Data Analysis and Phase Detection during Natural Disaster based on Social Data

Submitted By

Abdullah-Al-Mosharraf
ID: 2013-1-60-009
Khadiza Rahman
ID: 2013-1-60-028

Supervised By

Dr. Mohammad Rezwanul Huq
Assistant Professor, Department of CSE, EWU.

A Project
In
The Department of
Computer Science and Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science
(Computer Science and Engineering)

East West University

Dhaka, Bangladesh

April, 2017

Declaration

This thesis has been submitted to the department of Computer Science and Engineering, East West University in the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering by us under the supervision of Dr. Mohammad Rezwanul Huq, Assistant Professor at Department of CSE at East West University under the course 'CSE 497'. We also declare that this thesis has not been submitted elsewhere for the requirement of any degree or any other purposes. This thesis complies with the regulations of this University and meets the accepted standards with respect to originality and quality. We hereby release this thesis to the public. We also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature of the candidate

____

Abdullah-Al-Mosharraf

Id: 2013-1-60-009

____

Khadiza Rahman

Id: 2013-1-60-028

Letter of Acceptance

The thesis entitled “Data Analysis and Phase Detection during Natural Disaster based on Social Data” submitted by Abdullah-Al-Mosharraf, ID 2013-1-60-009 & Khadiza Rahman, ID 2013-1-60-028 to the department of Computer Science & Engineering, East West University, Dhaka 1212, Bangladesh is accepted as satisfactory for partial fulfillments for the degree of Bachelor of Science(B.Sc) in Computer Science & Engineering on April, 2017.

Board of Examiners

1______________________________

Dr. Mohammad Rezwanul Huq
Assistant Professor Supervisor
Department of Computer Science and Engineering
East West University, Dhaka, Bangladesh

2______________________________

Dr. Ahmed Wasif Reza
Associate Professor Chairperson (Acting)
Department of Computer Science and Engineering
East West University, Dhaka, Bangladesh

Acknowledgements

During the course of this research we have learned some techniques, some critical analysis, time management, how to work fluently and how to maintain a group.

First of all, we are grateful to the Almighty God. We would like to express our sincere thanks and gratitude to our honorable advisor Dr. Mohammad Rezwanul Huq, Assistant Professor at Dept. of CSE for the continuous support during our thesis study and related research, for his patience, motivation and immense knowledge. His guidance helped us in all the time of research and writing of the thesis. We will always be grateful for having the opportunity to study under him. We heartily thank to our advisor for helping us to complete this work.

We are thankful to all of our teachers, Department of CSE, East West University. We were grateful to all of our primary and secondary school teachers who were our first teachers in our life and initiator of our basic knowledge.

At last we would like to thank our parents and siblings for supporting us spiritually throughout writing this thesis. And we are thankful to all our friends and colleagues. And at last we again thanks to the creator Allah for everything.

Abstract

In social media, twitter is one of the most popular sources for communicating each other. Nowadays it becomes a communicating channel during natural disaster for detecting many disaster events because peoples share their opinions, feelings, activity during the disaster through the twitter. Twitter works as a news service because its news is routed with a limit of 140 characters. Twitter is not simply a platform for broadcasting information, but one of informational interaction. Twitter is rich of contents with user's profile, locations. As from this source we can extract huge data during many disaster events, so we use this platform for mining various disaster relevant tweets during natural disaster. In this paper we use a coding schema for classifying social media data into different phases during a time-critical moment. We manually examine more than 3,500 tweets which were generated during crisis moment.

In this paper we propose a classifier for classifying the disaster phases using social data and identify these types of phases. Here we make an equation for assigning a weight of the relevant keyword. We use KNN (K-nearest neighbor), a machine learning classification algorithm for classifying the disaster relevant tweets. By knowing this phase's disaster response team can detect where disaster will happen, medical enterprise can be prepared to mitigate the damage after disaster and neighborhood area may also be alert to face the disaster. For their benefit we actually classify the disaster relevant tweet into 3 phases that is: pre (preparedness before the disaster event), on (during disaster event), post (impact and recovery after the disaster).We also take the geo-location with latitude and longitude of the disaster event for visualizing it using an earth map.

Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

Social data now has become the popular media for extracting valuable information. Twitter is a major channel in social media and twitter users share news, photos, observation, their feelings and their problem during time-critical moment and other emergencies. During these natural calamities people share their experience using tweet and by seeing this tweet we can understand that disaster is occurring in a particular area. Whenever twitter users say rain is occurring or cyclone is acting to this area, by collecting this type of observation we can detect in which area the natural disaster is occurring. Many researchers examine the disaster related tweet for the sake of people and try to mitigate their sufferings. So we think social network twitter can be a great source for collecting twitter data during natural disaster. We try to collect disaster relevant data from the twitter but Social media is full of irrelevant and redundant noise. So the relevant tweet is extracted based on disaster keyword. The relevant tweet is like stay safe, cyclone is coming this is under pre-status, it's raining is under on-status and help, recovery is under post- status. Our main challenge is building a classifier for different phases and to classify the disaster relevant tweet and to identify different phases during natural disaster. In this paper we build a model for classifying the twitter into three phases (pre, on, post). And we also show the geographical map of the affected are by visualizing and pointing out that type of different phases. The main objective of our work is to help and inform the Medicine Company, people and disaster response team about the impact of the disaster and to alert them. Besides by knowing these type of phases one can clearly understand the nature of the disaster. As we pointed out these phases on the earth map the response team can easily understand where and when this disaster is happening by seeing the map and they can take immediate actions.

1.1 Tweets reflect phases during natural disaster

Social media is rich of contents and also broadcast disaster related tweet during time-critical moment. Twitter users tweet about the disaster event and before the disaster they tweet for making awareness and during the natural disaster they tweet about the condition of the nature and after the disaster has been occurred they tweet for remediation. Here many types of tweet we find and by seeing this tweet we can say in which position a disaster is occurring or it has been occurred or it will occur. Some types of the tweets are like be prepare for the Cyclone Debbie, Cyclone Debbie is coming, etc. and by seeing this type of tweet we can say it is a pre-status. During disaster event people can't tweet much because of the electricity loss or too busy to face the disaster. So on-time status tweet is little but some types of tweet like, it's raining or cyclone is occurring etc. and from these tweet we can say that this place is current disaster zone area. After the disaster, people share their sufferings, impacts and damages happened by the disaster and the tweets are like fund, food, help, injured, death etc. By seeing these kinds of tweet we can say that for remediation food, fund or any kinds of help is needed on that area for the mitigation of damage. Besides tweet also reflect the geographical status of the disaster affected area and by collecting this location with the GPS we also pointed out these three phases on an earth map which will help to make situational awareness.

1.2 Motivation

Our main challenge is identifying different phases of the natural disaster using social data. For this reason we try to classify the disaster relevant data into some phases. This classification provides a framework to predict the pre status before the disaster event, the on status during disaster and the post status after the disaster has occurred to take actionable steps and to produce awareness by the responder for that event.

We select social data for our work because twitter is a social media which is a great platform for producing data and it is like a big data warehouse. So as this platform can produce the disaster relevant data what we need to do our work we use this social data for predicting the status of a tweet.

There are many types of informative tweet can be found from the twitter and if we cannot collect those informative tweets, ultimately we won't be able to predict the status of disaster events. Besides by classifying twitter data during natural disaster one can understand clearly about the disaster event and by seeing the classifying phases of a tweet they can make awareness before the disaster and create a remediation after the disaster. Without detecting the phases they won't have any idea about the disaster event. So detecting different types of phases is needed.

