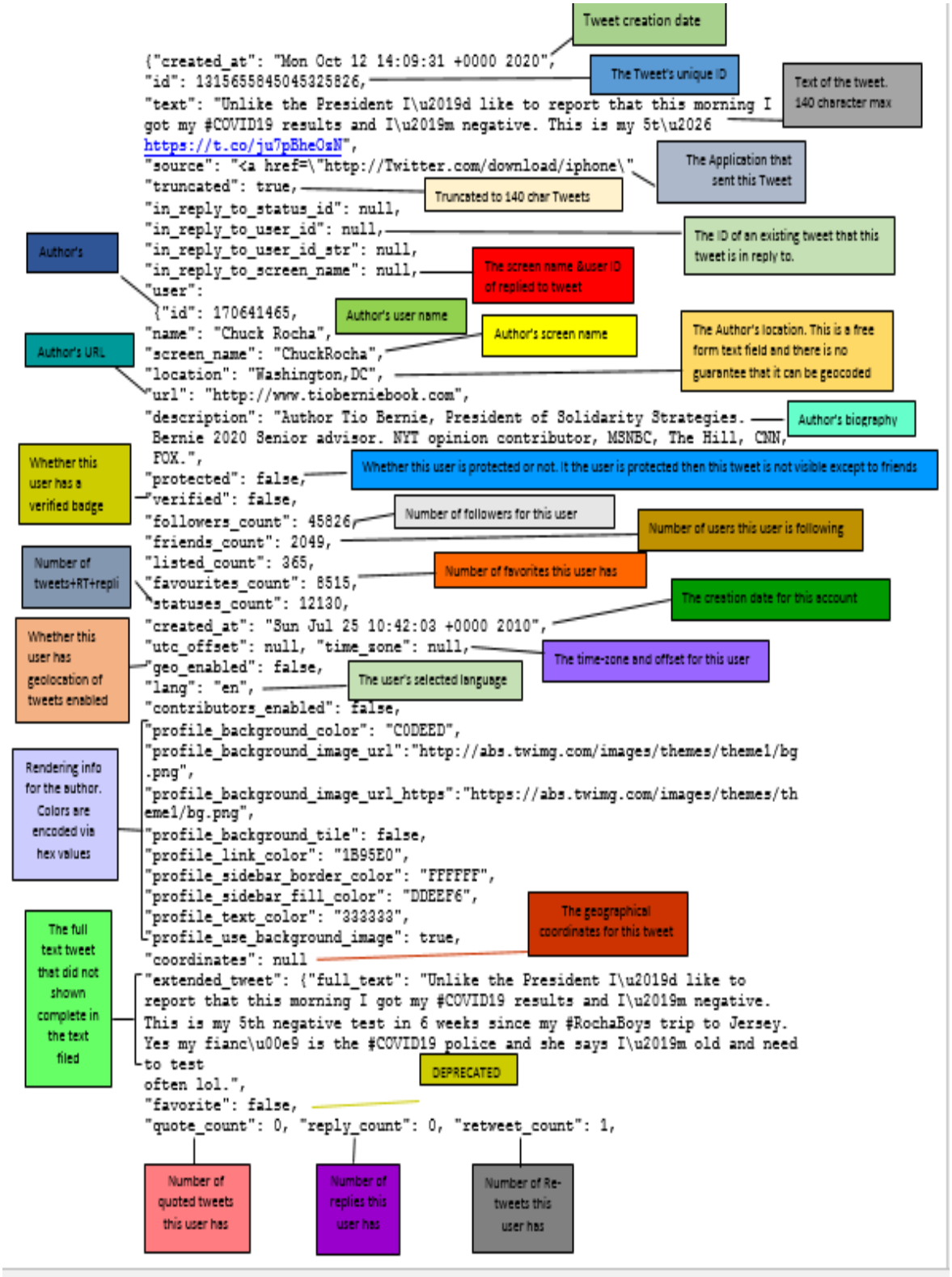

Figure (2.3): Map of a Tweet from the Twitter API [30]

2.2.5 Problems Associated with the Process of Collecting Data

At first glance, the topic of data collection from Twitter might seem interesting and attracts not only researchers but even business owners and individuals to implement the process without any single notice about the originality of this data. The risks that exist on the Internet extended to social media networks, and they are many. This is due to the widened growth of these networks. The Twitter network has evolved dramatically within the last decade. Statistics showed that the number of Twitter users reached millions. Not all the accounts of Twitter are real users, few of them are bots and spammers [31].

The main problem with the Twitter network will be discussed in detail and will demonstrate its disadvantage on the collected data and the scope of its influence on the results of its investigation.

1- Sample Bias

The term bias is happening quite often in statistics which means that when choosing members of a population one of the members will be chosen non-randomly. The same thing occurs when applying the stream API to collect data from Twitter. The collection of data is using a query, while the output samples stated a few tweets that do not match this query [32] [33] [34].

The policy of Twitter is open for sharing data, giving a query, and using the Sample free stream API will retrieve tweets that match the given query. This Sample API suffers from the limited volume it provides, only 1% of the whole stream at a given moment. When this volume overcomes the 1% of the whole stream then the reply is sampled.

A bias may be found in any sampled dataset that skews the data from its source. The problem with this bias is not requested it is unrelated data and it cannot be escaped.

2- Twitter Bots

A Twitter bot is a software that can act as a user account in social media. This bot (just like users) can send tweets, retweets, give likes, and follow other accounts. Its main purpose is to send tweets and retweets about definite topic to achieve its purpose. In many formalities, bots are used on social media networks to uphold certain issues that the creator of these bots could accomplish his intentional goals through using this software [35].

In many different social media networks like Twitter, Facebook, Reddit, and Wikipedia, these bots can be noticed, they either communicate with users or with their content in many uncommon ways. Several statistics have produced numbers to inventory the number of automated accounts on Twitter that represent bots and these statistics is constantly increasing again and again. These bots act strongly for ejecting spam [36].

Social media data can be calculated using metrics. These metrics are of no use any more due to the presence of these bots. It is very easy to create a bot as it costs nothing and it is impossible to count the number of bots on social media networks. It could be imagined that computers controlling Twitter accounts and users may discover that they are following bots and not humans [37].

A cruel intention behind many of these bots. These bots produce a huge amount of noise in order to bias the measurements of the data just like manipulation with sentiment of specific subjects or altering best keywords. Bots have the ability to manage the real time of posted tweets, offering allowing them to regulate (to a precise point) the time that Twitter obtain

their tweet. when bots have the ability to govern (in case their tweets are within the Sample API), it could change the reliability of this stream: bots could transform the statistics of this data stream, disturbing both research and applications that rely on this stream [38].

Botometer API

This API is used to discover bots with retrieved datasets. It is a joint project of the Indiana University Network Science Institute and the Center for Complex Networks and Systems Research. Originally launched in 2014 under the name BotOrNot [39], Botometer API uses a free-to-use classification system, it requires a Twitter user's entire public profile, last 200 tweets, and all recent "@mentions" to classify any account as a bot or human. It uses more than 1,000 traits grouped into six categories (content, friend, network, sentiment, temporal, and user metadata) to provide account classification [40]. It returns a score for the bot for each category as well as an overall score, called the Complete Automation Probability (CAP). Botometer defines CAP as: "The probability, that this account is completely automated, i.e. a bot. This probability calculation uses Bayes' theorem to take into account an estimate of the overall prevalence of bots, to balance false positives with false negatives." [41]. When the CAP Score is 0 it is definitely human and 1 it is definitely bot. [42]. Its versions have been promoted several times there are V1, V2, V3, V4, and Botometer Lite. [39] [40] [43] [44] [45].

3- Spam

With larger user databases in OSNs, are becoming more interesting targets for spammers/malicious users. Spam can take distinct shapes on social websites and is not easy to be identified. Anyone familiar with the

internet has witnessed spam of some kind like email spam, spam on media, newsgroups, etc. [46]

Spam can be expressed as the use of an electronic messaging system to send uninvited loose messages. With the increase of OSNs, it has become a media for disseminating spam. Spammers deliberately posting ads of products to unknown users. Some spammers post URLs as phishing websites which are used to steal users' critical data.

A spammer is an individual or a company that transmits unrelated posts through the internet, typically big numbers of users, for advertising, phishing, spreading malware, etc. [33]

4- Tampering with Sample API

It is quite obvious that social media is a dynamic aspect that has grown up during this decade. However, retrieving data free from bias is a difficult matter. The Sample Twitter API can be inserted with tweets, causing bias in the whole dataset collection. [23]

This bias could be bots or spam tweets. The bias can appear only in the Sample API. The bias appears only in the collected 1% of the whole live stream on Twitter, while the Firehose can provide 100% of the whole live stream on Twitter at a certain cost, depending on the number of desired tweets, keywords, and hashtags for collecting data. Firehose does not have any bias within the collected dataset and that due to the strong filters being used. [23]

Most of the researchers who wanted to detect bias in a Sample API used the Firehose API for collecting the same data collected by the Sample API and finally get two datasets and then make their comparisons [47] [22] [32]. Other researchers used different ways to discover bots within the Sample API. Some used the statistical approach, while others maintained

the keywords and hashtags used for searching [34] [38] [48]. Now how could these bots enter the retrieved sampled data? The answer is by controlling the issued time of tweets, because those bots and spammers have the power to affect the information within the Sample API, successfully inserting bias into this data mouth and producing a great difficulty for users who intend to guarantee that their data is a representative sample of the exact action on Twitter [38].

Scientists are working hard to find better methodologies to avoid bias in the retrieved dataset. The presence of bias is very harmful. Bias affects any analysis implemented using this data, as the bias will disseminate to the results and it will not be a true result.

A reasonable solution has been suggested by [38] stating that API creators can improve this problem through incrementing arbitrary bits within the timestamp slice of the ID, or by producing in advance IDs before issuing them.

2.2.6 Miscellaneous on Collecting Tweets

A short description introduced web crawling and web scraping using the same library Tweepy API. Tweepy is used for gathering tweets from the Twitter platform. The first usage of Tweepy was using the Sample, Search and REST APIs within a few constraints and limitations. But when using Tweepy with crawling and scraping this usage improves a little bit the process of collecting data, however, still several limitations showed in this field. Crawling Twitter may be used in certain situations, it can achieve the number of the retrieved tweets wanted, by allowing the user to state the number of the needed tweets, even millions and it will work properly. Tweepy can work when determining locations and with queries using keywords.

It can be seen that both crawling and scraping are accomplishing the same process but there is a difference between them that crawling works just like Google and Yahoo, it brings everything related, while scraping will search for what is needed only. When using these two processes to collect data from Twitter the first will bring relevant tweets and the second will bring the exact tweets by extracting them from the platform.

The Search API will bring part of the desired tweets, unlike the firehose that can bring 100% of the live stream. The problem with the Search is, not all the tweets are indexed, for this, some or part of the available tweets will be missed from being collected. And this is the main drawback of using this API, although it is free of charge. Using firehose will bring the exact tweets from the stream, but its cost depends on the number of required tweets and the searched keywords.

The Streaming API called Sample or Spritzer is allowed to collect 1% of the live stream. The Spritzer is preferred by most researchers due to its simple usage and does not require any payment. Another API that allows bringing 10% of the real-time stream is the Garden hose. At the moment there is no one able to use the Garden hose because this API was available for the old accounts only, and currently, it is not working anymore.

Instead of gathering real-time tweets, Twitter has made it possible to retrieve historical timeline tweets using either the same API Tweepy or other APIs, but with fewer limitations. For example, the function Get Old Tweets allows retrieving tweets older than seven days ago depending on a specific location. This function afforded to derive several tweets either from seven days ago or within a stated duration. The main advantage of this function is that it does not need to identify credentials just like Tweepy. One of its limitations, it does not retrieve the metadata of the tweet, but only some main information about the user.

There is a major drawback within some of these methods used to collect tweets from Twitter. This drawback is the data in the dataset cannot be represented, and this is a real problem for researchers. They cannot do their statistics when they use the 1% Sample API, because the information retrieved is not enough to be represented. In contrast, there seems to be an advantage from the collected dataset, which is data can be assessed and its quality can be measured. The value of any dataset is useful in the research area and as much as the dataset is valid then it can be considered as a benchmark to be used in analyzing and assessing.

The main difficulties appeared when retrieving data from Twitter and intend to implement analysis for this data, are bias and spam, refer to both of them as noise. Social Media Sites like Twitter sustains from spammers and bots wandering the network to polluted the datasets, have been aggregated by researchers to alter the direction of outputs after analysis to some other paths. Such harmful data ought to be cleared before applying any analysis, and different scopes presented for this purpose. Tampering with data when using the Sample API is very common although its amount is very small (1%), it is very easy for spammers to contaminate this data with their spam. Unlike the Firehose, which is considered free from spam and bots due to its high specifications.

The authors of this proposal tried to explain nearly most of the outcomes of the process of collecting data from Twitter. The first version of the Twitter developer lab in this explanation is adopted, as the second version lab was launched after starting working on this proposal. Furthermore, it is necessary to state that due to the updating of many APIs, some information might seem to look different in their limitations or function and this can be noticed when examining old studies and comparing them with recent ones.

2.3 Spatial Information Provided by Twitter Platform

During outbreaks of diseases, two things governments need to know: Where and How Many? Locating afflicted people and counting their numbers facilitates curative and preventive measures in such cases. Though, fetching the exact place of the infected users is what this study is looking for. Not only looking from to which country they belong, more than that, but it is also crucial to reach their home residence where they declare their infection. Twitter unlike Facebook provides different geospatial information. Two main parts: (1): the user geospatial; this can be taken from the location field in the user profile. (2): the tweet is geospatial; this information is associated with the tweet being issued. The main difference between the two is that the location of the user can be entered with any word and may not be the exact address of the user. For this, the location field should be investigated to identify whether the entrance of this field is a real toponym for all the sets. While the tweet geospatial is the place where this tweet is being posted and it can appear only when the user activates the GPS on his/her device. The problem that may arise here is the mobility of the user. The two locations may not be the same, and this is an ordinary matter. People may register their accounts several years ago and they might move to other places. Identifying the exact place of the infected user needs some judgment according to the status of the case [49] [50].

Spatial information may appear in two forms: the **first** one is a very rare field within the JSON tweet information file. It contains the geo field which provides a coordinate point that gives the exact place of the user. This information comes before the place dictionary as in the example shown in Figure (2.4), and it is sufficient and dispenses with the rest of the available spatial information. [51] [52].

"geo": {"type": "Point", "coordinates": [36.1947786, -115.2964064]}, "coordinates": {"type": "Point", "coordinates": [-115.2964064, 36.1947786]}, "place": {"id": "5c2b5e46ab891f07", "URL": "https://api.twitter.com/1.1/geo/id/5c2b5e46ab891f07.json", "place_type": "City", "name": "Las Vegas", "full_name": "Las Vegas, NV", "country_code": "US", "country": "United States", "contained_within": [], "bounding_box": {"type": "Polygon", "coordinates": [[[-115.384091, 36.129459], [- 115.062159, 36.129459], [-115.062159, 36.336371], [- 115.384091, 36.336371]]]},

Figure (2.4): The Geo Information Contain the Exact Coordinate Point

While the **second** one contains the location field together with the place dictionary field, which is shown in figure (2.5). For this case, when looking for an exact coordinate point the center point of the bounding box with the four corner points will be taken which provides an estimated location point. Though it is preferable to map the spatial data like the full_name or name provided by the place dictionary field. [51] [52].

"location": "Bangalore“ "place": {"id": "07d9ef1da0882001", "url": "https://api.twitter.com/1.1/geo/id/07d9ef1da0882001.json", "place_type": "poi", "name": "National Centre for Biological Sciences (NCBS)", "full_name": "National Centre for Biological Sciences (NCBS)", "country_code": "IN", "country": "India", "contained_within": [], "bounding_box": {"type": "Polygon", "coordinates":[[[77.58028030114127, 13.07136757041265],[77.58028030114127, 13.07136757041265], [77.58028030114127, 13.07136757041265], [77.58028030114127, 13.07136757041265]]]},

Figure (2.5): Both Location Information with Place Information

Twitter provides several attributes to locate infected users as can be seen from the above examples, the geo-coordinates (latitude, longitude), the place field (taken from JSON nested dictionary), the user location

(containing text only, and its free text entry field), and finally the text tweet content, which can allow inferring the location of the user.

The reliability of the location taken from these fields depends on its credibility as the most trusted and accurate one is the geo-coordinates it gives only one exact point of coordinates, then the place field comes next as it contains the full_name of the city and also the coordinates of the bounding box that the exact place lies inside this box which covers a very wide area that cannot locate the exact coordinates points of the place. The third one is the location field, it is known this field should be filled by the user when registering the account and it is up to the user to give the exact location or fill in any other words or just leave it empty. The last source to gain the location when the up-mentioned fields are null, then the text tweet content might have toponym words to locate the infected user [51].

2.3.1 Geocoding and Reverse Geocoding

It is possible to transform coordinates into location addresses and vice versa. When the spatial information available within the metadata of a tweet there are two ways to get the other side of the information. This is either getting a database of location names with the geographical position (for free or for a fee) or it is possible to query a geolocation service provider. Most APIs that offer maps can provide such a service but mainly for a price. Several providers for this service like Nominatim OpenStreetMap API (OSM), Mapbox, and Google Maps Platform [52]. The OSM can provide this service for free but it did not work with us and figure (2.6) shows the result when using it to code very few coordinates' points.

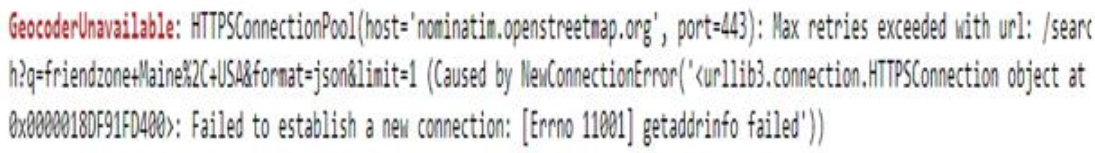

Figure (2.6): Geocoding Location Address Using Nominatim (OSM)

Geocoding means the process of obtaining geo-coordinates from a location name. While reverse geocoding means the process of obtaining a location name from geo-coordinates. [53].

2.4 String Matching Algorithms

To match between the spatial dataset and the spatial fields within the dataset of the tweets the most useful algorithm is the Levenshtein distance and other algorithms that Levenshtein is part of it like the fuzzy algorithm [54].

2.4.1 Levenshtein String Matching Algorithm

Approximate string matching is closely related to editing distance. Levenshtein edit distance is the most popular procedure of edit distance, and often the expression (edit distance) is adapted into it [55] [56].

Having two strings *p* and *t*, where *p* is the (pattern) and *t* is the (text), the edit distance $\delta(p, t)$ between them $(m = |p|, n = |t|)$ is the minimum number of insertions, deletions, and replacements to make *p* equal to *t*.

- **Insertion**: insert a new letter *a* into string *x*. An insertion operation on the string $x = vw$ consists in adding a letter *a*, converting *x* into $x' = vaw$.
- **Deletion**: delete a letter *a* from string *x*. (example a deletion operation on the string $x = vaw$ consists in removing a letter, converting *x* into $x' = vw$).

- **Substitution**: replace a letter *a* in string *x*. (example a replacement operation on the string *x = vaw* consists in replacing a letter for another, converting *x* into *x′ = vbw*). [57]

Algorithm (2.1) of Levenshtein uses the edit distance to fill up the matrix of integer *m* of two dimensions each dimension represents the length of one of the two strings, either *s1* or *s2*. For both strings the edit distances were computed. After executing the algorithm, the two entries *i* and *j* of the matrix should have the value of the edit distance between the the first *i* characters of *s1* and the first *j* characters of *s2*. This activity stated in steps 8-10 of the Algorithm (2.1), taken the minimum quantity of the three main acts (substitution, insertion, and deletion) which is correspond to substitution a character in *s1*, inserting a character in *s1*, and inserting a character in *s2*. [58].

Algorithm 2.1: Levenshtein Distance

Input : Two Strings (*S1, S2*) // *S1*= |*p*|, *S2* = |*t*|
Output: number (m) of matching characters between |S1|, |S2|
Begin ←
1. int m[i, j] 0
2. For i = 1 to |S1| do
3. m[i, 0] ← i
4. End for
5. For j = 1 to |S2| do
6. m[0, j] ← j
7. End
8. for i = 1 to |S1| do
9. for j = 1 to |S2| do
10. m [i, j] ← min (m[i-1, j-1] + if (S1[i] = S2 [j]) then 0 else 1),
12. m [i – 1, j] +1,
13. End for
14. m [i , j – 1] + 1
15. End for
16. return m[|S1|, |S2|]
17. **End**

2.4.2 Fuzzy String Matching Algorithm

Fuzzy string matching is the technique of finding a partial match strings with a given string. This match would never be an exact one. This kind of matching is used when a user commit misspelling of a word or partially enters a word. Though, fuzzy string matching give the opportunity to find the exact word.

The fuzzy string algorithm computes the closeness of two strings to each other. It does not take in consideration the similarity of these two strings. The fuzzy algorithm uses the Levenshtein Distance metric to measure the edit distance between the matched two strings. This metric would decide the closeness of two strings through detecting the least changes required to be done to alter one string into another [59].

Steps of token_set_ratio( ) steps

1. split sentence and remove duplicates
2. create three lists of
3. remainder1 = words that are only in the first sentence
4. remainder2 = words that are only in the second sentence
5. intersection = words that are in both sentences
6. sort the words in the three lists and join the elements to a combined string
7. sorted_remainder1
8. sorted_remainder2
9. sorted_intersection
10. join the strings in the following way:

11.combined1 = <sorted_intersection> <sorted_remainder1>

12.combined2 = <sorted_intersection><sorted_remainder2>

13.calculate the following similarities:

14.fuzz.ratio(sorted_intersection, combined1)

15.fuzz.ratio(sorted_intersection, combined2)

16.fuzz.ratio(combined1, combined2)

17.return the maximum of those similarities

2.4.3 Jaro Winkler Distance

This method measures metrics which is a function of the number of matching characters and several transpositions between the strings. The lower the value the more similar strings are. 0 means full match and 1 means no match [60].The Equation of Jaro similarity between two strings referred to (sim_j) is stated in equation (2.1):

$$sim_j = 1/3 * (m /|s_1| + m/|s_2| + (m-t)/m) \quad (2.1)$$

where m is the number of matching characters. Two characters from s_1 and s_2 are considered matching when they are symmetric**.**

$|s_1|$, $|s_2|$ represent the length of the first and second strings, respectively.

While t is the number of substitutions. Enumerated as the number of matching characters divided by 2 (matched characters with different arrangement order). The equation of Jaro-Winkler similarity refered to (sim_w) is stated in Equation (2.2):

$$sim_w = sim_j + Lp(1 - sim_j) \quad (2.2)$$

where (sim_j) is the Jaro similarity between two strings, s_1 and s_2.

L is the length of the combined prefix in the beginning of the string (this prefix should have at maximum 4 characters).

P is the scaling factor showing how much this score is regulated upwards for having combined prefixes. Usually, the scaling factor p should be 0.1 and should not surpass the value of 0.25.