For understanding the disaster event we classify the disaster data into three phases. That is pre (preparedness, public alert, and awareness), on-time (during the disaster event), and post (impact, damage, remediation, recovery). By detecting pre phase, emergency response team can say that this area is under the disaster zone. So they can make awareness between the people that a disaster is coming, so to get prepared. Doctor, nurse and medical enterprise may get ready for curing the affected people. They also make awareness between the neighborhood areasbecause there is a high probability of disaster occurring on that area. Thus awareness can be performed before the disaster takes place and that is on the pre-status phase. When disaster is happening on a certain area, people become very busy to fight against the disaster. In this case, they tweet rarely. Besides on the time of disaster event electricity loss happens in most of the cases and internet connection may cut down. As a result, people mostly cannot tweet during disaster. Thus, during disaster there are little tweet can be found. Most of the time this type of tweet is about the impact of the weather or nature during the disaster and that is in the on-time-status phase. By detecting the post-status phase, response team can know about the damage caused by the disaster and take quick action to recover the damage. Remediation is needed to help the affected disaster zone after the disaster has occurred. By detecting the post phase emergency the affected people should be transferred into the medical center and take immediate medical care and proper medicine to cure them. Besides by seeing the geographical map of that type of phase's people can get alert and response team can create a situational awareness.

1.3 Research Questions

In our paper we use KNN algorithm for classifying twitter data into three phases.

Why do we select keyword from twitter?

During disaster event many tweets can be tracked from the twitter. We use this social media for tracking disaster relevant data and this data is needed for classifying into three phases which we were determined in our work. Besides tracking public data is very easy. Without extracting this disaster data we could not be able to train our algorithm and that's why we use twitter data.

Why do we need classifying disaster phases?

We need a classifier for classifying the disaster data into three classes and detecting each class of a tweet that it is under a particular phase. We can detect the phases by comparing the training data with the test data. So classifying disaster phases is needed for detecting disaster phases of a tweet.

Why do we assign the weight of a tweet?

There may have some extra word in the keyword from which we cannot understand that what type of tweet is it. So basically ignoring the last letter of a word we match every word of a tweet with the relevant keyword which we manually extracted. Based on the matching keyword we assign a weight of a tweet. For declaring the phases of a tweet weight is needed.

Why do we use KNN algorithm?

KNN algorithm is easy to understand and easy to implement. This classification algorithm helps to predict the data with better accuracy. So we use this algorithm so that our result can be much better. We try to predict our data as accurate as we can. By using Euclidean distance we rank our features and take the majority votes. As this method helps to predict data we can compare our actual data with this predicted data and can measure whether our result is good or bad.

1.4 Thesis Overview

Our main goal is detecting disaster phases during natural disaster. So we trained a classification algorithm for classifying disaster data into three phases that is pre-status (before disaster), on-time-status (during disaster) and post-status (after disaster has occurred). So to fulfill our aim first of all, we extract data from social media (twitter) based on hash-tag using twitter 4j API. We also extract data using time-boundary and the bounding box (collecting data with latitude and longitude).

As there was many noisy data, so after extracting tweet based on the disaster hash-tags we clean the data using C++ code. For the data cleaning process we match a tweet word with a relevant keyword (which we extract manually from the collected data) and if any word didn't match then it was considered as a noisy data and if the word match then it was considered as a relevant data. For keyword matching we check and compare the tweet word and relevant keyword in three steps (matching the tweet word with relevant keyword directly, matching the tweet word with keyword considering last letter of a word as a relevant keyword, also consider the increasing length of 1 of a word as relevant).According to this process we assign a weight of a tweet. We also normalize the time into minutes based on the posting time of a file.