The similarity between two strings using Jaro-Winkler ranges between 0 and 1, for no similarity between the strings it is 0. While for exact match, the similarity is 1. The distance for Jaro-Winkler can be defined as ($1 - sim_w$). To calculate the Jaro-Winkler distance between string1(mouse) and string2 (mute), equation (2.1) will be used to calculate the Jaro similarity between the two strings $sim_j = 1/3 * (m /|s_1| + m/|s_2| + (m-t)/m)$. where m refers to the number of matching characters. It is obvious that two characters from s_1 and s_2 are assumed similar if they are the same and not far than ([max ($|s_1|$, $|s_2|$) / 2] – 1) character apart. For this case, [max ($|s_1|$, $|s_2|$) / 2] – 1 is calculated as 5/2 – 1 = 1.5. The resulted three matched letters are m, u, e. So, the value of m is 3. [61].

The length of the two strings $|s_1|=5$ and $|s_2|=4$ and t which is the number of transpositions. Taking a different sequence range to calculate the number of matching characters divided by 2. In such a case, three matching characters obtained, but they're already in the same sequence order, so t = 0. For this, the similarity of Jaro could be calculated as: $sim_j = 1/3 * (3/5 + 3/4 + (3-0)/3) = 0.78333$.

Finally, to calculate the Jaro-Winkler similarity (sim_w) as: $sim_w = sim_j + lp(1 – sim_j)$. $sim_w = 0.78333 + (1)*(0.1)(1 – 0.78333) = 0.805$. as long as the result near 1, it can be concluded that the two strings are very similar [61] .

2.4.4 Regex

The best definition for regular expression is a chain of characters that identify a search model, basically for utilize in model equality with strings, or in simple words string matching. [62]. Regular expressions are a widespread way to equal models with chains of characters. It is utilized in many programming languages like C++, Java, and Python.

Example: A regular expression may appear in the following chain of characters to check whether it is an email address or not:

^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})$

2.5 Fetching Followers/Friends Stage

After identifying the location of infected users, it is important to find spatially closest friends and followers to this user to investigate whether this infection affected the others or not. Users within the Twitter APIs are identified by two distinct terms, it is very common to use one of them to get the other:

- The screen_name of the user: refers to the Twitter name having the @ sign.
- The user_id: this is a distinct numeric identifier for each user account, it is an extensive numerical string.

The process of fetching the followers of a group of users is very common looked-for activities in Twitter research, because generating follower/friend networks can offer some very motivating intuitions into a definite group of users who tweet about a subject or hashtag. [63].

Looking for followers' ids of definite user, it is preferable to use the screen name of the user and utilize the API of followers' ids to get their ids altogether. This process has got many limitations where that should be overcome, paying attention to the number of followers that should not be more than 5000 otherwise another procedure should be implemented to bypass this dilemma.

Whenever the id of a user is available it is very easy to collect his tweets using the **Historical Time-Line** chronologically ordered comprehensive collections of single users' social media output are an almost ubiquitous feature of social media platforms (including Twitter, Facebook, Instagram, Tumblr, Reddit, and more). However, the choice to base a research project on user timeline data (rather than query keyword data), is more uncommon and confronts researchers with new methodological affordances and conceptual challenges. [64]**.**

Twitter provides a historical timeline of tweets of different kinds; the Search API can retrieve tweets for the last 7 days in general. This is the standard version of the Search API. Moreover, it can provide the last 3200 tweets for the last month of a specific user regardless of the query criteria. There are the Premium and Enterprise versions. It cannot provide the oldest tweets from more than a month ago. Yet, for a certain keyword, 5000 tweets per keyword can be retrieved. Finally, more limits were raised when acquiring tweets for a definite period. While the Full Archive can retrieve from 2006 until the present endpoint.

The REST API enables access to all Twitter resources like tweets and their information also useful information and many more. A new version of the REST API has been launched by Twitter in 2012 that changed methods and restrictions. The old version allowed 350 requests per hour while the new version distinguished the methods of the REST API

and limited the request number for each method. It stated a limitation for every 15 requests for every 15-minute window.

Firehose can be used not only to retrieve real-time tweets but historical timelines, giving a certain period. While Firehose still retains its qualities that are costly and cover 100%. Also, some sites like Gnip [65] and Sifter [66] provide a full archive of old tweets by registering and paying for big volumes of tweets, and at high rates.

Different functions have been stated to collect historical tweets from Twitter using the Tweepy library. Also there are some extra APIs for deriving timeline tweets with different limitations and constraints like getting Old Tweets.

Nevertheless, Twitter can retrieve timeline tweets using NodeXL and can also make analysis and visualization by using the same tool. [67] [68] [69].

Looking into Tweepy Documentation could find functions that can retrieve old tweets, when someone needs to retrieve his previous tweets he can use the API object function home_timeline( ). This function will pull the most recent twenty tweets. Also when having a user ID and needing to retrieve his past tweets the function user_timeline( ) will be used to pull an inadequate number of tweets per query. The last function is the search which returns tweets that match a specified query. The limitation of the search function is that for every 15 minutes in a window only 18,000 tweets can be retrieved. Some works used this function for collecting reasonable datasets searching for topics or specific language or certain names by improving the code to overcome its limitation like [20] [70] [71].

2.5.1 Spatial Information Extraction

From the file created for the collected tweets of the followers isolate records with location not null and then check the entry content to whether it is a spatial entry or not and keep only spatial entries. Create new columns for longitude and latitude that hold the coordinates points for the spatial entry in the location field [72].

2.5.2 Identifying Spatial Close Followers to the User

To measure the distance between the user and his social network when the point coordinate is available for all of them, the great circle distance is used which is a formula that computes the shortest distance path on the surface of the sphere of two points which assume the Earth is spherical figure (2.7). To use this method, two points of longitude and latitude are needed they are point A and point B. Having the latitude and longitude values it is necessary to convert them from decimal degrees to radians by dividing these values for both points by 180/pi. Pi its value is 22/7. Calculating 180/pi would give an approximate result that is 57.29577951. When the distance between two locations measured in miles, it is preferable to use the radius of Earth with value of 3, 963. Otherwise, when the measure in kilometers, then the radius of Earth is 6, 378.8 [73].
Giving two points A and B, they should be converted into Radians
The value of Latitude in Radians, latitude = Latitude / (180/pi)
The value of Longitude in Radians, longitude = Longitude / (180/pi)
To get the distance between point A and point B use the Formula (2.3) where D is the distance calculated in miles:

$$D = 3963.0 * \arccos[(\sin(lat1) * \sin(lat2)) + \cos(lat1) * \cos(lat2) * \cos(long2 - long1)] \quad (2.3)$$

For distance in kilometers it should be multiplied by 1.609344. Distance in kilometers = 1.609344 * D in miles. [74]

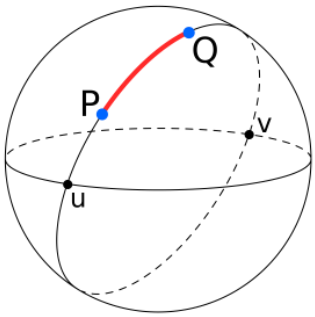

Figure (2.7) Calculating The Distance Between Two Points on the Spherical

2.6 Clustering

K-means algorithm is an unsupervised learning algorithm that does not need labeled data. It is iterative algorithm that tries to partition the dataset into *K* of pre-defined unique non-overlapping subgroups (clusters) where each data point belongs to **one group only**. It manages to make the data points in each cluster as similar as possible, and try to keep the clusters as far as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) reaches the minimum. The less variation gained within clusters, the more similar the data points are within the same cluster [75].

The following are the steps of the K-means clustering algorithm:

1. Identify number of clusters K.
2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.

- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).

- Compute the centroids for the clusters by taking the average of all data points that belong to each cluster.

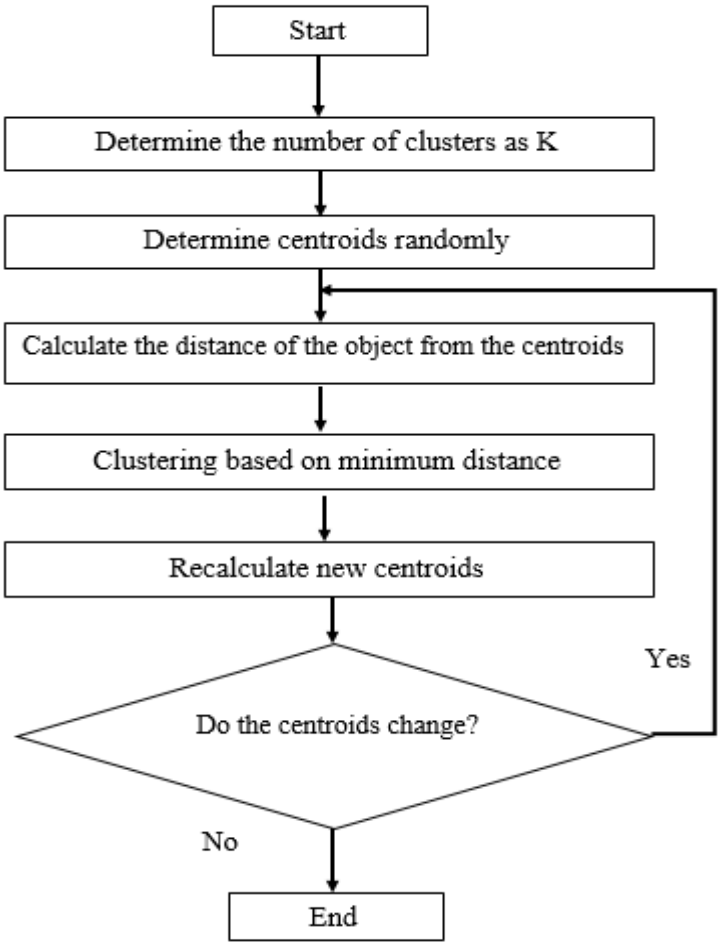

Figure 2.4 : Flowchart of the K-means Clustering Algorithm [75]

2.7 Natural Language Processing

Natural Language Processing is one of the fields of artificial intelligence that deals with analyzing, understanding, and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages. Some its application uses to extract data from text are: Topic modeling, sentiment analysis, summarization, named entity recognition, text classification, and many more [76].

2.7.1 Classification Using Machine learning

Many COVID monitoring systems searched for an active method to text handling and obtaining knowledge from COVID related tweets, producing articles in advance, which can be crucial for outbreak avoidance [77]. Using machine learning to classify the situation of different tweets as self-reported, non-infected, or spam to build a stage for more information helping health sectors to make plans, quarantines, determining the number of cases of the disease in a particular area, and making reports. Using natural language processing (NLP) to do preprocessing for tweets, and classifying using supervised or unsupervised machine learning (ML) algorithms. Classification is very useful for building faster disease surveillance models. Algorithm (2.2) shows the procedure to classify tweets.

Algorithm 2.2: Pseudo Code to Classify Tweets into Three Categories

Input: Collected dataset D combined with the dataset of the followers

Output: Three classes of tweets

```
Begin
    For each tweet where d ∈ D do
      Assign labeling d according to the stated criteria of three categories
      Convert the text to lowercase
      Remove numbers, punctuation, digits and special symbols
      Remove whitspaces, URLs, hashtags, and mentions
      Remove stop-words
      Apply stemming
      Apply tokenization
    End
    Split D into disjoint sets, training (Dt) and testing (DS)
    If training phase then
      Construct the classifier using Dt
    Else
      Apply the classifier on DS        // testing phase
      Get the category c ∈ C for each tweet in DS
      End
End
```

2.7.2 Cleaning and Preprocessing Tweets

The phase of preprocessing is an essential step in text classification, mainly when the text is a very short tweet. This process starts with deleting the special characters, emoticons, hashtags, punctuation, URLs, numbers, mentions, and some unnecessary symbols from the text, all of the above mentioned are irrelevant for analysis principles utilizing the NLP toolkit (NLTK) [78]. Then , converting these tweets into lowercase characters and apply tokenization process [79]. This process is mainly separating the text into a list of words using existing routines in the NLP library [80]. Finally, applying word stemming process by utilizing Porter stemming algorithm, after removing the English stopwords, this step is crucial to decrease tokens to their origins, through removing suffixes and infixes [81].

2.7.3 Feature Extraction

The phase of feature extraction is essential in any classification problem. This process means extracting and creating feature illustrations that are convenient for the type of NLP task needed to be accomplished and the type of model being planned to be used. Vectorizing text is the process of converting text documents into numeric representations. To filter out the irrelevant words, different vectorization techniques are used such as Bag-of-Words (BoW), as well as Term Frequency-Inverse Document Frequency (TF-IDF) [82].

1- Bag of Words (BoW)

It is also called Count Vector, it is considered a text representation disregarding the order of keeping words and their grammar. One of its main characteristics is that it does not consider the length of the tweet. It counts the frequency of each word that appeared in the tweet and saves it in a vector [82].

2- Term Frequency-Inverse Document Frequency

TF-IDF is an essential step utilized for converting the text tweet data into numbers before adding any classification algorithm. [83]. It is a feature extraction technique used to extract weighted features from text data. It provides the weight of each term in the corpus to improve the performance of learning models. [84]. It is implemented in two statistical phases: TF is the first, which is the complete number of words in a document; second, IDF, which indicates the occurrences of the total term in the document. The weight relies on the multiplication result of TF and IDF to quantify the weight and how the term is significant in a given document. TF can be computed as:

$$TF(t \cdot d) = \frac{n_t}{N_{(T \cdot d)}} \tag{2.4}$$

Where n_t refers to the number of occurrences of term t in a document d, while $N_{(T*d)}$ specifies total terms T in that document. IDF of a term shows how significant it is in the entire document. [85] and it can be computed as:

$$IDF = log\ \frac{D}{n_d} \tag{2.5}$$

Here D is the complete number of documents in the corpus, while n_d is the number of documents whenever the term *t* shows. Using TF and IDF, $TF - IDF$ can be computed as

$$TF - IDF\ = TF\ * IDF \tag{2.6}$$

2.7.4 Multi-Layer Perceptron Neural Network

The algorithm of Backpropagation (BP) [86] has acted as a suitable method to train MLP for large kinds of purposes. It is a supervised learning algorithm utilizing feed-forward networks, that benefit from

teacher signals or objective rates. It is a gradient descent procedure and its advantage is to decrease the mean squared error between the objective rates and the actual output of the network. One drawback in this algorithm is that when searching for the minimum error, it might get involve in local minima. The BP algorithm consists of two phases. The first one is the forward propagation step while the second is the backward propagation step. In the forward propagation step, there is no learning and the network will obviously yield an incorrect value at its output before learning. This value is then associated to an objective value and the resulted error is then backpropagated through the layers of the network. This process is the second phase where learning is done. The complete process is then rotated hundreds or even thousands of times relying on the job until the desired convergence, or when the error boundary is accessed. Figure (2.8) shows the network of MLP. [87].

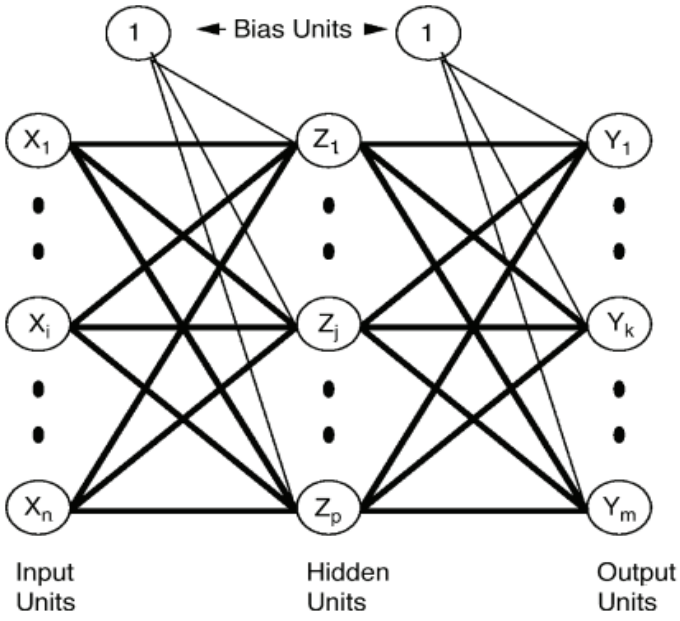

Figure (2.8): MLP Model with One Hidden Layer [87]

1- Architecture of Neural Network

MLP is a feed forward artificial neural network comprising of multiple layers of perceptrons. A perceptron is a basic unit of a neural network consisting of weights and biases [88]. Training of MLP includes finding the best set of weights and biases to make the best prediction. A perceptron is represented by the equation:

$$M = f\,(b+\sum_{i=1}^{n} x_i\ w_i\ ) \qquad (2.7)$$

where *b* is bias, *x* is the input vector, *w* is the weight, *n* is the cardinality of vector *x*, where *i* ranges from 1 to *n*. MLP has been shown to be a reliable model for classification of short text like tweets [89] [90]. Several of deep learning models are available for this architecture. A MLP is a neural network with multiple layers of artificial neurons stacked one after another. After preprocessing, tweets are transformed for each word into its corresponding vector. The outputs are flattened or pooled before passing to fully connected layers. Pooling is preferred to reduce computations. This layer is also called dense layer, that all inputs are connected by weight to all outputs. It is just like a matrix multiplication operation. While Pooling is used to down-sample the incoming vectors. The average of all the features in a pool should be taken in the global average pooling. [91].

The input layer of the model feeds the bag-of-words vector. The sentence vectors thus obtained as a 2D matrix of each. A fully connected layer with the Rectified Linear Unit Activation Function (ReLU). ReLu is represented as follows:

$$f\,(y) = max\,(0, y) \qquad (2.8)$$

During model training, the weights are updated aiming to minimize the loss function calculated as:

$$L(w) = \frac{-1}{n}\sum_{i=1}^{n}[ti\log(ti\hat{}) + (1-ti)log(1-ti\hat{})] \qquad (2.9)$$

where, $ti\rightarrow$ true label, $\hat{}ti \rightarrow$ predicted label, $w\rightarrow$ Model parameters and i ranges from 1 to n, i.e. the number of classes or target variables. The Output layer of the model uses a Softmax activation function, primarily employed for the multi-class classification task. Softmax normalizes the input values into a vector representing the probability of each class which adds up to 1. Here probability is calculated for a class $y = k$ with total classes being m. This is calculated by expression:

$$P(y=k|\ \theta^{i}\ ) = \frac{e^{(\theta i)}}{\sum_{i=0}^{m} e^{(\theta_m^i)}} \tag{2.10}$$

The standard exponent function is applied to each element θi when θ is input parameter represented by Equations (2.9) and (2.10). Here j is the index of summation which reaches n, the cardinality of vector θ. Also

$$\theta = \mathrm{b} + \sum_{j=1}^{n} w_{tj}\ x_j \tag{2.11}$$

Or

$$\theta = \mathrm{b} + w_i^T x \tag{2.12}$$

where x is feature vector of i^{th} training sample and w_t is the weight vector and b is bias. The models which show low bias and high variance are said to overfit the data. This may result in very high error on test data. Deep Learning models trained on smaller datasets are more likely to show high variance, trying to observe patterns that do not exist. This results in poor accuracy over test data. Models based upon deep learning are likely to overfit when trained on small data sets. This is because the model is not well generalized and neuron weights are adjusted just to fit the underlying few training examples. Therefore, data augmentation is often used to enhance the size of training data for a deep learning model for robustness, better overall performance and model generalization to reduce over-fitting [92].

Deep learning exploits algorithms and tries to mimic the way human brain works. It is a type of artificial intelligence [93]. Dropout provides a computationally cheap and effective method for regularization and reduction of overfitting. In dropout, some neurons are randomly excluded during each training cycle based on dropout rate which is a regularization technique for neural networks. The Adam optimizer performs better in terms of accuracy and speed.

1- Performance Metric for MLP

In general, for binary classification, several different metrics can be used to evaluate the efficiency of the MLP algorithm based on considering accuracy, f1-score, precision, and recall. The intention of these controls is based on calculating the confusion matrix. This matrix encapsulates the number of examples properly or improperly predicted by a classification model, as discussed in more detail in the table (2.1).

Table (2.1) The Confusion Matrix for Binary Classification

		Predicted Class	
		Positive +	Negative -
Actual Class	Positive +	*f*++ (TP)	*f*+- (FN)
	Negative -	*f*-+ (FP)	*f*- - (TN)

Ratings of accuracy are of a maximum rate of 1 and a minimum rate of 0. To calculate the evaluation metrics, it is necessary to use the values of TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). To predict the model correctly as positive TP prediction is used, while for negative class, correct prediction TN is used. But when

the model incorrectly predicts the negative sample as positive the FP is used, while FN is the example of the positive class expected as negative.

Accuracy represents the ratio of true predictions to total predictions. Sensitivity indicates the ability of a model to truly expect a sample of positive class while precision is utilized to estimate the correctness of a classifier. Taking precision and recall alone may not be suitable to estimate the pattern, so an F1 score is used that combines both precision and recall. [94].

Accuracy gives the proportion of the total number of correct predictions:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.13)$$

Sensitivity, recall, or the TP rate (TPR) is the fraction of positive values out of the total actual positive instances (i.e., the proportion of actual positive cases that are correctly identified):

$$Sensitivity = \frac{TP}{TP+FN} \quad (2.14)$$

Precision or the positive predictive value is the fraction of positive values out of the total predicted positive instances. In other words, precision is the proportion of positive values that were correctly identified:

$$Precision = \frac{TP}{TP+FP} \quad (2.15)$$

The *F*1 score, *F* score, or *F* measure is the harmonic mean of precision and sensitivity it gives importance to both factors: [95]

$$F1\ score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (2.16)$$

Algorithm (2.3) shows the calculations of the metrics report.

Algorithm 2.3 Confusion Matrix

Input: Features: (f_1, f_2, f_3, …, f_n) // Vector of features
Labels: (l_1, l_2, l_3) // Classes
Output: Precision, Recall, F-Measure, Accuracy // Metrics Report
Begin

1. TP = 0, FP = 0, TN = 0, FN = 0
2. Len = length of Features
3. For s = 1 to len do
4. IF (Predictions[s] equal to 1 and Labels[s] equal to 1)
   TP = TP + 1
5. IF (Predictions[s] equal to 1 and Labels[s] equal to 0)
   FP = FP + 1
6. IF (Predictions[s] equal to 0 and Labels[s] equal to 0)
   TN = TN + 1
7. IF (Predictions[s] equal to 0 and Labels[s] equal to 1)
   FN = FN + 1
8. Precision = TP/(FP+TP)
9. Recall = TP/(TP+FN)
10. Specificity = TN/(TN+FP)
11. F-Measure = 2 * ((Precision + Recall)/(Precision * Recall))
12. Accuracy = (TN + TP)/(TP+TN+FP+FN)
13. **End**

Chapter 3

The Proposed System

3.1 Introduction

In the field of public health, SNS offers an effective source for disease observation and a convenient method of communication to prevent disease outbreaks. Utilizing the data of SNS to find the dissemination of epidemics in societies may lead to obtaining immediate alerts. Users of SNS may act as initial sensors that present data to be analyzed for early trend detections and predictions. Techniques were developed for analyzing the stream of social media data that being used to have real-time analysis to provide fine facilities. This model is based on the content of users' tweets that adapt to the crises of Coronavirus. The overall objective is to build a well-defined system pipeline that can identify the spread of this disease in any region in the world.

This model presents an approach that can identify the infected area with disease outbreak through identifying the position of the infected user and the position of his close relation friends and followers network. The infected user is the target to be fetched and his friends and followers are under scope whether they are infected or not, taking into consideration their closeness to the target user spatially.