After doing this process we use KNN algorithm to classify this relevant data into three phases. Here we use test features as like 1,0,0 (indicates pre-status high); 0,1,0 (on-status high); 0,0,1 (means post-status is high); 1,1,0 (indicates pre-status and on-status both are same and high); 1,0,1 (pre and post both are same and high) etc. confusion occur in the last two cases because in these cases we cannot detect the phases. We use 4 descriptive features (pre-status weight, on-status weight, post-status weight, time in minutes). We split the file of a data into training data and test data according to 2:1 ratio that means the training data is 0.67 (67%) and the test data is 0.33 (33%). Then we trained our algorithm by calculating the Euclidian distance between each test data with all of the training data and we take the nearest distances 5 data as the value of K we assign is 5 and take the majority voted data from it. From that majority voted data we can predict the disaster phases. We extract disaster relevant data manually and compare the actual data with the predicted data which we get after classifying and show how accurate our result is. Then we get the accuracy of our result based on the correct prediction. We also visualize and show the percentage of three phases using a pie-chart and graphically represent the data with the posted time in minutes using axis-chart. After that we create a map using the latitude and longitude of the affected area. In an earth map we try to pointing out the particular area under each disaster phase.

After completing this classification step we evaluate our result by calculating the precision, recall, F1-measure of each phases and the overall accuracy of our project.

Chapter 2

2. Background

Twitter data has an impressive predictive power. Twitter is such a platform where people express and discuss their opinions on current issues. As a result it is helpful for variety of people. In our experiment we actually work with the twitter data which was posted during natural disaster.

We extract social data using twitter 4j API. We give a weight of a status. For calculating this weight we need max time and min time of tweets (in minutes) which is in the file, also need a ratio which we identified using a ternary search and the number of matches with the words of a tweet and selected keyword.In this paper we classify the disaster data into three phases.

For classifying disaster phases we use a machine learning algorithm that is KNN to predict the different phases and also use confusion matrix for calculating the accuracy of our experiment.

We use matplotlib platform for visualizing our work and use pie-chart, axis-chart and Earth-map for visualizing.

2.1 Twitter 4j API

Twitter4J is an unofficial Java library for the Twitter API [12]. With Twitter4J, we can easily use our Java application with the Twitter service. Twitter4J is an unofficial library.

The Twitter Platform connects our website or application with the worldwide conversation happening on Twitter.

For getting the recent tweets based on hash-tag we can use streaming API. We can use this streaming API for collecting recent tweets based on the given hash-tag. By streaming API code we can extract the data and also can search tweets based on the hash-tag. For searching historical tweets we can use search API. In our work we actually search historical tweets based on the disaster hash-tag. For collecting data in our work we use this twitter 4j API using java platform.

2.2 Ternary Search

For finding maximum or minimum point in U-shape graph, ternary search is the best choice. A ternary search [14] is an example of a divide and conquer algorithm. A ternary search determines either that the minimum or maximum cannot be in the first third of the domain or that it cannot be in the last third of the domain, then repeats on the remaining two-thirds.

Assume we are looking for a maximum of $f(x)$ and that we know the maximum lies somewhere between A and B. For the algorithm to be applicable, there must be some value x such that

- for all a,b with $\mathrm{A} \leq a<b \leq x$, we have $f(a) <f(b)$, and
- for all a,b with $x \leq a<b \leq \mathrm{B}$, we have $f(a) >f(b)$.