3.2 Model Description

Most studies about Coronavirus use the Twitter microblog as this platform can provide location information. Locating infectious users is the target of this research. Fetching sick users and their related social network persons will help in locating the spread of the disease and giving the number of infected people in that area. During the disease outbreak, people used to connect with their family and friends, discuss the latest news about the disease, share their symptoms, and look for answers to their questions. Geo-locating each patient is the focus of the subject of this topic, not only that locating neighbors who are in contact with this patient was also being

investigated. Therefore, the workflow of this system focused on geotagged tweets with GPS coordinates either available with the tweet or being inferred. The proposed system is shown in Figure (3.1).

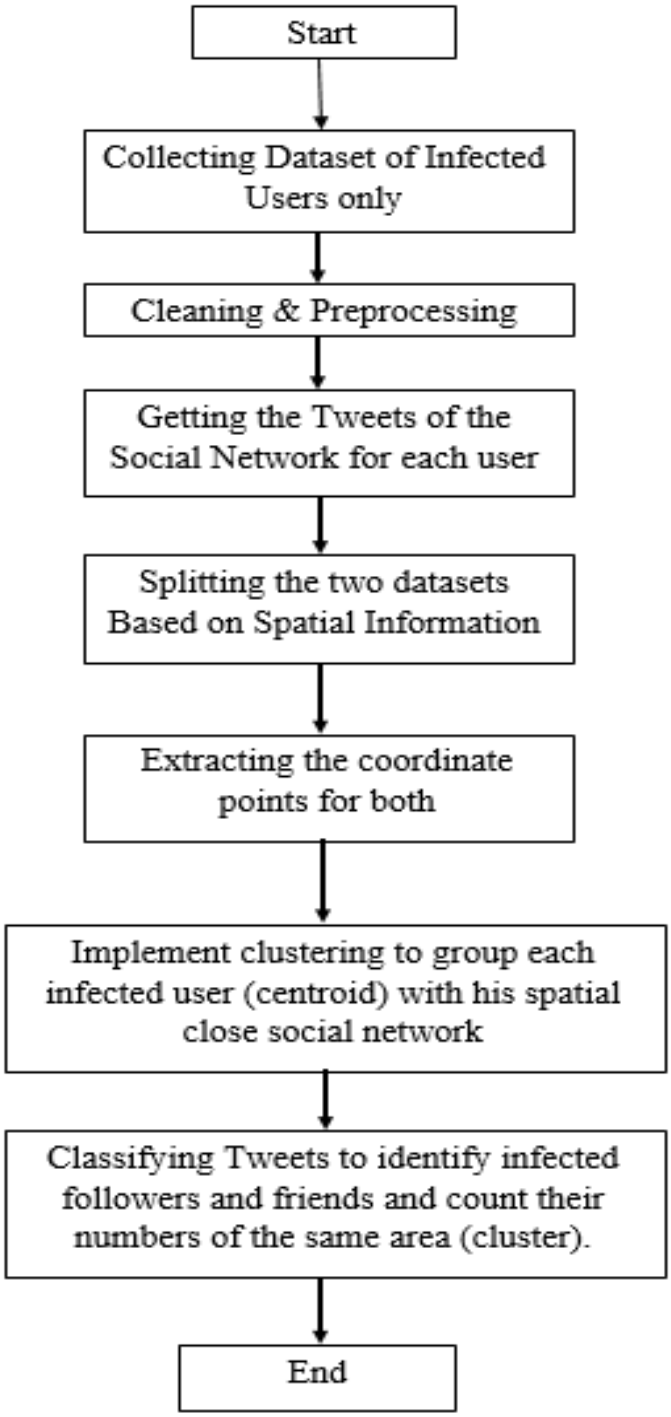

Figure (3.1): The Flow Chart of the Model

3.3 Dataset from Twitter

Crawling data from Twitter is one of the most important processes for this dissertation. Different APIs were used to collect data according to the type and purpose of the needed data. Figure (3.2) shows the steps of implementing this step.

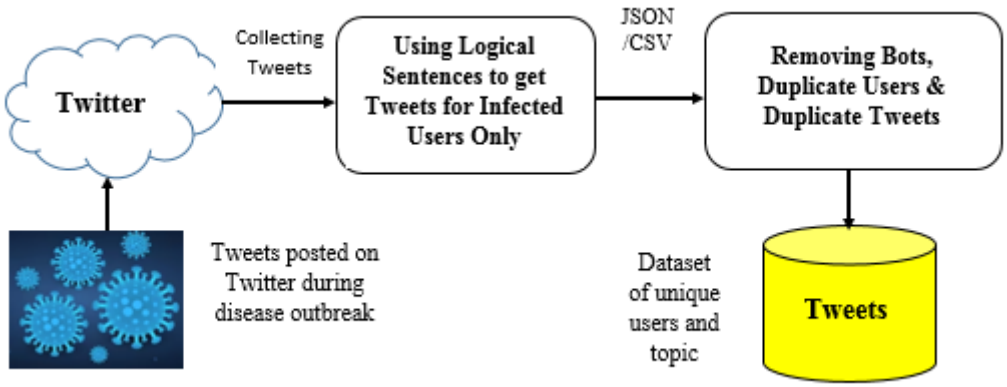

Figure (3.2): The Architectural Overview of Collecting Tweets

3.3.1 Data Collection

The emergence of Covid-19 virus prompted many researchers to collect data about this event and keep them in a state of anticipation in light of the mutation of this virus and its spread all over the world. Thus when events arise about this disease, more scientists insisted on collecting the largest possible number of datasets. While the number of studies using social data is growing rapidly, very few of these studies transparently outline their methods for collecting, filtering, and reporting those data. Keywords and search filters have been applied to social data, form the lens through which researchers may observe what and how people communicate about a given topic. Without a properly focused lens, research conclusions may be biased or misleading [96].

For this, different trending terms have been examined to identify the best filtration search keywords, hashtags, and logical statements and

phrases related to the Covid-19 pandemic. The target is to collect self-reported tweets that identify infected users explaining their health situation or describing the pandemic symptoms and how they feel. Though, it is necessary to avoid any declarative text tweets as such no preferable tweets should be considered spam.

It was found that using logical statements and phrases retrieved the best filtration tweets and the set collected using these statements is the perfect one for classifying this dataset. While using hashtags and keywords related to the pandemic gave different kinds of statements that are irrelevant to this dataset. The following examples have been used to collect the data with the help of logical operators:

"I" AND "have" AND "high" AND "fever", " I" AND "have" AND "dray" AND "cough", "I" AND "have" AND "sore" AND "throat", "I" AND "have" AND "diarrhea", "I" AND "have" AND "difficulty" AND "breathing", "I" AND "suffer" AND "from" AND "loss" AND "of" AND "taste" OR "smell", "I" AND "got" AND "the" AND "pandemic" OR "quote", "I" AND "got" AND "covid" OR "covid19", "I" AND "suffer" AND "from" AND "tiredness".

On the other hand, this study examined hashtags and keywords (the same hashtags without #) to represent the topic concerned using the following:

'#Infection', '#cold', '#flu', '#high-fever', '#difficult-breath', '#headache', '#dry-cough', '#tiredness', '#loss_of_test_or_smell', '#sore_throat', '#Epidemic', '#Pandemic', '#outbreak', '#COVID_19', ' #Covid_19', #Coronavirus', '#immunity', '#face-mask', '#self-quarantine', '#shelter-in-place', '#self-isolation', '#Community-spread', '#Incubation-period', '#vaccine', '#lockdown', '#National-Emergency', '#SARS-CoV2'.

Another useful operation added during scraping Twitter before filtration is to withdraw retweets and replies from the collected set. The collected set of tweets from all over the world using the English language only.

This set is free from retweets and replies because they are considered spam (RT is only a repetition of a tweet) and they are of no use for this proposal. Three benefits can be obtained when using this method:

1) To ensure collecting many tweets with fewer mentions.

2) Necessary to avoid retweets as they have no location information at all.

3) The subject matter of this dataset is focused on a self_reported user account describing the symptoms of the disease or how the patient feels. This may lead directly to avoiding retweets and replies.

The first collection utilized the Search API to collect the first dataset and as follows:

- The credentials of the developer account need to be called with the help of the Tweepy API.
- The English language needs to be selected (to retrieve tweets in English.
- Query applied using keywords, hashtags, and/or logical sentences.
- The period of retrieval should be given starting from 31th August 2021 backwards till 1st September 2020.
- A condition was applied to exclude the retweets and replies.
- Then tweets being retrieved and saved in a JSON file.
- the necessary attributes (post time, user name, user ID, full text, location, followers count, Friends count and place) have been taken from the JSON file and being saved in a tabular file format called CSV.

3.3.2 Cleaning Dataset

An API is being used recently with Twitter called the Botometer API to identify bots in any dataset collected from Twitter. This process needs to have a bot key called the Rapid Key1 and needs to use Twitter credentials and import the Botometer API for this purpose. The last version has been used V4 due to the last upgrade for the Botometer API. Three percentages were adopted: 75%, 60%, and 50% measuring how many of these accounts behaved as an automated account. While the second phase of preparing the dataset is removing all spam tweets from this set this includes removing duplicate users and tweets, (RT, replies if any).

It was necessary to get rid of duplicate tweets and users and keep only one tweet in the dataset for each user.

3.4 Getting the Social Network for each User

To reach any user's followers and friends there is a need to collect some Tweets, but in the past time before collecting the tweets of the users. The REST API and Search API are used for this purpose to collect historical Tweets. Figure (3.3) shows the steps of getting the social network for each user.

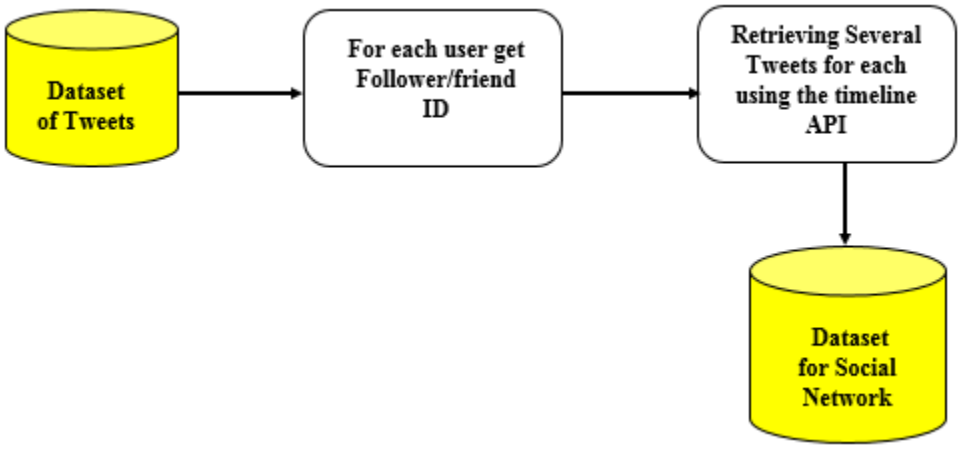

Figure (3.3): Flowchart Showing Procedure to Get the Social Network for each User

3.4.1 Getting Followers/Friends IDs

It is well known that fetching the followers and friends for any number of users is a very interesting action in Twitter research. This action creates follower/friend networks that call the attention of those users who tweeted about certain topics or hashtags. But in the case of this proposal, the interest is about their locations, the social network will investigate the followers'/friends' locations among the user location.

To get the followers or friends ids of a certain user the credentials of the developer account should be called first, then using the screen name of the user. An API should be called to get the follower's ids for this screen name. The output would be a list of ids of all the followers for this screen name. From these ids, it is possible to get the user name using the API get_user. For this proposal, the need is to get followers ids for all sick users, though a list of users' screen names is being used and an empty list for collecting followers' ids for each user screen name by iterating the same procedure for each user screen name. The output would be a list of followers for each user of the same index. Then this list is transformed into a column for the same data frame that contains all the user information. The same thing would be done to collect the ids of user friends. Algorithm (3.1) shows the steps of this process.

Algorithm 3. 1: Collect IDs of Followers Using Twitter REST API

Input: Consumer Secret I, Consumer Key II, Access Token III, and Access Secret IV // Twitter API access keys

Get_User API, Follower_Ids API // APIs from REST

User's screen names in a list // taken from Tweets File

Output: Followers_List // List of ids for followers

Begin

1. Call Twitter API (II, I, III, IV) //user access account and Twitter access key
2. Create empty list (Followers) // to collect the ids of followers
3. Call user list (user_list) // list contain screen names of users
4. For each user in (user_list) use the Get_User API to get the followers_id using the screen name of the user
5. Add Followers to Follower_list // fill the empty list with followers ids

End

3.4.2 Retrieving Tweets for each Follower/Freind

From the step before the followers' ids were retrieved, this will give the chance to retrieve the screen names by using the get_user API. This API can work in the opposite direction and get the id from the screen name account. After getting the screen names, it will be possible to retrieve tweets of followers by using the timeline for each follower. The timeline API provides tweets from this moment backward seven days, unless a given date might be encountered to start retrieving tweets. In the case of this model, timing is very important to retrieve for each follower. This model is looking for an infected user that tweeted about getting the corona virus in a certain date taken from the attribute (created_at), from this date backward seven days (according to the health instruction that the person might need seven days to start suffering from symptoms of corona virus after getting infected) should be retrieving tweets for all his followers, to investigate whether he got the virus from one of his spatially close

followers or might one of them got the virus from the user himself. In this case, the need to identify two dates is very necessary for retrieving tweets (the start date and the end date).

This method is unavailable with timeline retrieving API, though to implement this procedure, the start date and end date are fixed to a certain date time starting from the created_date of the user tweet and ends backward for seven days to retrieve tweets for one of the followers of the sick user. The following steps show the process of retrieving one follower's tweets: -

- In the beginning, credentials will be called and activating tweepy API using authentication.
- Start with the first username taken from a list called username.
- Stating start date and end date by using datetime API.
- Initializing screen_name of follower.
- Initialize an empty list called tweets.
- Starting the user_timeline API by using the screen_name and collecting tweets as temporary tweets.
- For each tweet in the retrieved temporary tweets check the created_date: if the created_date is between the start date and end date add this tweet to the list of tweets.
- Check the last retrieved tweet (in temporary tweets list): in case its created_date is more than the start date then states the created_date of this tweet and instruct to fetch some more tweets.
- Call the user_timeline API again to fetch tweets giving the same screen_name, state the last retrieved tweet_id to continue for the second list of tweets. Keep retrieved tweets in the temporary tweets list.

- For each tweet in the temporary tweets list if the created_date is between the end date and start date then add this tweet to the tweets list.
- Save needed attributes of the tweet with the text tweet (like location and place) in an Excel file to convert it into a CSV file.

The number of tweets to be retrieved depends on many things as the user timeline contains up to 3,200 tweets. For each request to this API, it will give 20 results per page. For this proposal, the count parameter did not show up as it could be set to 200 tweets per page, though the 20 results were quite fair to retrieve the spatial information, and also it may contain within its text tweets information about the disease to be classified in some next stages. The timeline of any account should contain not only tweets but also replies and retweets. The line (include_rts=False) was used to avoid retweets in the result set.

For each follower, all his tweets are collected and saved in an Excel file. For all followers of one user, their tweets were collected in one file and each follower was identified by his ID and screen_name with the rest of the attributes. This file saved in a CSV format.

3.5 Splitting Datasets Based on Spatial Information

In this section the main dataset and the Followers dataset is going to be splitted into four groups according to the availability of the spatial information within the attributes of each tweet information. Figure (3.4) shows the steps of this splitting together with analyzing each group to find a point coordinate for each record in the dataset.

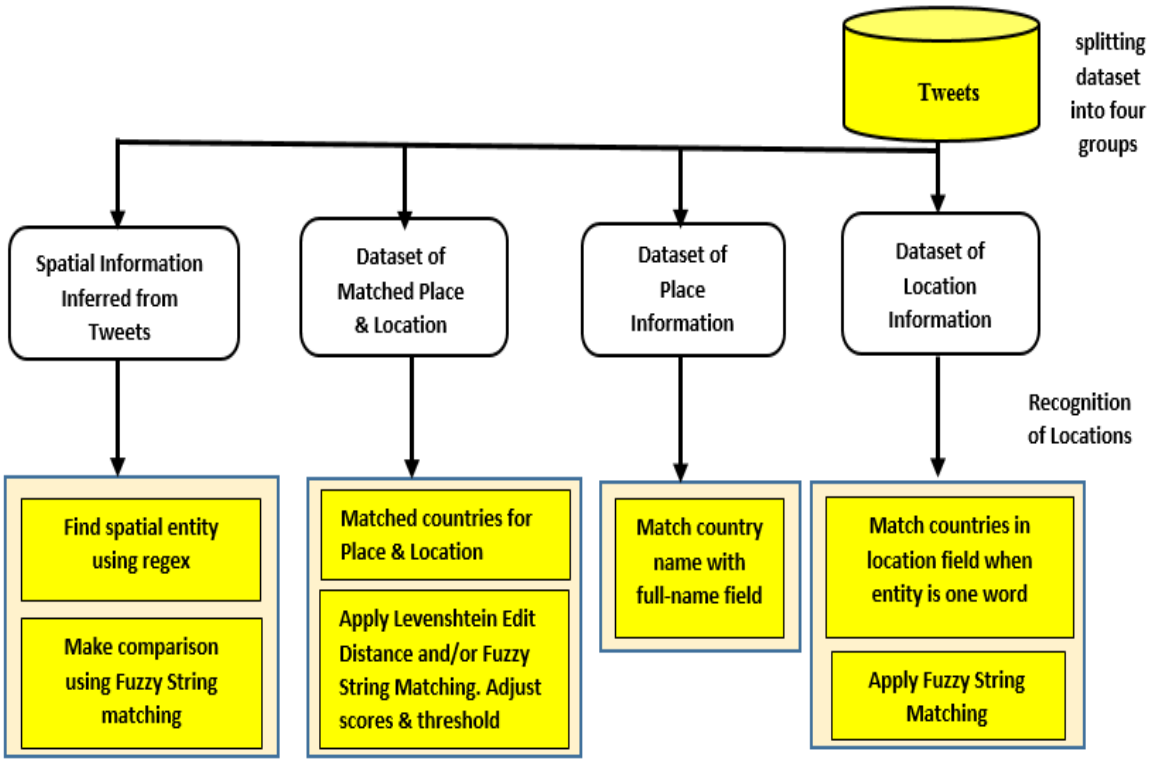

Figure (3.4): Splitting and Analyzing the Dataset Based on Spatial Information

3.5.1 Extracting Spatial Information

The mechanism used here can achieve the desired goal of this research in making the spatial information for every tweet available. In every dataset, the available information on the geospatial features is rarely encountered as they are not very well represented. Not only that, the structure of the place information as a nested dictionary is very difficult to be converted into a table using programming. The dataset being processed here is meant to gain a geospatial feature that is represented as a point of latitude and longitude for each tweet starting from the most accurate location till reaching the inferred location. When two points of coordinates are available, then distance can be calculated.

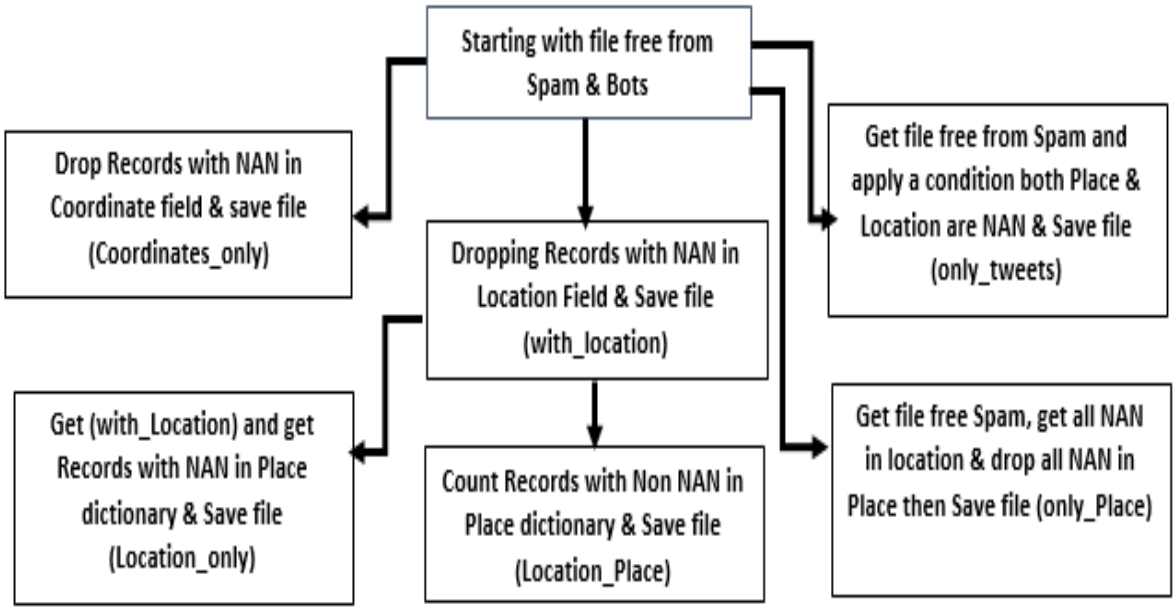

Figure (3.5): Splitting Dataset into Five Files According to its Spatial Information

Extracting user location presents five modes according to the information within each tweet. The main dataset converted into CSV file that contain the unique tweets with several columns that contain different attributes. The spatial columns are the 'Location' column and the place dictionary divided into sub-columns contain 'Name' (means the name of country), 'Full Name' (gives the detailed address), 'Code' (state the code of the country like Canada= CA), and finally 'Country' which gives the name of the country. There are some other attributes within the place dictionary have been neglected due to its unimportance. For each record and in any column when the information is not available it is stated as **NAN**.

The dataset is free from spam and bots. Its file has been used as a main file. The splitting process is as follow: **First**: isolating users accounts having information in their location and place fields. **Second**: isolating accounts having information in the location field only. **Third**: isolating accounts having no information in both fields the place and location. This will be processed to infer the location from the tweet text. **Fourth**: isolating

accounts having only place information from the tweet metadata. **Fifth**: isolating accounts having information in coordinates within the geo field (this is a very rare case).

This step needs to do geocoding and reverse geocoding in certain situations. The geocoding meant having a coordinate point and needing to revert it into an address. While reverse geocoding means having an address and needing to revert it into coordinates point of longitude and latitude. The famous free geocoding Nominatim API from OpenStreetMap is used for getting coordinates from an address and acts reversely when longitude and latitude are available to give the exact address for these coordinates. Unfortunately, it was inappropriate to use this geocoding service as it gives a limited free offer to convert locations into coordinates points while the dataset got thousands of records that need to be converted, the coasted offer is very expensive. Thus, we are obliged to build a geocoding dataset that has the name of the country and its coordinate points to be matched with the spatial information columns of our dataset. The file is called a country file.

3.5.2 Analyzing Spatial Information

For this stage, each extracted file in the previous process will be processed to find for each record a point of coordinates, longitude, and latitude.

I. Dataset with Location Information

This dataset has spatial information in the location field filled by the user only once when applying to have a Twitter account. This field could be filled with any word; it is up to the user. The location filed may be filled with one word or more. For each condition to reach the exact spatial name of a country a small process will be implemented as shown in figure (3.6).

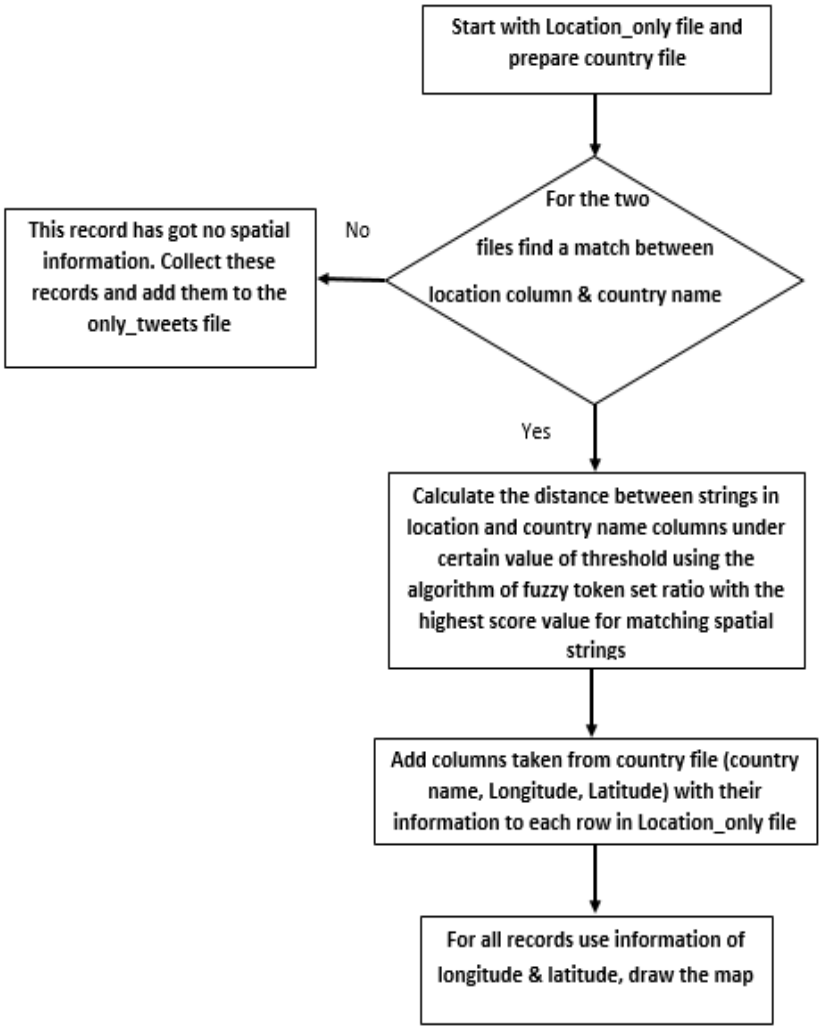

Figure (3.6): The Block Diagram for Processing the Location Dataset

A match is being implemented between the two files: the location field and the country field. When there is no match this record is isolated in another file and then added to the Tweet file. If there is a match new columns are added to the location file containing the name of the country and two coordinates. When adding the coordinates for the matched records, in case the location field contains only one country name without additional spatial details, the coordinates will be added to the file, while when there is more than one spatial word they wouldn't be added. In this case, the fuzzy string matching function (Fuzzy. token_set_ratio ( )) will

be used to make a match. The processed algorithm used brought the coordinates for all records. Algorithm (3.2) shows the steps of this process.

Algorithm 3.2: Converting Spatial Information into Coordinates Point

Input: Location Dataset File (F1), Country File (F2) // both are CSV Files

Output: F1 // Location Dataset File with Coordinates Values

Begin

1. Load Two Files F1, F2 // read the two CSV files
2. Create three empty lists (mat1, mat2, P) // empty lists for storing the matches
2. Convert Location Column in F1 and country column in F2 into two lists (list1, list2) // comparing lists of different lengths
3. Threshold = 90 % // state value of threshold
4. For I in list1 do
5. Find a match between (I, list2) and add the highest scorer to mat1 // collect scorers for all matches in empty list mat1
6. Create new column in F1 contain the values of mat1 named (matches) // F1 (matches) = (country name, score)
7. End For
8. For J in F1(matches) do
9. If J[scorer value] >= threshold do // find the closest match
10. Add j[country name] to P // list P got only country name
11. Add P to mat2 // add only country name to mat2
12. P = [ ] // empty list P
13. End for
14. F1 (matches) = mat2 // insert country name in F1(matches)
15. Join F2 (coordinates) to F1 // merge F2 with F1 depending on country name for both files
16. Save (F1) // save CSV with new column

End

II. Dataset with Place Information

The place field provides many attributes. The best one is the full name it can be used to find the coordinates of the user by using the country file and finding a match with the mentioned place.

III. Dataset with Matched Place and Location

When two spatial information is available it is necessary to find a match between them to know whether this user has changed his home address (moved from one place to another) or still living in the same place (has no mobility). The following procedure has been implemented (see figure 3.7): -

For the matching records, the coordinates were added, while for the non-matched records when the entry of the location field matched with the country field of the country file this file is called the mobility file. Records did not match with the country file, they will depend on the information available in the place column and neglect what is written in the location field. These records should be added to the place_only file. Some problems alert here:

- When the information in the location field contains one spatial word, the coordinates appeared smoothly.
- It is preferable to process all the records of this file and find a coordinate point for users who have information in both fields the place and location. Such records give the exact location of patients.
- The user can enter anything in location field. It may be filled with many spatial words and does not provide the exact location. when matching this entity with the place dictionary, this may lead to no information on coordinating points.

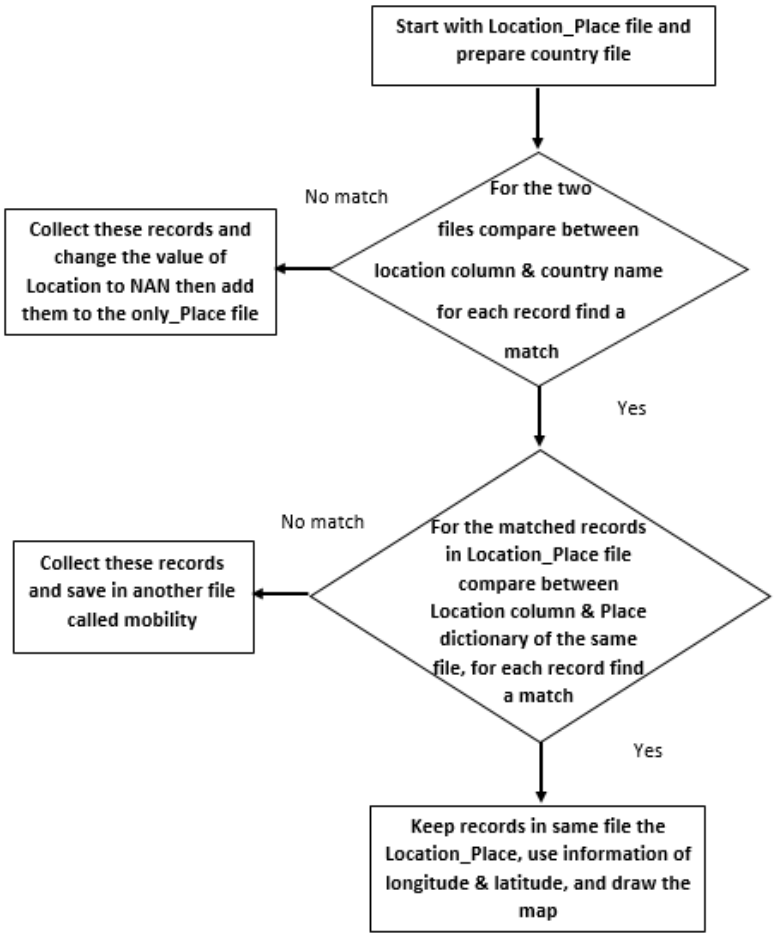

Figure (3.7): The Block Diagram for Processing the Location - Place Dataset

To solve these difficulties, process (A) have been applied here:

Process (A):

- Using a condition to isolate records if the location column matches the full_name column or the country column or the name column then when it is true these records are isolated and give a name of totally-matched.
- The rest will be processed in process (B).
- Check if there is a match between the country column and the full_name column.
- The coordinates were added to the true file smoothly.

- The false file will be given the name of no_match and two columns will be merged, the country and the full_name column and the new column will be the place column.
- The last file will check the location column with the country column of the country file when there is a match it is true otherwise it is false.

For the true records, it is very easy to add the coordinates, but the location column has got more than one word and this kind of matching would not be useful. For this reason, the Edit Distance algorithms will be used to solve this problem. Using the main file and country file, with the location column, for both apply algorithm (3.2).

The matches column gave the exact country and from joining with the country file the coordinates are added to the main file.

Process (B): -

For the false records, a condition has been used that none of the place dictionary columns should match the location column, so the result file is called no match.

- To find the score between the location and full_name columns, the Levenshtein Jaro Winkler has been used to calculate the score. For the highest scores the matched spatial entity should be obtained and bring the coordinate points for this entity.
- To find the distance between the location and full_name columns, the Levenshtein – Distance is used to calculate the distance.
- Isolating records with a score between 0.488 and 0.788.
- Isolating records with a score higher than 0.788.
- Isolating records with a score less than 0.488.

IV. Spatial Information Inferred from Tweets

Tweets are very short; it is supposed to transmit messages very fast. Therefore, it may contain acronyms, videos, images, and emoji's [97]. One of the characteristics of a tweet is that it may be incomplete, incorrect or overstated, for this it should be tested before applying any process. Inferring the geographic location from the content of a tweet constitute an important challenge when analyzing this content [98] [99] [100].

This model needs searching for geographic locations from users' tweets [101]. In this case, the tweets need to be cleaned and the geographic location can be inferred by matching tokenized content of the tweet to a dataset of geographic entities namely a Gazetteer examples [53] [102]. In general, a gazetteer is a geographical directory or dictionary of places and addresses (e.g. cities, towns, streets, urban institutions, residential entities, cultural sites, etc.) whereas each of its entries contains an illustrative features of a given location along with a geographic position, i.e., longitude, latitude. For the case of this proposal a file called country file being used having the names of countries and main cities together with their longitude and latitude coordinates points.

Several techniques may implement the task of identifying the location from the content of a tweet the most popular one is the regular expression to match a string. Its primary function is to offer a search, where it takes a regular expression and a string. Here, it either returns to the first match or else none.

- Prepare the only_tweets file with the country file.
- Clean the text tweet from stop words, digits, numbers, http, url, none ASCII, …

- Apply the regex method to identify the match between the country file and the only_tweet text column.
- Isolating the rows that were able to find the name of the country within its text tweets.
- Adding the coordinate columns to the only _tweets file.

3.5.3 Extracting Followers/Friend's Spatial Information

This step depends on what is available in the spatial fields, for most tuples the location field was filled with information while the place field was very poor in providing any information. It was decided to take only the location field and extract coordinates points of longitude and latitude for all followers by using the steps of section 3.5.2-I.

The output of this stage is the point location of a sick user with many location points of his followers.

3.6 Clusters for Grouping Followers Who Live Closely to the User

This method is to take the coordinates points (longitude and latitude) of the user and his followers and apply a clustering algorithm using K-means clustering which take the longitude and latitude in radians for all the followers of one user (whose coordinating points are the centroid of the cluster) and clustering their entities by giving a label to each one. With this procedure, it will be convenient to identify the spatial entity of one label as the user label to isolate them as the closest individuals to the infected user.

3.7 Classification Stage

Personal tweets discussing the symptoms of Covide-19 or any medical action like taking the test for the illness, need to be recognized to distinguish these tweets from others related to the same topic. It is well known that tweets related to Covide-19 are of different purposes and it is very hard to distinguish the self-reported infected person from other tweets

describing various medical symptoms or tweets posted from different health centers giving instructions to avoid getting infected with this deadly virus, not only that there are also many discussions from users reporting information about other related people. With all of these different tweets looking for self-reported information seemed to be a very difficult task.

3.7.1 Dataset Labeling

Each follower within the same spatial region of the user collects his tweets within one file which contain the metadata and the text tweets. A table contain the following attributes (user ID, post time, follower name, follower ID, full-text tweet, location, tweet place information) to be used for the labeling process.

For each tweet, a label is given according to the situation of the tweet. The last retrieval from Twitter platform could bring any topic as this retrieval is enclosed between two terms within the timeline of the follower. Thus the labels given are three kinds positive, neutral, and negative, each tweet should belong to one of these categories: -

1- Self-Reported Tweets: this category contains an infected individual who suffers from the symptoms of the virus or talking about a positive test result for the disease. This category took the positive sign.
2- Non-Infected Tweets: such tweets took the neutral sign as they have been issued by individuals describing and explaining the outbreak of the disease or symptoms of the virus. Sometimes they are talking about the case of a relative who got the disease.

3- Spam Tweets: individuals talking about any other topic that has nothing to do with the coronavirus. This category took a negative sign.

3.7.2 Cleaning and Pre-Processing Tweets

For a large number of text tweets which is unstructured data, the method of pre-processing is to say altering text into an appropriate form to be classified. In other words, convert text tweets into a structured and consistent format to be analyzed and learned. In natural language processing, it is the first step to be implemented to reach construct a model.

This method decreases the space of features and computational processes, to lead to reasonable classification accuracy. Cleaning tweets is a very crucial step because they are filled with URLs, non-ASCII, special characters, hashtags, symbols, numbers, digits, punctuation, and stopwords that need to be removed. Furthermore, words should be converted into lowercase and tokenized. The following will verify the steps of pre-processing text tweets: -

- The first thing to do is to convert the text to lowercase. This will help to reduce the size of the vocabulary.
- Remove numbers by using either regular expressions or converting numbers into their textual representation. It is possible to convert any number to a word by using the inflect library.
- Removing punctuation from tweets.
- Remove whitespaces from a tweet to avoid unnecessary spaces.
- Remove URLs, hashtags, mentions, and digits from text tweets. They are of no use at all for the classification process.
- Remove stop-wards because they have no contribution to the meaning of the text tweet, and it is safe to remove them as they

make no change in the meaning of the tweet. NLTK got a set of these stop-words that can be used to remove stop-words from the text tweets.

- Stemming tweets to reduce all words to their root word and manipulating with affixes, suffixes, and prefixes by using Porter stemmer.
- Tokenization is the process of splitting tweets into their token words. This process is very important in many processing steps to break up the text tweet into tokens based on commas, semicolons, periods, space, and quotes between words.

3.7.3 Data Splitting

For splitting the dataset used in this dissertation is three quarters for the training set and one quarter for the testing set. This has been done randomly to reduce the variance in data and maintain the generalizability of the models. Shuffling gives a better representation of the training data among overall distribution and prevents model overfitting. The number of the training set is 65857 and the testing set is 21953.

3.7.4 Feature Extraction

Feature extraction refers to extracting and presenting feature representations that are convenient for the task needed to be accomplished and the type of model planned to be built. It is performed to extract meaningful features or attributes from textual tweets to be ready for the classification method. To classify text tweets using machine learning algorithms, they need to work on numerical vectors only as they are unable to use raw text data that got formats that obstruct the work of these algorithms.

1- Bag of Words

This model is a simplified representation used in NLP. The bag of words represents the count of each word in the text tweet disregarding grammar or word order. It keeps only the word and its occurrences. The procedure is for each tweet in the dataset create a zero vector with N dimension, and for the words found in the tweet increase the values in the vector by 1. This vector will be converted into a sparse vector in order to leverages sparsity and actually stores only nonzero entries.

In the whole the tweet is very short and words cannot be repeated more than once, though, for each stored nonzero entry the occurrence should be accumulated for the whole dataset. Sorting the final set and taking only the top 64 most repetitive words to be the input to the system.

2- Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a score focused on each relevant word in the tweet. The term frequency TF is the number of repetitions of a word in a tweet divided by the total number of words in the same tweet, while the inverse document frequency is the logarithm of the number of tweets divided by the number of tweets containing the word. The steps of finding the importance of a word and assigning a weight to it are as follow:

1- Cleaning and preprocessing the tweets using standardization, normalization, and lemmatization.
2- Tokenization of words with their frequency.
3- Using equation (2.4) to find the TF for each word.
4- Using equation (2.5) to find the IDF for the same words.
5- Vectorization of the vocabulary.

3.7.5 Classification using Multi-Layer Perceptron

There are three layers in this model, the input layer is the TF-IDF vectors, while the hidden layers include dropouts and dense layers, their sizes were manually selected by trial and error. MLP patterns represent the basic deep neural network. It is combined of a sequence of completely linked layers. Nowadays, MLP machine learning techniques may utilized to pass over the needs of high processing operations needed by current deep learning structures. [103].

1- Model Architecture

The proposed model using MLP consists of multiple layers of perceptron. Training such a network incorporate obtaining the useful set of loads and biases to find the wright prediction. After cleaning and preprocessing tweets, the data being splitted into training and testing. Transforming each word in the text tweet to a number of its occurrences as such systems cannot deal with characters. The TF-IDF is being utilized for this task, its vectorizer converted tweets into a numeric matrix of TF-IDF features with a dimension of 256 vector for each tweet. The corresponding labels of the text tweets is the input of a label encoder with three target classes.

Both these vectorized inputs are fed to a multi-layer perceptron classifier of three layers. The input layer receives the array of vectors and passes it to the first hidden layer. The parameters of the MLP of Sikit-Learn are set as follows:

- an Activation Function (ReLU),
- an Optimizer (Adam),
- a Learning rate of 0.0001,
- a Batch Size of 32,
- a Regularization Term of 0.0001,

- a Random State of 42,
- Dropout rate of 0.3
- max_iter of 200, and activating the Early_Stopping.

During the training of the model, the weights are updated using Adam to perform better for accuracy and speed. The regularization term was very small according to non-overfitting in this architecture which is also considered as a dropout rate. For a large dataset, the use of batch is preferable as the dataset is divide into batches or clusters flow within the system all together in one time. For the solver Adam, the sum of iterations is the number of how many times each data point will be used. The model stores the weights after training for 200 epochs, this led to the best accuracy on the test set. The output layer uses the Softmax Activation Function.

The model consists of an input layer consisting of 256 nodes because the count number of input samples was 256. As long as there are three classes, the output layer of the model should be set to three nodes for the three classes according to the health condition of the tweep. The hidden layers consist of three layers of (128,128,128) with ReLU activation function. Figure (3.8) shows the flow diagram of the model, while algorithm (3.3) shows the implementation of the MLP classification.

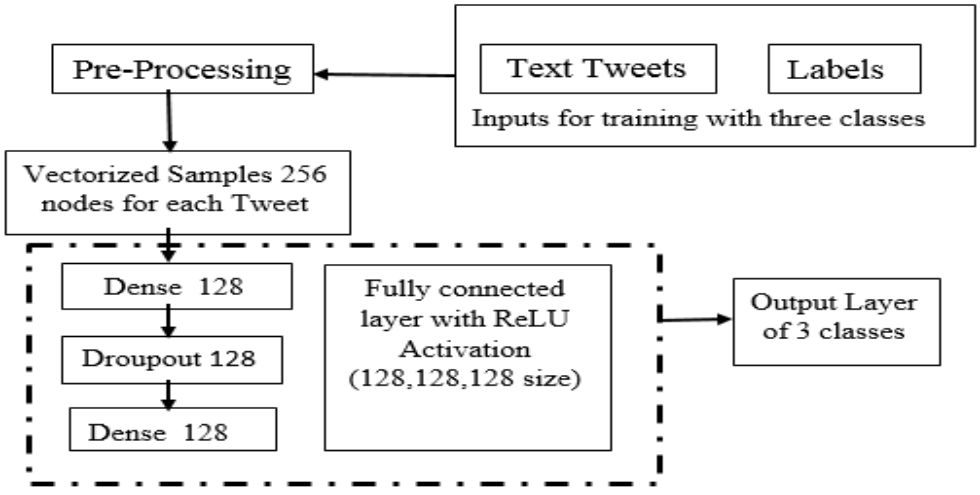

Figure (3.8): Flow Diagram of the Multi-Layer Perceptron Nueral Network

Algorithm 3.3: Tweets Classification using MLP

Input: Text Tweets (labeled manually)

Output: Tweets classified into three classes // Tweets classification

Begin

1. D = get Twitter data
2. L = tweets lowercase(D)
3. R = removal of stop words (L)
4. S = stemming (R)
5. T = tokenization (S)
6. G = generation of TF-IDF model (T)
   // convert labeled tweets into weighted vectors
7. C = create input layer (pre_trained matrix, G)
   // input layer receiving features of same number of neurons
8. While Epoches <= 200 do
9. Tweet transformation via word (C) // pass features into input layer
10. For MLP layers do
11. Dense layer (C) // first hidden layer
12. Droupout layer (C) // second hidden layer
13. Dense layer (C) // third hidden layer
14. End For
15. Output layer (C) // consist of three neurons
16. If early stopping on validation loss <= 0 then
17. End training End
18. End
19. Training output ready

Chapter 4

Implementation and Results

4.1 Introduction

This chapter presents the implementation and experimental results of the dissertation proposal which has been performed using the Twitter dataset. For collecting tweets, Twitter APIs were utilized under certain conditions to retrieve the required data. Then, filtering these tweets to reach the target dataset. Exploring and discovering the spatial information provided within the metadata of the tweet for each account is the main target of this proposal to locate every infected user being provided with point of coordinates. After that, the social network for each infected user was collected and investigate their spatial information from their timeline tweets. Allocating close followers/friends to the infected user is very useful for identifying infected areas with the virus for the sake of controlling disease spread. Finally, classifying the collected tweets of the followers/friends would help in finding three categories of tweets (these are: self-reported, none infected, and spam) by implementing the deep learning neural multi-layer perceptron network.

4.2 System Requirement

- Processor: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50 GHz
- Memory: 6.00 GB RAM.
- Operating System: Windows 10 Pro 64-bit.
- Programming Language: Python 3.8
    - IDE Environment: Jupyter Notebook 6.3.0

4.3 Twitter Dataset

Twitter has been considered a valuable source of different event detection, monitoring, and tracking, mainly for disease outbreaks. It is useless to employ the available Covid_19 datasets to use them in this work due to the presence of confusion. This is because Twitter allows only ID downloads. In most public datasets, they provided the user's ID and not the

Tweet ID itself, which complicates the retrieval process. This model concentrates on the text tweet and its attributes only, (mainly the spatial attributes) and has to avoid replies, mentions, quotes, and retweets as the last one has got no spatial information.

This work needs a tweet of an individual talking about his/her infection with the disease. It is important to distinguish between those individuals who have a disease from those who are worried about getting the virus or talking about a relative who has contracted the disease. Therefore, historical tweets data have been collected from Twitter using the Search API using the logical sentences mentioned in the chapter (3.3.1). Tweets were collected globally using the English language only.

With the use of the Search API the dataset has been collected. The total size for this dataset is 2.5 gigabytes. The number of tweets is 1,010,930 tweets for the period from 1 September 2020 until 31 August 2021. Table (4.1) shows the similarity of the retrieved tweets with the search query logical sentences and also the hashtags and keywords being used.

Table (4.1) Examples of Retrieved Tweets

	Search Query	Tweet
1	"I" AND "have" AND "sore" AND "throat"	@ChrisMdDc Itâ€™s only because I have a horrid sore throat. I only stop drinking when sick (and even then barely)
2	"I" AND "have" AND "headache"	Went into school yesterday with Joey and was there 2 hours max. I now have a headache and my whole body aches.
3	"I" AND "have" AND "diarrhea"	I have chronic diarrhea and the jackass prescribed Miralax at night.