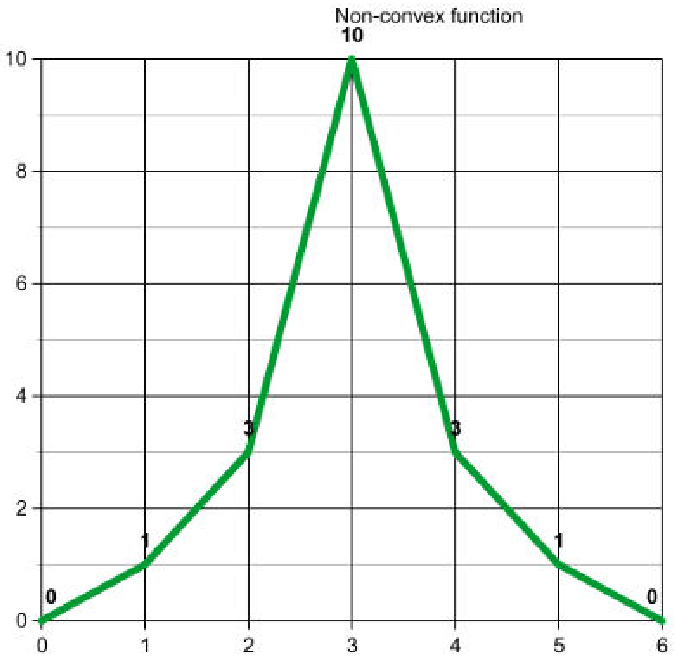


Figure 2.1: A U-Shaped Ternary Search Graph

2.3 K-nearest neighbor (KNN)

A K-nearest neighbor is a machine learning classification algorithm [8].It is a similarity base learning. This is lazy learning algorithm. Whenever we have a new point to classify, we find its K nearest neighbor from the training data and the new point is assigned from the majority of classes. The distance is calculated by using the following measures: Euclidean, Minkowski, Manhattan. It can be used for both classification and regression purposes. In classification problems KNN are most commonly used. The main drawback of KNN is the complexity in searching the nearest neighbor for each sample.

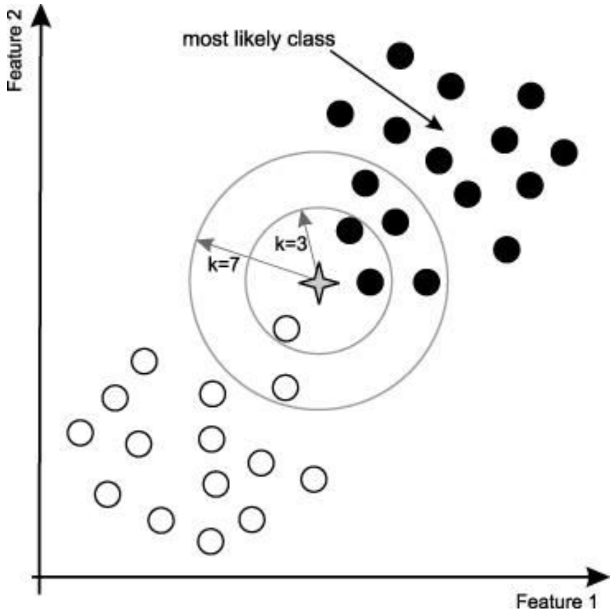


Figure 2.2: An example of KNN-algorithm

2.4 Matplotlib

Matplotlib[13] was originally written by John D. Hunter, has an active development community,and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in 2012.

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

pyplot is a matplotlib module which provides a MATLAB-like interface. matplotlib is designed to be as usable as MATLAB, with the ability to use Python, with the advantage that it is free.

2.5 Confusion matrix

Confusion matrix contains information about actual and predicted classifications done by a classification system and describes the performance of a classifier model [9]. For evaluating the performance[18] of such systems we have to use the data in the matrix. The following table shows the confusion matrix for a two class classifier.

- **TP** is the number of **correct** predictions that an instance is **positive**
- **FN** is the number of **incorrect** of predictions that an instance **negative**.
- **FP** is the number of **incorrect** predictions that an instance is **positive**.
- **TN** is the number of **correct** predictions that an instance is **negative**.

	Predicted		
Actual		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

Table 2.1: Confusion Matrix

Several standard terms have been defined for the 2 class matrix

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

$$\text{accuracy} = \frac{TP+TN}{P+N} \qquad \text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$$

- **Accuracy**: The accuracy means the proportion of the correct prediction and the total number of predictions.
- **TP (true positive) rate:** The TP means the proportion of positives that are correctly identified.
- **FP (false positive) rate:** The FP means the proportions of negatives that are incorrectly identified.
- **Precision**: The precision means the proportion of positively identified that are correct.
- **Recall**: The recall means the proportion of positively identified that are actually positive.
- **F1-measure:** A measure that combines precision and recall. It is the harmonic mean of precision and recall.