4	"I" AND "have" AND "high" AND "fever",	@animebellies I have a high fever, body heat, headache and vomiting, my body is weak. It's very painful
5	"I" AND "got" AND "covid"	@singularityuta I GOT COVID LMAO a wild ride, still positive but got sent home since doctor said it's good enough
6	”positive_case”	@wyliedpod @TravellingTabby But, a person with "positive case" can surely transmit virus to others who will then become sick.
7	“infection”	@Mark_J_Harper If you believe the media, aren't you putting your dog's at risk of infection and then transmitting to others?
8	“vaccine”	'The vaccines have played a pretty decisive role in breaking the link between infection and death
9	#lockdown	Even as #lockdown restrictions ease across India, it's important to follow safety norms. To enhance social awareness
10	#COVID-19, #coronavirus, #vaccine, #SARSCoV2	Study claiming #COVID-19 #coronavirus #vaccines kill prompted scientists to quit journal board ... | #SARSCoV2 |

This dataset was filled with spam tweets and bots, when searching for duplicate tweets and duplicate users the number was **(**285182). All these accounts were removed. The bar chart in figures (4.1.a) & (4.1.b) depicts the most active users who issued highest number of tweets in this dataset.

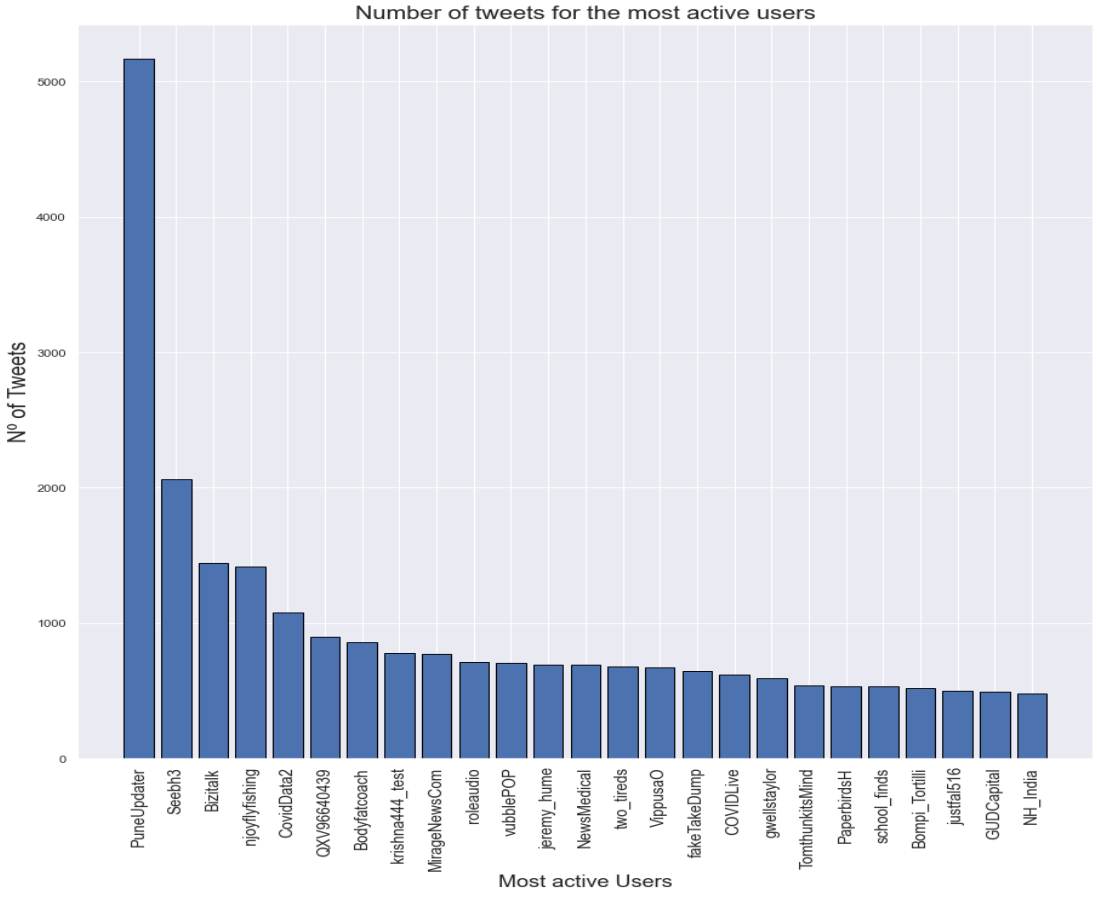

Figure (4.1.a) The Bar Chart for the Most Active User


```
[{"user_name": "PuneUpdater", "NO_twets": 5165},
{"user_name": "Seebh3", "NO_twets": 2062},
{"user_name": "Bizitalk", "NO_twets": 1444},
{"user_name": "njoyflyfishing", "NO_twets": 1418},
{"user_name": "CovidData2", "NO_twets": 1076},
{"user_name": "QXV96640439", "NO_twets": 900},
{"user_name": "Bodyfatcoach", "NO_twets": 860},
{"user_name": "krishna444_test", "NO_twets": 780},
{"user_name": "MirageNewsCom", "NO_twets": 772},
{"user_name": "roleaudio", "NO_twets": 711},
{"user_name": "vubblePOP", "NO_twets": 707},
{"user_name": "jeremy_hume", "NO_twets": 695},
{"user_name": "NewsMedical", "NO_twets": 692},
{"user_name": "two_tireds", "NO_twets": 679},
{"user_name": "VippusaO", "NO_twets": 669},
{"user_name": "fakeTakeDump", "NO_twets": 648},
{"user name": "COVIDLive", "NO twets": 621},
```


Figure (4.1.b) JSON File for the Most Active Users and Number of Tweets

Identifying automated accounts, the newest version V4 of Botometer API is used to find accounts considered as bots in this dataset. The process need a bot key called the Rapid Key1, then applying the Twitter credentials and call the Botometer API to accomplish the process. Three percentages were adopted: 75%, 60%, and 50% to locate accounts behaving as automated accounts, the number of accounts for each percentage are as follow:

- For 75% number of accounts behaved as bots =4929
- For 60% number of accounts behaved as bots=2464
- For 50% number of accounts behaved as bots=75586

The reason behind these percentages is that any account that encounter a behavior of being a bot from 40% and less, it is not a bot, but from 40% and more it is considered as a bot. When this process finished a JSON file format enclosing three parts, the user name with his ID and the percentage score. The total number of bots are 82,979 out of this dataset. Figure (4.2) shows examples for each of the three percentages.

{"user_name": "Healthwatch Soton", "user_id": 46113715, "botometer": "50%"}
{"user_name": "Paperbirds_Coronavirus", "user_id": 1191745846615957506, "botometer": "60%"}
{"user_name": "Cynthia Flynn, MD", "user_id": 1556302518, "botometer": "75%"}

Figure (4.2) Samples of Automated Accounts

4.4 Data Extraction

Tweets were scraped from Twitter and stored in JSON format. After removing spam and bot accounts the JSON file needs to be converted into CSV format to extract the necessary attributes for this work. These attributes are (post time, user name, user ID, text, location, followers count,

friends count, and tweet place information) as shown in table (4.2). The convergence of the file to CSV format encountered some difficulties with the (place dictionary) attribute. For any dictionary, it needs to be converted into key-value pairs of CSV format, but a place dictionary is a nested dictionary as shown in figure (4.3) and it should be extracted first from the whole data frame as a sub-data frame and then with the help of Excel software the needed attributes of this dictionary to be converted into columns like (place_type, name, full_name, country, country_code) then return it to the main data frame.

"place":
{"id": "07d9ef1da0882001",
"url": "https://api.twitter.com/1.1/geo/id/07d9ef1da0882001.json",
"place_type": "poi",
"name": "National Centre for Biological Sciences (NCBS)",
"full_name": "National Centre for Biological Sciences (NCBS)",
"country_code": "IN",
"country": "India",
"contained_within": [],
"bounding_box": {"type": "Polygon", "coordinates": [[[77.58028030114127, 13.07136757041265], [77.58028030114127, 13.07136757041265], [77.58028030114127, 13.07136757041265], [77.58028030114127, 13.07136757041265]]]},

Figure (4.3) JSON Format for Place Attribute Dictionary

Table 4. 2 Sample of CSV Tweets

Location	Text	User-Id	Place-Type	Name	Full-name	Country	Country-Code
Sacramento, CA	"Several top wine estates in the Cape wine lands have shut some or all of their operations after a member of a Dutch wine tour - which visited 30 estates and venues during a 10-day trip - tested positive for Covid-19 at the weekend. " https://t.co/0sFGVGZOs6	3029456157	City	Sacramento	Sacramento, CA	United States	US

4.5 Extracting and Splitting Dataset

As seen in table (4.2), the spatial information is available in the location column and the place dictionary columns. For this, the dataset is going to be split according to the information available as described in chapter three (3.5). The splitting tables are shown in Appendix (1) containing tables [(4.3), (4.4), (4.5), and (4.6)]. While figure (4.4) shows the distribution of the spatial dataset.

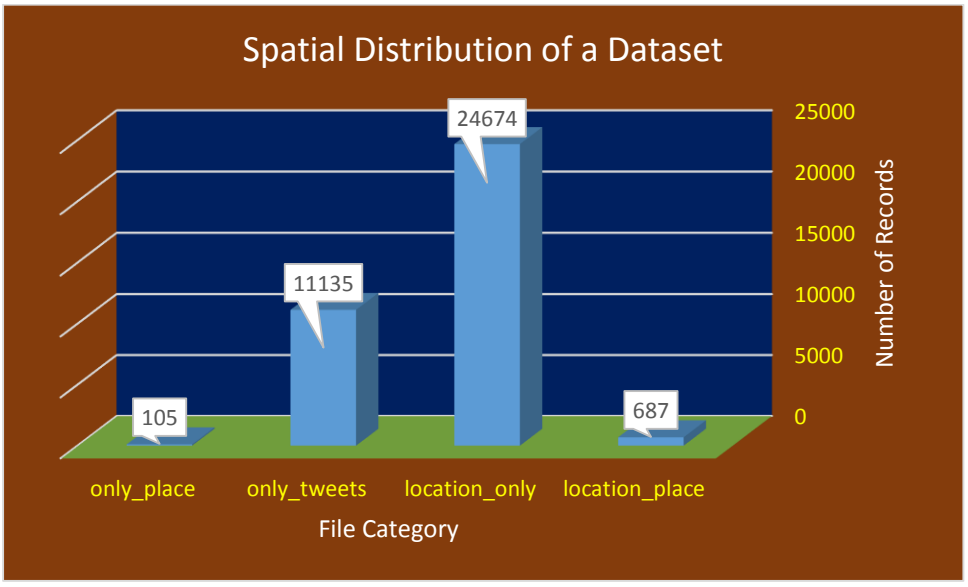

Figure (4.4) The Spatial Distribution of the Dataset

4.5.1 Processing the Location File

As seen from the table (4-3), the spatial information is available in the location column. This entity has to be checked whether it is a real spatial entity or else by comparing it with the country dataset. Then, it is necessary to find the coordinates point of longitude and latitude for this spatial entity by applying algorithm (3.2). All entities were able to get the coordinate points, now it is convenient to draw the distribution map for the disease. The location file (table 4.7 in Appendix (1)) will show the following fields (location, longitude, latitude, text, user ID) only. While for records that have no spatial information in the location field it is going to be extracted from this file and added to the tweet file, they will be identified with red color.

Figure (4.5) Distribution of Disease for Location CSV File

4.5.2 Processing the Place File

This spatial attribute is the most accurate one, the geocoding for the full-name column is done and the coordinates point has been processed. Table (4. 8) in Appendix (1) shows sample of place CSV file.

Figure (4.6) Distribution of Disease for Place CSV file

4.5.3 Processing the Location plus Place file

In this file, there are two spatial entities available. It is very common to choose the place dictionary information to locate the infected user, but this type of spatial information may lead to inferring different statuses concerning the distribution of the virus. In case, the location entity does not match with the place. This means there is a movement from one place to another and the disease may be spread to another place. This situation needs to track the mobility of the infected user to locate the infected area (shown in table (4.9) records 0 and 5). Another status is that the information in the location field does not match with the country file entities. This means that the location field has no spatial information as it is filled with spam and this record could be added to the place file as the only spatial information available is the place dictionary (shown in Appendix (1), table (4.9) records 1 and 4).

For the matched fields, figure (4.7) shows the distribution of infections. When there is no total match another process is implemented

mentioned in process B in chapter three (3.5.2-III). When there are several words in the location field only part of them may match with one word of the place dictionary, the edit distance algorithms were applied to calculate the score and the distance between these fields and were able to identify the matched depending on the value of the score as shown in tables (4.10.a, 4.10.b, 4.10.c). The dataset was split according to the scores to show how far the distance between the two spatial fields was. In the end, the highest score table will be approved then the coordinates point for the full-name column will be figured out. This process is for the nearly exact match between the place residence of the user and the place of issuing the tweet. For the middle score and the far score, part of the location was matched just like the country, or there are more granularity details for the address in the other column or a movement from one country to another or from a city to another in the same country.

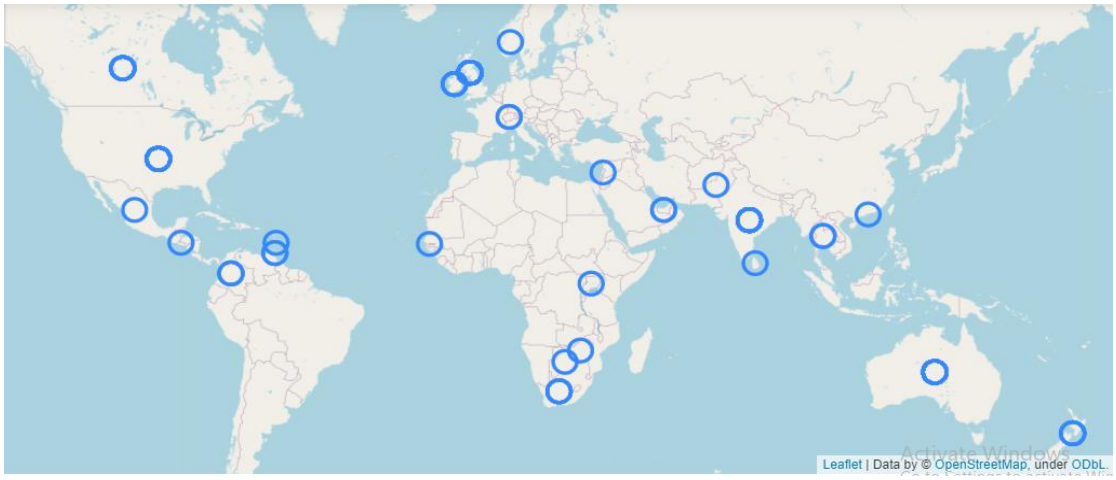

Figure (4.7) Distribution of Infections for the Matched Fields

Table (4. 10-a) Score> 0.788 lead to a Full Match

	Location	Full-name	Score	Distance
10	Arden-Arcade, Sacramento, CA	Arden-Arcade, CA	0.914286	12
18	Perth, Western Australia ????	Perth, Western Australia	0.965517	5
49	Melbourne, Victoria	Traralgon, Victoria	0.800752	9
59	Greater Sudbury	Greater Sudbury / Grand Sudbury, Ontario	0.875000	25
85	Saffron Walden, England	Saffron Walden, East	0.936522	5
101	Washington DC	Washington, DC	0.985714	1
135	Halifax, NS Canada	Halifax, Nova Scotia	0.881270	8
137	Dublin, Ireland	Dublin City, Ireland	0.936667	5
145	Manchester, UK	Manchester, England	0.897744	7
150	Miami, Florida	Miami, FL	0.892063	6
153	Dartford, Kent	Dartford, South East	0.867143	9
166	phoenix	Phoenix, AZ	0.800866	5
169	Washington DC	Washington, DC	0.985714	1
186	Leuven, Belgium	Louvain, Belgium	0.857596	3
194	Staines-upon-Thames, England	Staines-upon-Thames, South East	0.912673	9

Table (4. 10-b) Score Between 0.488 and 0.788 lead to Only One Match Spatial Entity Name

	Location	Full-name	Score	Distance
3	Nairobi, Kenya	Kabete, Kenya	0.704314	6
4	Vypin, Cochin	Cochin, India	0.585470	9
5	Islamabad, Pakistan	Rawalpindi, Pakistan	0.740852	8
7	pune	Pune, India	0.674242	8
15	land of misfit wizards, FL	Miami, FL	0.515954	20

22	Jakarta, Indonesia	Kebayoran Baru, Indonesia	0.744444	11
27	Toronto, and Vancouver Canada	Vancouver, British Columbia	0.581749	25
30	Ashville, OH	Gahanna, OH	0.623737	7
31	Poulsbo, WA	Silverdale, WA	0.603896	10
37	London, England	Paddington, London	0.607407	13
40	Nebraska	Norfolk, NE	0.549242	9
41	Cornwall UK preferably the Moon!	Perran Downs, England	0.509921	27
45	Twickenham	Richmond, London	0.537500	14
50	Mount Vista, WA	Minnehaha, WA	0.632906	8

Table (4. 10-c) Score <0.488 lead to No Matching

	Location	**Full-name**	**Score**	**Distance**
2	Los Angeles, CA	Buffalo, WY	0.376768	12
6	UK-England	Milton Keynes, England	0.443939	14
11	Jakarta	Ciracas, Indonesia	0.420635	15
19	Chandkheda, Ahmedabad, India	New C G Road	0.398413	24
20	Seoul, korea	Virgin Islands, U.S.	0.422222	18
24	Missouri, USA	Bella Villa, MO	0.365812	13
25	Worldwide	Richmond, VA	0.416667	11
29	East Yorkshire	Beverley, England	0.352474	17
32	Poolfoot Farm	Fleetwood, England	0.487179	14
35	debraheggs@rocketmail.com	Wisconsin, USA	0.333651	23
42	Ireland | Scotland | Inuit | Skandinavien	Mascouche, Qu&#xFFFD;bec	0.333094	35
43	Washington	Port Orchard, WA	0.384722	15

In the end, the content of the full-name column will be considered to locate the infected user.

4.5.4 Processing the Tweet File

Many researchers used different methods to infer the location from the text of tweets especially when there is no spatial entity to locate those tweets. In this dissertation the spatial entity needs to be extracted from the text of the tweet, in such a case when the text got any spatial entity, then it is unable to locate this tweet and it should be neglected. To complete the process, tweets need to be cleaned from special characters as shown in figure (4.8.a) the tweets before cleaning and (4.8.b) tweets after cleaning.

After applying the regex method to identify a match between the country file and the text tweet the records that got a spatial entity need to be isolated and apply the coordinates point for them to draw the map. Table (4.11) shows the final result, while figure (4.9) shows the distribution map.

149	2021 JUL 16 Great to see Spanish stay-at-home ...
150	@VOsborne_3 @100DollarBenj @ClayTravis I�m hom...
151	@angrybklynmom NYC Publ.Elem school will start...
152	Recommend board games for 10 days of self isol...
153	@Anglosax0ngurll @Uce101 @insanescot118 @europ...
154	So a plumber came round to give a quote on a j...
155	Incorporate Face Mask As "New Normal": Health ...
156	Self-isolation is over today so I've gone cycl...
157	My features began to speak of the tiredness I ...
158	We have been warned by the authorities for the...
159	@larke_robert @idiots_sick @ChattyCathyAU Also...
160	@kronocide1 Got a couple of bottles in a shipm...
161	Not sure some Transport for Greater Manchester...
162	@POTUS Let�s treat the climate crisis as the n...
163	These people are totally corrupt and criminall...
164	Timeline of #cancelled #gameoff #off #covid #n...
165	" Stay Healthy! We Have Hospitals to Close Dow...
166	Great news!, me and Boboy are now #vaccinated!...
167	169 new cases in Azerbaijan [15:54 GMT] #coron...

Figure (4.8.a) Tweets Before Removing Special Characters

```
149   2021 JUL 16 Great to see Spanish stay at home ...
150    VOsborne 3  100DollarBenj  ClayTravis I�m hom...
151    angrybklynmom NYC Publ Elem school will start...
152   Recommend board games for 10 days of self isol...
153    Anglosax0ngurll  Uce101  insanescot118  europ...
154   So a plumber came round to give a quote on a j...
155   Incorporate Face Mask As  New Normal   Health ...
156   Self isolation is over today so I ve gone cycl...
157   My features began to speak of the tiredness I ...
158   We have been warned by the authorities for the...
159    larke robert  idiots sick  ChattyCathyAU Also...
160    kronocide1 Got a couple of bottles in a shipm...
161   Not sure some Transport for Greater Manchester...
162    POTUS Let�s treat the climate crisis as the n...
163   These people are totally corrupt and criminall...
164   Timeline of  cancelled  gameoff  off  covid  n...
165     Stay Healthy  We Have Hospitals to Close Dow...
166   Great news   me and Boboy are now  vaccinated ...
167   169 new cases in Azerbaijan  15 54 GMT   coron...
```

Figure (4.8.b) Tweets After Removing Special Characters

Table 4. 11 Extracting Location from Text Tweet

	Text	**User-id**	**Where**	**Latitude**	**Longitude**
0	Welcome back to locked down #Victoria as 10 ne...	9.264950e+08	Victoria	-4.616667	55.450000
1	@RepLizCheney Thank you for your smack down of...	1.280000e+18	Jordan	30.585164	36.238414
2	@Jack_Russell_UK I am fine with people choose ...	2.038323e+08	UK	55.378051	-3.435973
3	@NassauExecutive Please start reporting hospit...	7.502592e+07	Nassau	25.083333	-77.350000
4	@antonioguterres @UN Please take action to #My...	1.400000e+18	Myanmar	21.913965	95.956223

...	...	...	...	...	...
2010	The South African men&#65533;s rugby sevens squad has...	6.198551e+08	South Africa	-30.559482	22.937506
2011	Updated SARS-CoV2 VOC fits for Belgium. Actual...	3.300788e+09	Belgium	50.503887	4.469936
2012	Positive case at Barwon Heads primary school a...	1.050000e+18	Victoria	-4.616667	55.450000
2013	Casina di Pio IV, Vatican City (Holy See) -18&#65533;...	1.210000e+18	Vatican City	41.902916	12.453389

Figure (4.9) Distribution Map for the Tweets File

4.6 Fetching Followers and Friends for each User

The output of the last stage was the location of the infected users. The need in this stage is to identify their social network locations and

investigate the closest followers and friends to the infected user. The metadata with the tweet of the infected user has got only the number of his followers and the number of his friends. Some users have got many followers and friends more than thousands. For this work, the decision was to search for fewer followers and friends to overcome many difficulties concerning the retrieval process. The following procedure will show how to reach the followers' and friends' locations.

4.6.1 Identifying the Followers/Friends IDs

It is very important to find the IDs of the social network of infected users. From the data frame of the collected data from Twitter, each record has got the user-id value from this value it is possible to get the screen name of the user by applying the following code as shown in figure (4.10).

```
User = api.get_user (1405677175060193280)
Print(user.screen_name)
Output: Shuntaraaa
```

Figure (4.10) Block of Code to Show How to Retrieve the User screen_name

Getting the user screen name would make the process of retrieving the ids of his followers very easy. The follower API (followers = api.followers_ids(screen_name=user) can get the ids of all the followers for the user screen_name. This step can get up to 5000 ids and no more. In case the number of followers is more than 5000 then another process needs to be done to overcome this limitation with the follower API. (the same can be done for the friends' ids).

To overcome the last limitation, it is necessary to set the parameter wait_on_rate_limit to True with Tweepy API. This is because when downloading a large amount of data, the rate limit should be avoided to

continue retrieving data. Table (4.12) shows a sample of retrieved ids for the followers of the first infected user.

Table (4. 12) Sample of Retrieved Followers IDs

Output : 1000	Previous _cursor	Next_cursor(user ID)	IDs of followers
	0	1710813885960930399	1437949410021978117
			1437948541901713409
			1437947905420058628
			1336494026925740032
			1437948713834483713
			782688591331766272
			1381602541243310085
			1437948842566160385
			1437949440740904961

4.6.2 Get Tweets between Two Dates for Followers

As mentioned in chapter three (3.4.2) the steps of retrieving tweets between two dates the start date and end date, this service is unavailable within the REST APIs. For this, the procedure of the up mentioned part was implemented using the timeline of the follower to check the retrieved tweets and to investigate their time of issue using the created_date attribute to be compared within the created_date of the original user returning backward seven days before. This method was able to retrieve several tweets for each follower. Unfortunately, this method encountered a drawback, that the information of geo-coordinates and place dictionary was null. This happened because the formal were dictionaries filled with other data. Neglecting this shortcoming, the location field for most of the tweets was filled by the followers and was processed using the method mentioned in (3.5.2-I) to identify the exact address of the followers.

After collecting the tweets of the followers any retweet was removed from the new dataset that contain the follower-id, text tweet, location, and user-id. As can be seen from the table (4.13) for each follower only one tweet is shown and the topic of these tweets is different and they need to be classified to identify the self-reporter of the coronavirus.

Table 4. 13 Sample of Retrieved Followers Data

	User-ID	Follower_ID	Text-tweet	location
0	171081388596093039 9	1437949410021978117	Energy Mining Comparison Of Different Blockchains https://t.co/GI6eXIllE8 via @YouTube \n\n#Libonomy #crypto #LockDown",	USA
		1437948541901713409	@GOPLeader Trump says everyone knew Covid-19 was airborne in February: \u2018This was no big thing\u2019\n\nPresident Donald Tr\u2026 https://t.co/FC8djzw1wt	Pennsylvania, USA
		1437947905420058628	Hi all... I may not chat as much this week, because I find this movie so absorbing and so relaxing... And honestly given that we're surviving a global pandemic right now, I grab every chance to relax I can take. #BMovieManiacs	Bts World
		1336494026925740032	@SharylAttkisson @trish_regan Did you fact check all the false information from Trump White House, right wing media🤔 https://t.co/I7qPk5bgZn	England

		1437948713834483713	if you think wearing a mask and adhering to social distancing and other restrictions is pointless, maybe read this https://t.co/gAntuTrZQQ	Northwest Indiana
		782688591331766272	This has been an AMAZING and busy week. Thank you to Both Sides of the Conversation for inviting us back to spread https://t.co/JntZnJNkYG	Brussels
		1381602541243310085	The director of the Centers for Disease Control and Prevention (CDC) said the coronavirus outbreak https://t.co/Bh6ej2ElIj	Manhattan, NY
		1437948842566160385	@EW_2021 I chased an iowa tornado outbreak! https://t.co/0uzoCQgEpa	San Francisco
		1437949440740904961	@MichelleMarieTV My friend has a breakthrough case and they didn't even test him. His wife was positive (unvaxed) and he had symptoms (JNJ)	Mumbai, Indai

4.6.3 Identifying the Closest Followers to the User

To determine the infected area that contains the infected user and his social network, the K-means clustering algorithm was applied to the field of location with coordinates points for both the user and his followers. Classes were given for each entity using numbers. Matching between the cluster number of the user and his followers to give those who have the same cluster number. The final result is to isolate those followers and find their coordinate points to draw the map. Table (4.14.a, b) shows a sample

of the dataset for followers of one user who lives in New York City in the USA of Longitude: -73.935242 and Latitude: 40.730610. The clusters show different countries for all followers as in table (4.14.a), after matching the cluster label with the user, isolate followers who have the same cluster label as in table (4.14.b). The location was converted into the address title using the Nominatim OpenStreetMap. Figure (4.11) shows clusters distribution using four groups only. Figure (4.12.a.b) shows the distribution map of those followers who live very close to the infected user.

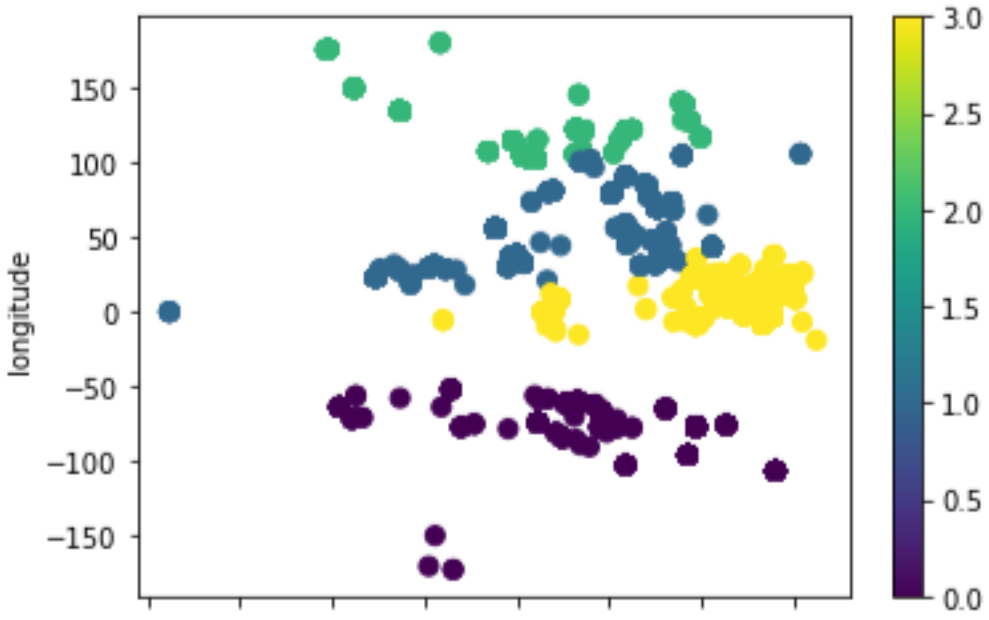

Figure (4.11): Spatial Distribution Clusters

Table (4. 14-a) Sample of Clustering the Followers of One User

	Location	Text	Follower-ID	Lat	Long	Lat/radians	Long/radians	Class
0	friendzone Maine, USA	@eullrich11 @KimmMathiesen @ThomasKlineMD @JSG...	4.129480e+08	37.0902	-95.7129	0.647346	-1.670505	0
1	Rochester, NY	COVID-19 Restrictions Lifted - Greater #Darwin...	3.149729e+09	43.161030	-77.610924	0.441121	-2.334816	0
2	Fort Collins, CO	@alew_plf Trees and I don't mean the weeds! I'...	3.207330e+08	4.5709	-74.2973	0.079777	-1.296733	5
3	Singapore	asked the doc if I could be released tmr since...	1.040000e+18	1.3521	103.8198	0.023599	1.811997	12

Table (4. 14-b) Isolating Addresses of all Followers Who Live in N.Y

Address & Location	Latitude	Longitude
2040 Forest Avenue Staten Island NY 10303 (40.62609762251°, -74.157474896698°)	40.626017	-74.156541
460 Brielle Avenue Staten Island NY 10314 (40.597953866387°, -74.131766325901°)	40.593798	-74.135437
111 Canal Street Staten Island NY 10304 (40.626576449488°, -74.076994470087°)	40.626584	-74.076758
60 Madison Street New York NY 10038 (40.711969729879°, -73.997542215578°)	40.712019	-73.997309
34 Spring Street New York NY 10012 (40.721735692317°, -73.995719412648°)	40.721721	-73.995732

2201 Neptune Avenue Brooklyn NY 11224 (40.578488319967°, -73.989497148121°)	40.578468	-73.989614
227 Madison Street New York NY 10002 (40.712841397448°, -73.987598534724°)	40.712784	-73.988417
227 Madison Street New York NY 10002 (40.712841397448°, -73.987598534724°)	40.712784	-73.988417
227 Madison Street New York NY 10002 (40.712841397448°, -73.987598534724°)	40.712784	-73.988417
295 Flatbush Avenue Extension Brooklyn NY 11201 (40.692053402323°, -73.982413019989°)	40.691986	-73.982496
280 Delancey Street New York NY 10002 (40.716367017869°, -73.980135660503°)	40.716079	-73.980373
462 First Avenue New York NY 10016 (40.73962320748°, -73.976572846645°)	40.739173	-73.976862
462 First Avenue New York NY 10016 (40.73962320748°, -73.976572846645°)	40.739173	-73.976862
2601 Ocean Parkway Brooklyn NY 11235 (40.586645433957°, -73.965830115778°)	40.586552	-73.966168

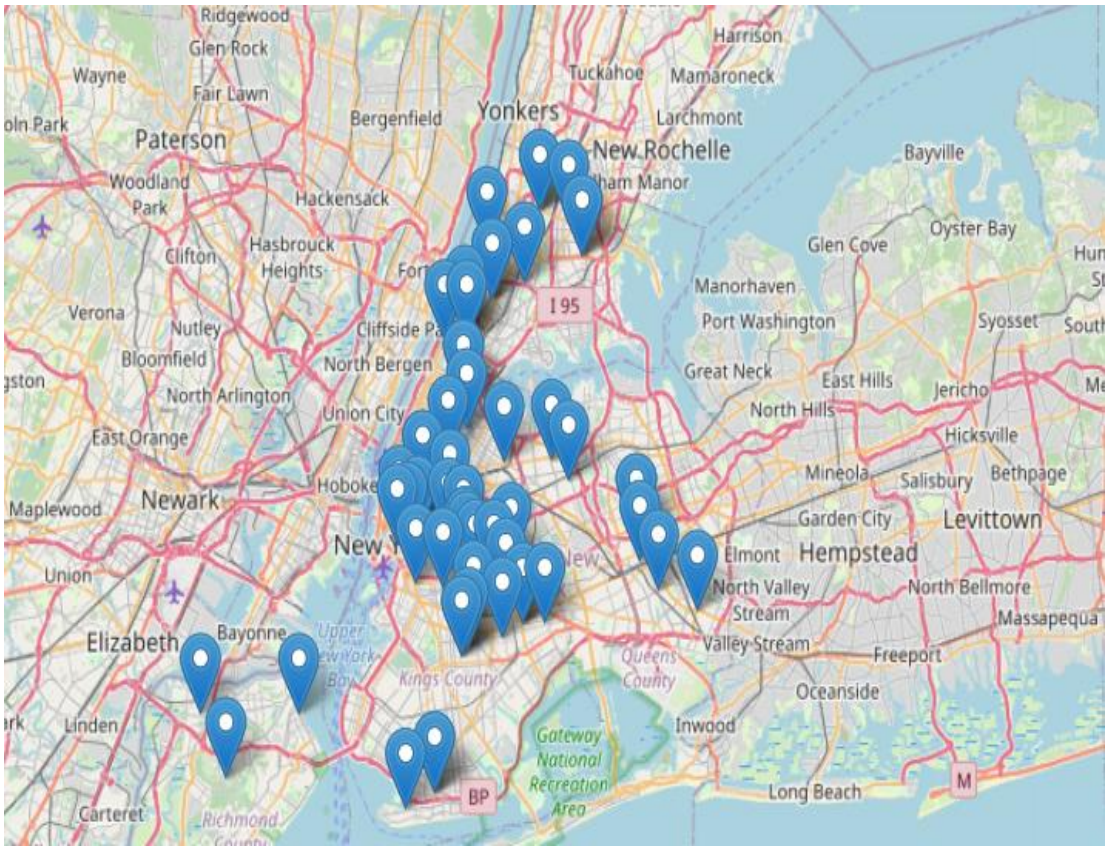

Figure (4.12.a) Distribution of Followers Who Live in New York City

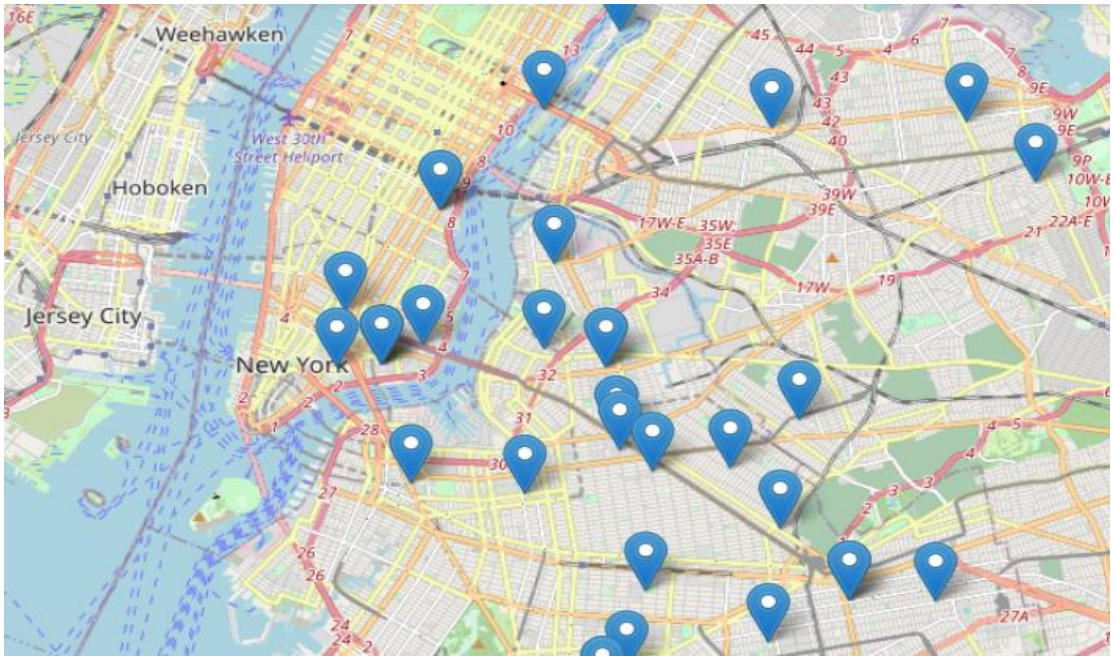

Figure (4.12.b) A Closer Map View Showing Locations of Followers in N.Y.C.

4.6.4 Labeling and Collecting Followers Dataset

For all the followers who have been isolated and located as they have the same cluster of the infected user, all their tweets were collected in one dataset and labeled manually. Three kinds of labeling positive, neutral, and negative. Table (4.15) shows tweets for different followers, their text tweets being labeled according to the situation of the tweet. While figure (4.13) shows a bar chart of the percentage of the three labeled data.

Table (4.15) the Followers Labeled Dataset

	Created-Date	Follower-ID	Text	Orientation	Label
0	2020-09-07 06:56:47+00:00	8.180000e+17	Second covid jab..Astra Zen...done last Thursd...	positive	1
1	2020-09-07 08:23:13+00:00	1.461401e+08	Virtual Training https://t.co/GFkstMoeTe\nUS |...	neutral	0
2	2020-09-07 09:19:41+00:00	1.253571e+08	Metso Outotec hopes to expand its mobile &...	neutral	0
3	2020-09-07 11:04:31+00:00	3.359246e+08	NOTICE: A positive case of #Covid-19 was ident...	positive	1
4	2020-09-07 12:43:10+00:00	1.606591e+08	Vinton County in rural Southeast Ohio was the ...	negative	-1

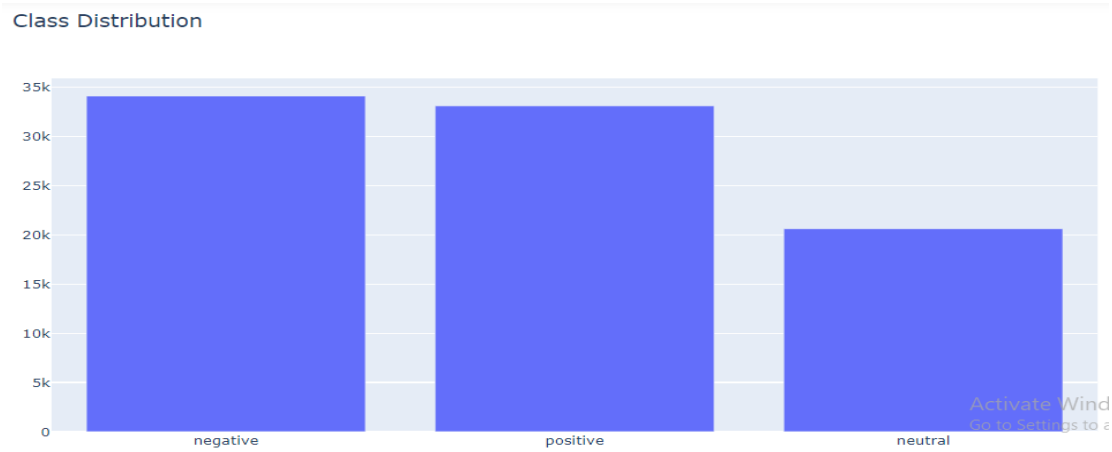

Figure (4.13) Bar Chart Showing the Amount of the Three Labels

The word cloud for the three labels (positive, negative, neutral) is shown in figure (4.14.a, b, c) respectively.

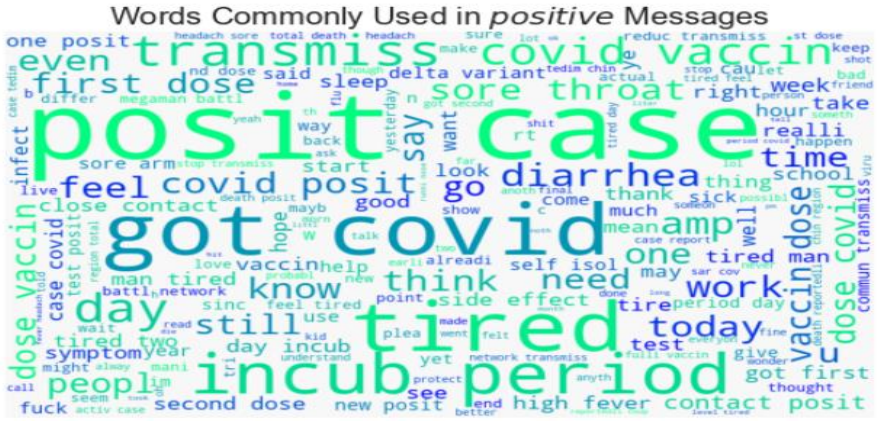

Figure (4.14.a) Word Cloud for Positive Tweets

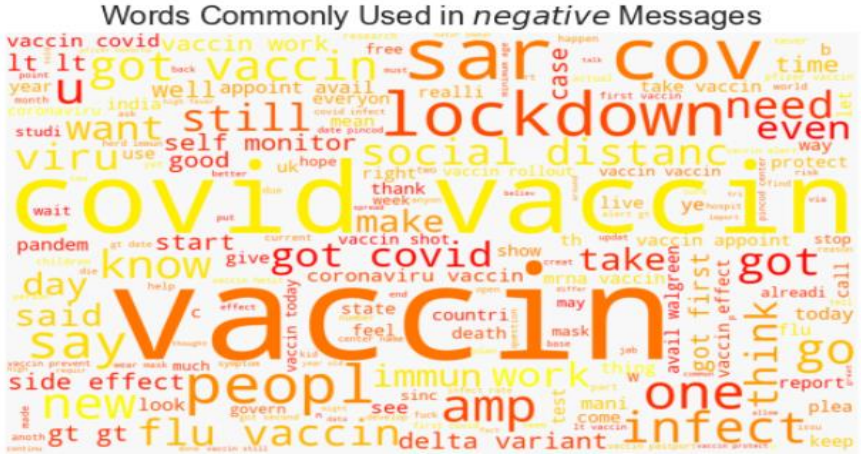

Figure (4.14.b) Word Cloud for Negative Tweets

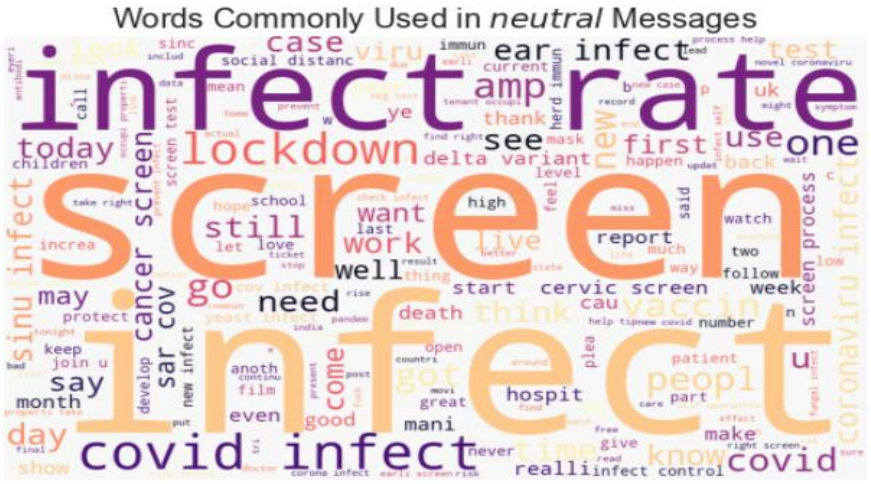

Figure (4.14.c) Word Cloud for Neutral Tweets

4.7 Classification Stage

This stage would identify spatially close relatives or friends of the infected user that they might catch the disease from the user or vice versa. This stage is looking for a self-reported follower who was talking about his infection. It is a very complicated matter to identify such tweets. The MLP classification algorithm was used to classify the three classes being manually labeled. The dataset collected is about 87810 records.

4.7.1 Preprocessing Tweets

Tweets are unstructured data, it contains different information like dates, numbers, letters, characters, and a small part of the text. This made tweets no analytics unless converting it into structured data because there is no code to parse this complex string. The main part is the text portion which needs to be analyzed using natural language processing and some text analysis algorithms. Stemming is one kind of these algorithms, its main duty is finding the root of words to enable the process of classification afterward.

Preprocessing tweets is a basic way to change the text to rise its facility to be processed by NLP algorithms. Several steps are implemented for this purpose like converting all capital letters with a small letter to unify

the words. This may help the process of searching and matching. Some tweets may contain non-English words that make no sense when processing them, for this they need to be removed. Hashtag signs (#) also should be removed, as the words or words that came after the sign may help in analyzing the tweet. The user name or mention in a tweet also needs to be removed as it gives no valuable information to the text tweet, it has got no meaning. It signs is @ and the name that came after also needs to be removed. Another part that is useless to keep is the URL in the tweet. The part that starts with (*http*) should be removed. Tweets are filled with punctuation signs, numbers, digits, spaces, and special characters all should be removed from the tweet. Another part should be removed the stop words like (‘a’, ‘me’, ‘the’, ‘what’,..) there is a very long list that contains these stop words. After removing stop words tokenization should be implemented that separates each word in the tweet. Then stemming process takes place. It converts words to their origin root (detecting, detected both return the word detect). Table (4.16) shows a sample of preprocessing operations on the dataset.

Table (4.16) Sample of Preprocessing Operation

	Processing Operation	**Result Action**
0	Text	@Kit_Yates_Maths "A ""Vaccine"" is a compound which ""prevents infection"" none of the COVID snake oils meet the *patentable standard * https://t.co/zOYvtbKmSq"
1	Convert to Lowercase	@kit_yates_maths "a ""vaccine"" is a com pound which “”prevents infection”” none of the covid snake oils meet the *patentable stan dard * https://t.co/zoyvtbkmsq"

2	Remove non-english words	@kit_yates_maths "a ""vaccine"" is a com pound which “”prevents infection”” none of the covid snake oils meet the *patentable stan dard * https://t.co/zoyvtbkmsq
3	Remove Hashtags	@kit_yates_maths "a ""vaccine"" is a com pound which “”prevents infection”” none of the covid snake oils meet the *patentable stan dard * https://t.co/zoyvtbkmsq
4	Remove URL	@kit_yates_maths "a ""vaccine"" is a com pound which “”prevents infection”” none of the covid snake oils meet the *patentable stan dard *
5	Remove Mentions	"a ""vaccine"" is a compound which “”pre vents infection”” none of the covid snake oils meet the *patentable standard *
6	Remove punctuation, special character, and numbers	a vaccine is a compound which prevent s infection none of the covid snake oils meet the patentable standard
7	Remove Stop Words	vaccine compound prevents infection covid snake oils meet patentable standard
8	Tokenization	['vaccine', 'compound', 'prevents', 'infect ion', 'covid', 'snake', 'oils', 'meet', 'patentable' , 'standard']
9	Stemming	['vaccine', 'compound', 'prevent', 'infect', 'covid', 'snake', 'oil', 'meet', 'patent', 'stand']

4.7.2 Feature Selection

Feature representation is needed for the task of classification. These features are attributes taken from a textual tweet in the form of a numerical

vector to accomplish and assist in building the classification model being chosen which is the MLP. Two kinds of features the BoW and the TFIDF are employed for this step.

1- TF-IDF Term Frequency Technique

This technique gives the meaning of tweets in weight scores that conformed to several words, unlike the BoW technique that gives words in numbers. The following sample of processing three tweets shows in the table (4.17) and table (4.18).

Table (4.17) Sample of Vectorized Preprocessed Tweets

Tweet	**Preprocessed Tweet**	**Apply Vectorizer**
Tweet1	'second covid jab astra zen done last thursday high fever one day headach back normal'	['second', 'covid', 'jab', 'astra', 'zen', 'done', 'last', 'Thursday', 'high', 'fever', 'one', 'day', headach', 'back', 'normal']
Tweet2	'metso outotec hope expand mobil amp stationary crush screen equip reach central southern'	['metso', 'outotec', 'hope', 'expand', 'mobil', 'amp', 'stationary', 'crush', 'screen', 'equip', 'reach', 'central', southern']
Tweet3	'vinton counti rural southeast ohio last counti state record posit case covid vinto'	['vinton', 'counti', 'rural', 'southeast', 'ohio', 'last', 'counti', 'state', 'record', 'posit', 'case', 'covid',vinto']

Table (4.18) Matrix of Weights for the Up-Mentioned Sample

```
[[0.         0.26577704 0.26577704 0.         0.         0.
  0.20213029 0.         0.26577704 0.26577704 0.         0.
  0.26577704 0.26577704 0.26577704 0.         0.26577704 0.20213029
  0.         0.         0.26577704 0.         0.26577704 0.
  0.         0.         0.         0.         0.         0.26577704
  0.         0.         0.         0.         0.26577704 0.
  0.         0.26577704]

 [0.2773501  0.         0.         0.         0.2773501  0.
  0.         0.2773501  0.         0.         0.2773501  0.2773501
  0.         0.         0.         0.2773501  0.         0.
  0.2773501  0.2773501  0.         0.         0.         0.2773501
  0.         0.2773501  0.         0.         0.2773501  0.
  0.         0.2773501  0.         0.2773501  0.         0.
  0.         0.        ]

 [0.         0.         0.         0.26577704 0.         0.53155409
  0.20213029 0.         0.         0.         0.         0.
  0.         0.         0.         0.         0.         0.20213029
  0.         0.         0.         0.26577704 0.         0.
  0.26577704 0.         0.26577704 0.26577704 0.         0.
  0.26577704 0.         0.26577704 0.         0.         0.26577704
  0.26577704 0.        ]]
```

2- Classification Results

Dataset was split randomly into a training set with a ratio of 75% of the total dataset and a testing set with 25%. Then applying the classification model the MLP which is a deep learning algorithm the result of the whole process for the three classes is shown in figure (4.15) the confusion matrix of the actual and predicted values.

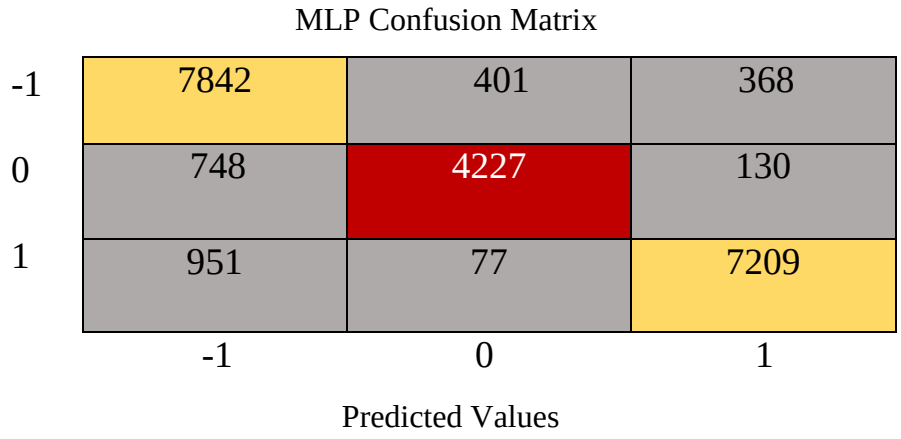

Figure (4.15) Confusion Matrix of Three Classes

The performance metrics report is shown in table (4.19).

Table (4.19) The Performance Metrics Report for the MLP Classification Model

Text Mining Algorithm	***Label***	***Accuracy Metrics***			
		Accuracy	***F1-Measure***	***precession***	***Recall***
ML-Perceptron	**-1**	**0.878**	**0.86**	**0.82**	**0.91**
	0		**0.86**	**0.90**	**0.83**
	1		**0.90**	**0.94**	**0.88**

3- Performance Metric for MLP Classification Algorithm

When evaluating the classification of predictive tweets, it is very important to identify the type of classified tweets whether they are true or false according to the model being used. As for most machine learning algorithms accuracy, precision, and recall are the most common parameters to evaluate the performance of the MLP algorithm. From the confusion matrix, the four categories true positive, false negative, false positive, and true negative are as follows:

1- TN: the model correctly predicted the author of the tweet was not infected. And the real or actual value was also not infected.

2- FN: the model incorrectly predicted the author of the tweet was not infected. And the actual value was infected.
3- FP: the model incorrectly predicted the author of the tweet was infected. And the actual was none infected.
4- TP: the model correctly predicted the author of the tweet was infected. And the actual value was also infected.

The accuracy of the model used Equation (2.13) in chapter 2 to give the correct predictions.

4.8 Identifying Infected Area

Isolating the predictive and actual positive tweets and retrieving their records to find the followers' location. Collecting locations of one identified place like New York City and drawing the map for all of them. According to the number of infected followers, action should be taken to announce that this city got a disease outbreak.

4.9 Challenges and Problems of Model Implementation

Due to the presence of many obstacles to evaluating this project, it is quite obvious that this evaluation would not be accurate for many reasons. Let us start from the beginning of the first step of collecting the data. Tweets came from all over the world most of them represent an infected case the number of these cases may exceed the real number registered with the approved health departments in any country. This is a truth fact, many people were infected but did not report their infection for many reasons related to their countries' policies, or because they were afraid of being incubated and socially isolated, and they preferred to stay at home and took the usual medicine. This happened because people use social platforms like Twitter to talk about their experiences with the pandemic. They were receiving much advice concerning how to cure from this disease. This is an

advantage to this project because this can predict so many infections in different places.

The second step, identifying the exact location, this step is crucial. Some users have their GPS on while others are off, this affects locating the exact place of the infected user. Also, some users fill the location field honestly, while others do not. According to these obstacles, the stage of splitting and extracting the exact location was implemented and analyzed in some situations to ensure getting the most accurate location among the available spatial information accompanied by the text tweet. The problem of geo-decoding and geo-coding using the famous OSM API took place and prevented the procedure of this project from getting any progress in having the exact address that helps in calculating the distance very easily.

Many problems have been faced in the operation of collecting the social network of the user through reaching his followers and friends, a main problem was issuing the new second version of the Twitter lab that retired most of the REST APIs at the time the project started. Still, most of the retrieved tweets were videos, images, and hashtags that should be withdrawn. In this aspect, the loss of so many followers' accounts affected the retrieval process and the size of the dataset. Two things that need to be available for this retrieval to make use of this operation are: 1) the text tweet to be classified, 2) the spatial information is taken from the location field and place dictionary to find the address of this follower.

In the meanwhile, identifying the closest followers to the user can be solved in many ways. But clustering the coordinate points of all followers was the best way to identify groups of the same cluster label. This made calculating the distance between the user and the followers using kilometers to measure this distance, giving the indication that these cluster groups may have the same distance from the user or slightly near that measure.

It was decided to identify the closer followers before classifying their tweets to find out who caught the pandemic. This may result in isolating the infected group and locating the area as an infected area.

Chapter 5

Conclusions and Future Works

5.1 Conclusion

Since the global lockdown as a result of the spread of the COVID-19 pandemic platforms all over the world became an outlet for exchanging news users, transmitting their health situation or their relatives, and receiving advice from either health agencies or from cured or infected friends who have contracted this disease and recovered from it. Twitter is one of these famous platforms due to the spatial information it provides. Several spatial attributes within the metadata of the retrieved tweets are available that assess finding the exact location address of the infected user. Starting with the first infected user the rest of his social network can be investigated whether they caught the disease or not, also investigating how far are they from the infected target. The need is to identify the geospatial area that has got so many infected cases.

Several conclusions have been found by implementing the pipeline of this model.

1- The lack of an available public benchmark dataset that provides the spatial information together with the text tweets for the user and his network. For this, the authors were obliged to download and construct a dataset from Twitter to accomplish this work.
2- In every dataset, the available information on the geospatial features is rarely encountered as they are not very well represented, not only that, the structure of the place information as a nested dictionary is very difficult to be converted into a table using programming. The dataset being processed here is meant to gain a geospatial feature that is represented as a point of latitude and longitude for each tweet starting from the most accurate location till reaching the inferred location. When two

points of coordinates are available, then distance can be calculated.

3- In order to identify the exact location, the fuzzy and Levenshtein matching algorithms being used for each file that have been extracted according to the available spatial information being used. Some of them need more investigation, while others were quite adequate to give the exact location without any inferring process.
4- The MLP algorithm works as a deep neural network. This algorithm succeeded in training this model and testing the dataset. And due to the high number of infected people in New York city the classification process where able to predict the infections.

5.2 Future Works

Based on the project findings and description of the proposed model, the following are suggested for future work:

1- For the classification part, the idea of labeling the tweets manually was a tedious job, the suggestion is to use the semi-supervised approach instead of the supervised one. This could be done by preparing two lists positive and negative lists filled with words of the same kind of its title.
2- To avoid the difficulty of obtaining the exact address of the infected user being provided with two coordinate points the longitude and latitude through obtaining the Nominatim (the search engine for OpenStreetMap (OSM) which did not work with us, several world cities databases could be used but it is preferable to obtain a database of a certain country.

3- The proposed pipeline could be improved to analyze any dataset being retrieved from Twitter concerning any other disease and not only COVID-19.

4- Testing the proposed model with other simulation distributions like the SEIR epidemic model, or SIR.

References

[1] M. Zohar, "Geolocating tweets via spatial inspection of information inferred from tweet meta-fields," *International Journal of Applied Earth Observation and Geoinformation,* no. 102593, p. 105, 2021.

[2] J. Burgess and N. Baym, "Twitter: A Biography," *New York University Press,* 2020.

[3] C. E. Slavik, C. Buttle, S. L. Sturrock, J. C. Darlington and N. Yiannakoulias, "Examining tweet content and engagement of Canadian public health agencies and decision makers during COVID-19: mixed methods analysis," *J Med Internet R,* vol. 23(3), no. e24883, 2021.

[4] H. Jang, E. Rempel, D. Roth, G. Carenini and N. Janjua, "Tracking COVID-19 discourse on twitter in North America: Infodemiology study using topic modeling and aspect-based sentiment analysis," *J Med Internet Res,* vol. 23(3), no. e25431, 2021.

[5] A. Java, X. Song, T. Finin and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *Workshop on Web Mining and Social Network Analysis*, 2007.

[6] M. Kaigo, "Social media usage during disasters and social capital: Twitter and the Great East Japan earthquake," *Keio Commun. Rev,* no. 34, pp. 19-35, 2012.

[7] F. Psallidas, H. Becker, M. Naaman and L. Gravano, "Effective event identification in social media.," *IEEE Data Eng. Bull.,* no. 36, pp. 42-50, 2013.

[8] G. Bakal and R. Kavuluru, "On Quantifying Diffusion of Health Information on Twitter.," *IEEE EMBS Int Conf Biomed Health Inform,* pp. 485-488, 2017 Feb;.

[9] H. Liang, "Broadcast versus viral spreading: the structure of diffusion cascades and selective sharing on social media.," *Journal of Communication ,* vol. 3, no. 68, pp. 525-546, 2018.

[10] O. NIAKSU, "CRISP DATA Mining Methodology Extension For Medical Domain", Vols. Vol.3, No. 2, *Baltic J. Modern Computing*, 2015, pp. 92-109.

[11] K. Chulis, "CRISP Data Mining Twitter for Cancer, Diabetes, and Asthma insights", In PhD Dissertation, *Purdue University West Lafayette, Indiana*, 2016.

[12] M. Shah, "Disease Propagation in Social Networks: ANovel Study of Infection Genesis and Spread on Twitter," in *JMLR, Workshop and Conference proceedings 53:1-17*, 2016.

[13] J. Teodoro and A. Macaraeg, "Disease Monitoring System Using CRISP Data Mining," *Computer Studies Department University of Caloocan City, Philippines*, 2017.

[14] R. Matsumoto, M. Yoshida, K. Matsumoto, H. Matsuda and K. Kita, "Visualization of the occurrence trend of infectious diseases using Twitter," *Institute of Technology and Science, University of Tokushima,* pp. 511-514, 2018.

[15] H. Liang, I. Fung, T. Chun-Hai, T. H. Zion and J. Yin, "How did Ebola information spread on twitter: broadcasting or viral spreading?," *BMC Public Health,* vol. 19, no. 438, 2019.

[16] A. Alessa and M. Faezipour, "Preliminary Flu Outbreak Prediction Using Twitter Posts Classification and Linear Regression with Historical Centers fro Disease Control and Prevention Reports: Prediction Framework Study," *JMIR Public Health Surveill*, vol. 5, no. 2, 2019.

[17] A. Alshehri, "A Machine Learning Approach to Predicting Community Engagement on Social Media During Disasters", Florida: PhD Dissertation, *University of South Florida*, 2019.

[18] Yousefinaghani S., Dara R., Pojak Z., Bernardo TM. and Sharif S., "The assessment of Twitter's potential for outbreak detection: avain influenza case study," vol. 9(1), pp. 1-17, 2019 Dec 3.

[19] M. Paul and D. Mark, "You are what you tweet: Analyzing twitter for public health.," in *In Proceedings of the International AAAI Conference on Web and Social Media*, 2011.

[20] Hernandez-Suarez, Aldo, Gabriel Sanchez-Perez, Karina Toscano-Medina, Victor Martinez-Hernandez, Victor Sanchez, and Hector Perez-Meana. "A web scraping methodology for bypassing twitter API restrictions." *arXiv preprint arXiv*:1803.09875 (2018).

[21] I. Dongo, C. Yudith, A. Ana, M. Fabiola, Q. Yuni and B. Sergio, "Web scraping versus twitter API: a comparison for a credibility analysis," *In Proceedings of the 22nd International conference on information integration and web-based applications & services,* pp. pp. 263-273, 2020.

[22] Z. Saeed, R. Abbasi, O. Maqbool, A. Sadaf, I. Razzak, A. Daud, N. Aljohani and G. Xu, "What's happening around the world? a survey and framework on event detection techniques on twitter," *Journal of Grid Computing,* vol. 17(2), pp. 279-312, 2019.

[23] A. Pappurajan, "Design and analysis of web mining algorithms", *Manonmaniam Sundaranar University*, 2014.

[24] D. Freelon, "Computational research in the post-API age," *Political Communication,* Vols. 35, no. 4, pp. 665-668, 2018.

[25] S. Ruiz and Javier, "Twitter research for social scientists: A brief introduction to the benefits, limitations and tools for analysing Twitter data," 2017.

[26] M. Martinez, "Digital sources: a case study of the analysis of the recovery of historical memory in Spain on the social network Twitter," vol. 7(2), pp. 16-35, 2018.

[27] Jackson and Wallace, " JSON quick syntax reference," *Apress,* 2016.

[28] Wisdom, Vivek and G. Rajat, "An introduction to twitter data analysis in python," *Artigence Inc,* 2016.

[29] Firmino, A. Anderson, d. S. B. Claudio, L. F. A. Andre, O. S. d. A. Davis, F. d. F. Hugo, B. F. Geraldo and C. d. P. Anselmo, "Towards

Metadata Analysis on Opinionated Content in Tweets," *In ICEIS (2),* pp. 314-320, 2016.

[30] Edo-Osagie, Oduwa, S. Gillian, L. Iain, E. Obaghe and D. L. I. Beatriz, "Twitter mining using semi-supervised classification for relevance filtering in syndromic surveillance," *PloS one,* Vols. 14, no. 7, no. e0210689, 2019.

[31] Kumar, Manish, B. Rajesh and R. Dhavleesh, "A survey of Web crawlers for information retrieval," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* Vols. 7, no. 6, no. e1218., 2017.

[32] Morstatter F., J. Pfeffer and H. Liu, "When is it baised? assessing the representativeness of Twitter's streaming API," in *proceedings of the 23rd international conference on world wide web*, 2014.

[33] Pfeffer J., K. Mayer and F. Morstatter, "Tampering with Twitter's sample API.," *EPJ Data Science*, vol. 7(1), p. 50, 2018.

[34] Kim, Yoonsang, Jidong Huang and Sherry Emery, "Garbage in, garbage out: data collection, quiality assessment and reporting standards for social media data use in health research, infodemiology and digital disease detection," *Journal of medical Internet research* , vol. no. 2, no. 18, p. e41, 2016.

[35] Chu, Zi, G. Steven , W. Haining and J. Sushil, "Detecting automation of twitter accounts: Are you a human, bot, or cyborg?," *IEEE Transactions on dependable and secure computing,* Vols. 9, no. 6, pp. 811-824, 2012.

[36] Alothali, Eiman, Z. Nazar, A. M. Elfadil and A. Hany, "Detecting social bots on twitter: a literature review," in *In 2018 International conference on innovations in information technology (IIT)*, 2018.

[37] Samper-Escalante, D. Luis, L.-G. Octavio, M. Raul and M.-p. Miguel Angel, "Bot datasets on twitter: Analysis and challenges," *Applied Sciences,* Vols. 11, no. 9, no. 4105, 2021.

[38] F. Morstatter, H. Dani, J. Sampson and H. Liu. "Can one temper with the sample API? Towards neutralizing bais from spam and bot

content," in *in Proceedings of the 25th International Conference Companion on World Wide Web.* , 2016.

[39] Davis, Clayton Allen, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. "Botornot: A system to evaluate social bots." In Proceedings of the 25th international conference companion on world wide web, pp. 273-274. 2016.

[40] O. Varol, E. Ferrara, A. D. Clayton, M. Fillippo and F. Alessandro, "Online Human-Bot Interactions: Detection, Estimation, and Characterization," *Cornell University Library ICWSM, AAAI, ArXiv, arXiv: 1703.03107,* p. pp. 18, 27 Mar 2017.

[41] ""Botometer by OSoMe," [Online]. Available: http://botometer.iuni.iu.edu/#!/faq#whatiscap.," [Online].

[42] D. Troupe, *survey of political bots on Twitter*, PhD Dissertation, Rutgers University-Camden Graduate School,, 2019.

[43] Yang, Kai-Cheng, Onur Varol, Davis C. A., Emilio Ferrara, Alessandro Flammini and Filippo Menczer, "Arming the public with artificial intelligence to counter social bots," *Human Behavior and Emerging Technologies*, no. 1 no. 1, pp. 48-61, 2019.

[44] Sayyadiharikandeh, Mohsen, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. "Detection of novel social bots by ensembles of specialized classifiers." *In Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 2725-2732. 2020.

[45] Yang K., Varl O., Hui P. and Mencezer F., "Scalable and Generalizable Social Bot Detection through Data Selection," in *In Proceeding of the AAA International Conference on Artificial Intelligence*, 2020.

[46] "http://www.spamhaus.org/consumer/definition-Spam," [Online].

[47] Morstatter, Fred, Jürgen Pfeffer, Huan Liu, and Kathleen Carley. "Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose." *In Proceedings of the international AAAI conference on web and social media*, vol. 7, no. 1, pp. 400-408. 2013.

[48] González-Bailón, Sandra, Ning Wang, Alejandro Rivero, Javier Borge-Holthoefer, and Yamir Moreno. "Assessing the bias in samples of large online networks." Social Networks 38 (2014): 16-27.

[49] Zhang, Wei and G. Judith, "Geocoding location expressions in Twitter messages: A preference learning method," *Journal of Spatial Information Science,* vol. 9, pp. 37-70, 2014.

[50] Kocich, David and H. Jiri, "Twitter as a source of big spatial data," in *In SGEM Conference Proceedings, Presented at the SGEM 2016: 1,16th international multidisciplinary scientific geoconference*, 2016.

[51] Kinsella, Sheila, M. Vanessa and O. Neil, "I'm eating a sandwich in Glasgow" modeling locations with tweets," in *In Proceedings of the 3rd international workshop on Search and mining user-generated contents*, 2011.

[52] Mooney, Peter and M. Macro, "A review of OpenStreetMap data," *Mapping and the citizen sensor,* pp. pp. 37-59, 2017.

[53] U. Qazi, M. Imran and F. Ofli, "Geocov19:a dataset of hundreds of millions of multilingual covid-19 tweets with location information,," *SIGSPATIAL,* vol. Spec 12(1), pp. pp.6-15, 2020.

[54] Hasan, S. Syeda, A. Fareal and K. S. Rosina, "Hasan, Syeda Shabnam, Fareal Ahmed, and Rosina Surovi Khan. "Approximate string matching algorithms: a brief survey and comparison," *International Journal of Computer Applications,* Vols. 120, no. 8, 2015.

[55] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," vol. 10(8), pp. 707-710, 1966.

[56] Hyyro and Heikki, "Practical methods for approximate string matching," *Tampere University Press,* 2003.

[57] Ukkonen, Esko. "Algorithms for approximate string matching." *Information and control* 64, no. 1-3 (1985): 100-118.

[58] Schütze, Hinrich, Christopher D. Manning, and Prabhakar Raghavan. "Introduction to information retrieval". Vol. 39. *Cambridge: Cambridge University Press*, 2008.

[59] "https://www.analyticsvidhya.com/blog/2021/07/fuzzy-string-matching-a-hands-on-guide/," [Online].

[60] G. Navarro, " A Guided Tour to Approximate String Matching.," *ACM Computing Surveys.* , no. 33. 10.1145/375360.375365., pp. p36-38, 2000.

[61] "https://www.statology.org/jaro-winkler-similarity/," [Online].

[62] "https://en.wikipedia.org/wiki/Regular_expression," [Online].

[63] "https://towardsdatascience.com/downloading-data-from-twitter-using-the-rest-api-24becf413875," [Online].

[64] Brooker, Phillip, Julie Barnett, John Vines, Shaun Lawson, Tom Feltwell, Kiel Long, and Gavin Wood. "Researching with Twitter timeline data: A demonstration via “everyday” socio-political talk around welfare provision." Big Data & Society 5, no. 1 (2018): 2053951718766624.

[65] "Gnip. http://gnip.com/," [Online].

[66] "Sifter. Available Online : http://sifter.texifter.com/," [Online].

[67] M. A. Smith, B. Shneiderman, N. M. Fraylying, E. M. Rodrigues, V. Barash, C. Dunne, T. Capone, A. Perer and E. Gleave "Analyzing (social media) networks with NodeXL.," in *in Proceedings of the fourth international conference on Communities and technologies.* , 2009.

[68] M. A. Smith, D. Hansen and B. Shneiderman, " Analyzing Social Media Networks with NodeXL: Insights from a Connected World," *Taylor & Francis*, no. ISBN: 978-0-12-382229-1, p. 284 pages, 2011.

[69] W. Ahmed and S. Lugovic "Social media analytics: analysis and visualisation of news diffusion using NodeXL.," Online information Review, 2019.

[70] Y. Wang, J. Callan and B. Zheng, "Should we use the sample? Analyzing datasets sampled from Twitter's stream API.," *ACM Transactions on the Web (TWEB),* pp. 9(3): p. 1-23., 2015.

[71] A. Hino and R. A. Fahey, " Representing the Twittersphere: Archiving a representative sample of Twitter data under resource constraints," *International journal of information management,* pp. 48: p. 175-184., 2019.

[72] Zenasni, Sarah, K. Eric, R. Mathieu and T. Maguelonne, "Spatial information extraction from short messages," *Expert Systems with Applications,* no. 95, pp. pp. 351-367, 2018.

[73] Bean, R. Bradford and H. J.D., "Climatic charts and data of the radio refractive index for the United States and the world," *US Department of Commerce, National Bureau of Standards,* vol. 22, 1960.

[74] "https://www.geeksforgeeks.org/program-distance-two-points-earth/?ref=gcse," [Online].

[75] Park, Jongseo and C. Minjoo, "A K-Means Clustering Algorithm to Determine Representative Operational Profiles of a Ship Using AIS Data," *Journal of Marine Science and Engineering,* Vols. 10, no. 9, no. 1245, 2022.

[76] Sun, Shiliang, L. chen and C. Junyu, "A review of natural language processing techniques for opinion mining systems," *Information fusion,* no. 36, pp. pp. 10-25, 2017.

[77] Didi, Yosra, A. Walha and A. Wali, "COVID-19 Tweets Classification Based on a Hybrid Word Embedding Method," *Big Data and Cognitive Computing,* Vols. 6, no. 2 (2022): 58, 2022.

[78] E. Loper and S. Bird, "Nltk: The natural language toolkit," *arXiv 2002, arXiv:0205028.,* 2002.

[79] M. Hedderich, L. Lange, H. Adel, J. Strötgen and D. Klakow, " A survey on recent approaches for natural language processing in low-resource scenarios.," *arXiv 2020, arXiv:2010.12309.*

[80] " Python for NLP: Sentiment Analysis with Scikit-Learn.https://stackabuse.com/python-for-nlpsentimentanalysis-with-scikit-learn/," [Online].

[81] P. Willett, " The Porter stemming algorithm: Then and now.," Program Electron. Libr. Inf. Syst. 40, 219–223., 2006.

[82] Hasan, R. Md, M. Maisha and A. M., "Sentiment analysis with NLP on Twitter data," in *In 2019 international conference on computer, communication, chemical, materials and electronic engineering (IC4ME2)*, 2019.

[83] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood and G. Choi, " A performance comparison of supervised machine learning models for COVID-19 tweets sentiment analysis.," *PLoS ONE 2021, ,* pp. 16, e0245909., 2021.

[84] W. Zhang, T. Yoshida and X. Tang, " A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Syst. Appl.," vol. 38, p. 2758–2765, 2011.

[85] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for IDF.," *J. Doc.*, 2004.

[86] D. Rumelhart, G. Hinton and R. Williams, "Learning representations by back-propagating error, .," *Nature*, p. 323 (1986) 533–536, 1986.

[87] C. Gupta, R. Palaniappan, S. Swaminathan and S. Krishnan, "Neural network classification of homomorphic segmented heart sounds.," *Applied soft computing*, vol. 7(1), pp. 286-97, 2007 Jan 1.

[88] Hastie, Trevor, R. Tibshirani and J. H. Friedman, "The elements of statistical learning: data mining, inference, and prediction", 2009, vol. Vol. 2, *New York: Spring*er, 2009.

[89] Akhtar, M. Shad, A. Ekbal, S. Narayan and V. Singh, "No, that never happened!! Investigating rumors on Twitter," *IEEE Intelligent Systems*, no. 33, no. 5, pp. pp: 8-15, 2018.

[90] Mamgain, Nehal, E. Mehta, A. Mittal and G. Bhatt, "Sentiment analysis of top colleges in India using Twitter data," *IEEE In 2016 international conference on computational techniques in*

information and communication technologies (ICCTICT), pp. pp: 525-530, 2016.

[91] S. Behl, A. Rao, S. Aggarwal, S. Chadha and H. Pannu, "Twitter for disaster relief through sentiment analysis for COVID-19 and natural hazard crises," *International journal of disaster risk reduction,* no. 55: 102101, 2021.

[92] Pannu, H. Singh, S. Ahuja, N. Dang, S. Soni and A. K. Malhi, "Deep learning based image classification for intestinal hemorrhage," *Multimedia Tools and Applications,* no. 79, pp. 21941-21966, 2020.

[93] LeCun, Yann, Y. Bengio and G. Hinton, "Deep learning," *nature,* no. 521, no. 7553, pp. 436-444, 2015.

[94] Mujahid, Muhammad, E. Lee, f. Rustam, P. B. Washington, S. Ullah, A. A. Reshi and I. Ashraf, "Sentiment analysis and topic modeling on tweets about online education during COVID-19," *Applied Sciences,* Vols. 11, no. 18, p. 843, 2021.

[95] Sharma, D. Kumar, M. Chatterjee, G. Kaur and S. Vavilala, "Deep learning applications for disease diagnosis," *In Deep Learning for Medical Applications with Unique Data, Academic Press,* vol. 1, pp. pp. 31-51, 2022.

[96] K. Zahra, F. O. Ostermann and R. S. Purves, "Geographic variability of Twitter usage characteristics during disaster events.," *Geo Spat. Inf. Sci.,* pp. 20, 231–240, 2017

[97] Z. Cheng, J. Caverlee and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010.

[98] S. S. Ribeiro Jr, D. R. R. Oliveira, T. S. Gonçalves, C. A. Davis Jr, W. Meira Jr. and G. L. Pappa, "Traffic observatory: a system to detect and locate traffic events and conditions using Twitter," in *5th ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, 2012.

[99] C. White, "Social media and meta-networks for crisis mapping: Collaboratively building spatial data for situation awareness in

disaster response and recovery management", University of California Santa ra Center fo Spatial Studies, 2010.

[100] L. Lutz Bornmann, R. Haunschild and V. M. Patel, "Are papers addressing certain diseases perceived where these diseases are prevalent? The proposal to use Twitter data as social-spatial sensors.," *Plos one,* vol. 15(11) e0242550, 2020.

[101] Q. Zhang, P. Jin, S. Lin and L. Yue, "Extracting focused locations for web pages.," in *In: International Conference on Web-Age Information Management.*, 2011.

[102] D. Inkpen, J. Liu, A. Farzindar, F. Kazemi and D. Ghazi, "Location detection and disambiguation from twitter messages.," *J. Intell. Inf. Syst.*, pp. 49, 237–253, 2017.

[103] R. Reed and M. J. Robert, "Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks," *Mit Press*, 1999 Feb 17.

Appendix (1)

Table (4.3) Sample of Location CSV file

	Location	Text	Follower-Count	Friends-Count	Created-at	User-Id
0	Feeding the ducks	@Arbeit_Fish remember when we were told covid has this distinctive dry cough and we'd all know if we had it	3362	4000	Fri Apr 09 08:12:40 +0000 2021	5774442.0
1	she/her | 22	i have difficulty breathing	679	183	Fri Apr 09 03:56:40 +0000 2021	92267869.0
2	California, United States	@MoniCute72 @Halfmykingdom @ABC7 This last week We are running into low grade temps, runny nose , body aches, cough https://t.co/NRo5CkIfvz"	1078	1351	Fri Apr 09 11:30:26 +0000 2021	265776903.0
3	Bostatlantavegas	Got byke a negative covid test, idk why I questioned my elite immune system ????	3748	1485	Fri Apr 16 13:42:14 +0000 2021	32290927.0
4	London, England	" HMP Belmarsh","... therefore 70,000 PEOPLE ARE PRESENTLY INFECTED, AT ANY OTHER TIME THIS WOULD BE A NATIONAL EMERGENCY, https://t.co/gaAEGTdbjd",151,498,	151	498	Fri Apr 16 14:10:54 +0000 2021	1.24e+18
5	North Carolina, USA	@emmajupiter: woke up with a sore throat and have been chugging water but it s still not great T_T so I m gonna try streaming tomorrow i	109	509	Wed Sep 14 22:28:52 +0000 2020	1.13e+18

Table (4. 4) Sample of Place CSV

	Text	Created-at	User-Id	Name	Full Name	Code	Country
0	@velo4321 I don't know how and where I got it. Over the weekend(10-11 July) I'm not feeling well. cough, a little s… https://t.co/cT5ymmqx9j	Wed Feb 14 23:55:41 +0000 2021	425842646.0	Kuala Lumpur	Wilayah Persekutuan Kuala Lumpur	MY	Malaysia
1	was going to Uber eats some soup cause I have a sore throat and feels yucky but that $14 turned into $21 :-) so nah	Wed Apr 14 20:46:58 +0000 2021,	7.47e+17	Cypress	Cypress	CA	United States
2	I think we should stop talking about positive case. Maybe focus on: 1. How many ppl get vaccinated 2. Ppl been ICU… https://t.co/uvlu9IXudS	Tue Jan 13 04:16:28 +0000 2021,	1.22e+18	Kuala Lumpur City	Kuala Lumpur Federal Territory	MY	Malaysia
3	@trustablegenius i noticed before i knew i had covid that i couldn…t smell anything which was why i got tested… i h… https://t.co/Uwx2MUUO4z	Sat May 17 05:13:50 +0000 2021	376151982.0	Aptos	Aptos, CA	US	United States
4	"@FoxNews It is now the unvaccinated that are increasingly being hospitalized, have long term covid symptoms or dyin… https://t.co/xpXeweVGb5	Sat Dec 17 00:03:07 +0000 2020	1.15e+	Tampa	Tampa, FL	US	United Sates
5	@BorisJohnson @10DowningStreet offers #FreedomDay with enhanced chance of getting #Covid_19 and ending up dying in… https://t.co/4ouhl5dXi4	Fri Sep 16 18:08:16 +0000 2020	148760112.0	Yorkshire and the Humber	Yorkshire and the humber,England	GB	United Kingdom

Table 4. 5 Sample of Location + Place CSV file

	Location	Text	Created-at	User-Id	Name	Full Name	Code	Country
0	Newcastle, England Upon Tyne,	That's bad deb. But now you're fixed ????. I've got a fever, hot n cold sweats severe sore throat and coughing. Every https://t.co/qYPVnqxczG	Fri Apr 16 03:07:37 +0000 2021	9.59E+17	Newcastle	Upon Tyne, England	GB	United Kingdom
1	Vancover, British	Vancouver, British Columbia",@KimbleGathers @NBCNews Getting COVID is the same except the symptoms last two weeks and you have permanent effects https://t.co/cJL7DKpyrp	Fri Dec 16 00:38:19 +0000 2020	9.75E+17	Vancover	Vancover, British, Colombia	CA	Canada
2	New Delhi	We Are Still In A State Of A #Shock As My #Dad Was Under Self Isolation On The Day 5th In The #Night He Kept https://t.co/Hjj2SeEUQK	Fri Feb 16 16:40:11 +0000 2021	2680719923	New Delhi	New Delhi, India	IN	India
3	kilifi county	,Kilifi County has today recorded 224 positive cases for the Covid-19 virus. This is the highest number we have ever https://t.co/8HsQnIrhxM,50155,4466, , ,city,Kilifi,"Kilifi, Kenya",KE,Kenya	Fri Feb 16 16:42:52 +0000 2021	3289579859	Kilifi	Kilifi, Kenya	KE	Kenya
4	Bendigo,Victoria	My COVID couplet from @FridayFlashFict: We are locked down together yet apart. The end is in sight but so is the st https://t.co/mzjG0fXnrj	Fri Feb 16 13:56:42 +0000 2021	2949642253	Bendigo	Bendigo, Victoria	AU	Australia

5	Ealing/Berlin	It depends what you determine as a 'case' and a positive test. If I've a case of the flu, it usually means I'm sick🤒 https://t.co/QpIaKAstrg	Fri Oct 16 17:00:01 +0000 2020	62830137	Ealing	Ealing	GB	United Kingdom

Table 4. 6 Sample of Tweets CSV File

	Text	Follower-Count	Friends-Count	Created-at	User-Id
0	@ThatMummyLife My throat feels like it's closing up. I never have difficulty breathing, but it feels like I should be. Fun times."	884	363	Fri Apr 09 19:38:30 +0000 2021,	9.63e+17
1	@methakae Difficulty breathing I suppose? Some of my family have sought medical help but have remained at home.	260	362	Fri Apr 16 08:05:20 +0000 2021,	1.33e+18
2	@PPaulCharles @ThePCAgency What I don�t understand is Switzerland on Amber while having high vaccinated rate and lo� https://t.co/Gd1Xck4C6p	1	7	Fri Apr 16 09:37:52 +0000 2021,	1.37e+18
3	@akhileshsarita @ChinaDaily I think there should be a big light show in New Delhi as well, It mainly shows the fine� https://t.co/kaWDvPJhUc"	13	54	Fri Apr 16 09:39:45 +0000 2021,	1.24e+18
4	Tokyo confirmed 1,271 new cases of novel coronavirus infections on Friday, exceeding 1,000 for the third straight d� https://t.co/SU0NSIjjuS"	62	0	Fri Apr 16 11:06:04 +0000 2021,	1.11e+18
5	@4wding @katrylarenmanor @9NewsAUS Hundreds of people died in 2019 in Australia due to the severe flu outbreak. Did we lockdown?	11	46	Sat Jun 17 00:26:48 +0000 2021,	1.41e+18

Table (4. 7) Sample of Location CSV File

	Location	Longitude	Latitude	Text	User-Id
0	Feeding the ducks			@Arbeit_Fish remember when we were told covid has this distinctive dry cough and we'd all know if we had it	5774442.0
1	she/her | 22			i have difficulty breathing	92267869.0
2	California, United States	-119.417931	36.778259	@MoniCute72 @Halfmykingdom @ABC7 This last week We are running into low grade temps, runny nose , body aches, cough\ufffd https://t.co/NRo5CkIfvz"	265776903.0
3	Bostatlantavegas			Got byke a negative covid test, idk why I questioned my elite immune system ????	32290927.0
4	London, England	-0.136439	51.507359	" HMP Belmarsh","... therefore 70,000 PEOPLE ARE PRESENTLY INFECTED, AT ANY OTHER TIME THIS WOULD BE A NATIONAL EMERGENCY, https://t.co/gaAEGTdbjd",151,498,	1.24e+18
5	North Carolina, USA	-80.793457	35.782169	@emmajupiter: woke up with a sore throat and have been chugging water but it\ufffds still not great T_T so I\ufffdm gonna try streaming tomorrow i\ufffd	1.13e+18

Table (4. 8) Sample of Place CSV File

	Text	Longitude	Latitude	User-Id	Name	Full Name	Code	Country
0	@velo4321 I don't know how and where I got it. Over the weekend(10-11 July) I'm not feeling well. cough, a little s\ufffd https://t.co/cT5ymmqx9j	101.693207	3.140853	425842646.0	Kuala Lumpur	Wilayah Persekutuan Kuala Lumpur	MY	Malaysia
1	was going to Uber eats some soup cause I have a sore throat and feels yucky but that $14 turned into $21 :-) so nah	-95.680346	29.975158	7.47e+17	Cypress	Cypress	CA	United States
2	I think we should stop talking about positive case. Maybe focus on: 1. How many ppl get vaccinated 2. Ppl been ICU\ufffd https://t.co/uvlu9IXudS	101.693207	3.140853	1.22e+18	Kuala Lumpur City	Kuala Lumpur Federal Territory	MY	Malaysia
3	@trustablegenius i noticed before i knew i had covid that i couldn\ufffdt smell anything which was why i got tested\ufffd i h\ufffd https://t.co/Uwx2MUUO4z	-121.892624	36.980614	376151982.0	Aptos	Aptos, CA	US	United States
4	"@FoxNews It is now the unvaccinated that are increasingly being hospitalized, have long term covid symptoms or dyin\ufffd https://t.co/xpXeweVGb5	-82.452606	27.964157	1.15e+	Tampa	Tampa, FL	US	United Sates

5	@BorisJohnson @10DowningStreet offers #FreedomDay with enhanced chance of getting #Covid_19 and ending up dying in🔲 https://t.co/4ouhl5dXi4	1.3918	54.3875	148760112.0	Yorkshire and the Humber	Yorkshire and the Humber,England	GB	United Kingdom

Table (4. 9) Sample of location + Place CSV File

	Location	Text	Longitude	Latitude	User-Id	Name	Full Name	Code	Country
0	Seoul korea	#Breaking: ⁦6 COVID-19 positive cases from Sin BaKyine IDP camp, biggest camp of Rakhine state⁦s Mrauk-U township,⁦ https://t.co/4h9oKdur7h	-64.896335	18.335765	8.25E+17	Virgin Islands, U.S.	Virgin Islands, U.S.	VI	Virgin Islands, U.S.
1	Planet Gaia	1500 people testing positive out of 10 million with a test that's out of 10 million with a test that's not even fu%⁦ https://t.co/OXybCjuV3Z	-118.243683	34.052235	136714419	Los Angeles	Los Angeles	CA	US, United States
2	New Delhi	We Are Still In A State Of A #Shock As My #Dad Was Under Self Isolation On The Day 5th In The #Night He Kept⁦ https://t.co/Hjj2SeEUQK	77.216721	28.644800	2680719923	New Delhi	New Delhi, India	IN	India
3	kilifi county	,Kilifi County has today recorded 224 positive cases for the Covid-19 virus.	39.909327	-3.510651	3289579859	Kilifi	Kilifi, Kenya	KE	Kenya

		This is the highest number we have ever🔴 https://t.co/8HsQnIrhxM							
4	Earth..... between Venus and Mars	,#Delta variant risk means face mask use even if vaccinated against #COVID https://t.co/LLyzQXvhiA	2.294694	48.858093	137806129	Paris	Paris, France	FR	France
5	Islamabad, Pakistan	I injected both Doses of Vaccine But still face mask is necessary #WearAMask #DeltaPlusVariant #COVID🔴 https://t.co/vtZvp3C6aC	73.053810	33.598827	2953466414	Rawalpindi	Rawalpindi, Pakistan	PK	Pakistan

الخلاصة

تويتر، منجم الذهب الذي يوفر معلومات قيمة لا تصدق لجميع قطاعات العمل ، وخاصة قطاع الصحة. نمت سمعته بسرعة مع انتشار فيروس COVID-19 عالميًا .سابقا كانت المعلومات الجغرافية محدودة قبل ظهورهذا المرض. وقد بدأت مجموعات البيانات لـ Twitter بالظهور تزامنا مع ظهور هذا المرض حيث تم اطلاقها بصورة واسعة وكان من ضمنها البيانات المكانية المصاحبة للتغريدة. مثل هكذا بيانات يزداد الطلب عليها للتحري عن مناطق انتشار المرض. اما فيما يخص هذه الاطروحة فان حاجتها للبيانات المكانية تشكل اهمية قصوى. اذ لم يكن بالامكان الاستفادة من مجاميع البيانات التي اطلقها العديد من الباحثين كونها تفتقد للعديد من المعلومات المهمة.

عند جمع البيانات الخاصة بهذه الاطروحة تم استخدام كلمات مفتاحية مسيطر عليها تسمى الجمل المنطقية. حيث تم الوصول الى التغريدات التي يتحدث اصحابها عن اصابتهم بالمرض من بين ملايين التغريدات على منصة Twitter والخاصة بهذا الموضوع حيث تمكنا من تحديد المعلومة المكانية الصحيحة من خلال عدة عمليات. ووفقا لتعدد هذا المعلومة فقد تم تجزئة البيانات الى اربع مجاميع وحسب المعلومة المكانية المتوفرة مع تلك التغريدة. وقد تم فحص هذه المعلومات لتحديد موقع كل مريض وكانت النتيجة تشكيل اربع مجاميع وحسب تواجد المعلومة المكانية التابعة لحقل معين. اما في حالة عدم تواجد المعلومة فيتم فحص نص التغريدة والبحث عن مدخلات مكانية ونسبها لمكان المريض.

تم بناء معجم جغرافي للتحقق من الكلمات الخاصة بالمواقع الجغرافية لكل من المريض ومتابعيه واصدقائه ومن ثم تحديد ايهم اقرب اليه جغرافيا. يتم ذلك من خلال سحب تغريدات المتابعين واستخراج مواقعهم. عملية سحب تغريدات المتابعين تتم وفق فترة زمنية معنية خاصة بفترة اصابة المريض اول مرة. وبعدها تم استخراج المعلومات المكانية الخاصة بهم ومقارنتها مع البيانات المكانية المعروفة للمستخدم. الخطوة التي تليها تم عزل أولئك الموجودين في نفس منطقة المصاب عن طريق تطبيق طريقة التجميع خوارزمية k-mean إلى نقطة إحداثيات خطوط الطول والعرض لكل منهم. وتم عزل أولئك الذين يحصلون على نفس مجموعة التسمية للمصاب.

بعد الانتهاء من عملية العزل المكاني ، يتم جمع تغريداتهم لغرض تصنيف هذه التغريدات ، ولكن قبل البدء في عملية التصنيف ، تم تحديد ثلاث فئات بحيث يمكن نسب التغريدة إلى إحدى هذه الفئات الثلاث. تم استخدام الشبكة العصبية MLP وكان ناتج هذه العملية بدقة ٠,٨٨. ومن خلال توفرالكود الشخصي للمقربين تم جلب بقية بياناتهم الوصفية ومن ضمنها البيانات المكانية

حيث تم عزل المصابين وتوزيعهم على الخارطة مع بيان اعدادهم . في النهاية يتم الإعلان عن المنطقة النهائية كمنطقة مصابة بسبب كثرة الإصابات.

هذا البحث قدم مجموعة بيانات للحالات المصابة بوباء كوفيد تم جمعها خلال سنة كاملة وايضا قدم مجموعة بيانات مؤشرة ومفروزة بحيث كل تغريدة تحمل تاشيرة خاصة بها وفق معايير الاصابة. تمكنت عملية التصنيف من تحديد الشخص المصاب والتعرف عليه من خلال تغريدته.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل - كلية تكنولوجيا المعلومات
قسم البرمجيات

نموذج مطور بالاعتماد على تحليل النص والتغريدات الجغرافية لكشف انتشار كوفيد ١٩

اطروحة مقدمة
الى مجلس كلية تكنولوجيا المعلومات - جامعة بابل وهي جزء من متطلبات نيل درجة الدكتوراه فلسفة في تكنولوجيا المعلومات / برمجيات

من قبل
اقبال عبد الباقي محمد مهدي

إشراف
أ .د احمد سليم عباس جاسم

١٤٤٤هـ **٢٠٢٣م**