Chapter 3

3. Related work

Using social media data many researchers provide disaster relevant information during natural disaster. Some researchers make an automatic method for extracting information nuggets so that they can help professional emergency responders and the twitter data has been categorize into different features such as Caution, Advice, Fatality, Injury, Offers of Help, Missing and General Population Information [1].

Recently researchers also found that for helping emergency responders to act quickly for disaster response and disaster relief, actionable data can be extracted from social media. Ashktorab et al. introduce tweedr, a twitter-mining tool which extracts actionable information during natural disaster for helping disaster responder team. The tweedr pipeline consists of three main parts: classification, clustering and extraction [2]. In the classification phase, they use various classification methods to classify the disaster damage or casualty information. In clustering phase, for merging the similar tweets they use filtering. And in the last extraction phase they extract tokens and phrases of damage, damage type and casualty information.
Here [3] a tweet tracker tool has been proposed. During disaster, by tracking and monitoring disaster related twitter data researchers help Humanitarian Aid and Disaster Relief (HADR) respondents for gaining valuable insights and situational awareness in the affected area

In this paper [4] the geographic locations of tweets of a category was mapped to familiarize with the location of the disaster event so that response team can know where the disaster has occurred.

To understand the disaster event social media messages has been separated into 4 categories (mitigation, preparedness, emergency response, and recovery) and this framework has been done with the relevant tweet to take action quickly and efficiently in the impacted communities. This is useful for emergency managers to identify the transition between phases of disaster management so that they can know about the suffering people, damaged areas and alert them or relief them from suffering and recover the damage assessment. Here [7] also describe about the use of disaster phases (mitigation, preparedness, emergency response, and recovery) which has assisted both disaster researchers and managers. In this paper it has been suggested that the use of disaster phases can improve the theoretical and applied dimension of the field during disaster periods. Here it also mentioned that disaster researchers have used disaster phases to organize important findings and recommendations about disasters. Here also discuss that preparedness fills in where mitigation efforts cannot reduce the effects of a disaster, response occur right after the disaster, recovery is related to following the response period.

Here in this paper[5] it has been described that annotating social media data with geographic coordinate is more valuable for quickly finding out the area under the victim. But in social media geographic information is rarely found. Twitter user locations are being estimated by propagating the locations of GPS-known users across a Twitter social network. Using this publicly visible twitter data, a method has been invented to locate the overwhelming majority of active Twitter users by examining their locations. The algorithm assigns a location to a user based on the locations of their friends by calculating the min, median and max distance between their friends. Here a technique has been developed to estimate per-user accuracy of the geo-tagging algorithm.

The contributions of this paper [6] are to introduce AIDR (Artificial Intelligence for Disaster Response), a platform designed to perform automatic classification of crisis-related social data. The objective of AIDR is to classify messages that people post during disasters into a set of user-defined categories of information (e.g., “needs”, “damage”, etc.) For this purpose, the system continuously ingests data from Twitter, processes it (i.e., using machine learning classification techniques) AIDR has been successfully tested to classify informative vs. non-informative tweets posted during the 2013 Pakistan Earthquake. Overall, they achieved a classification quality.

Here [10] the benefit of the automated geo-location of social media messages has been declared. Here they also analyze that since different people, in different locations write messages at different times, these factors can significantly vary the performance of a geo-location system over time. They demonstrate cyclical temporal effects on geo-location accuracy in Twitter, as well as rapid drops as test data moves beyond the time period of training data. They also show that temporal drift can effectively be countered with even modest online model updates. In this paper they consider the task of tweet geo-location, where a system identifies the location where a single tweet was written. And in this [11] paper it has been investigated and improved on the task of text-based geo-location prediction of twitter users. They present an integrated geo-location prediction framework and investigate what factors impact on prediction accuracy. Here they evaluate the impact of temporal variance on model generalization, and discuss how users differ in terms of their geo-locatability.

The contribution of this paper [15] is that conducting a systematic comparative analysis of nine state-of-the-art networkbased methods for performing geo-location inference at the global scale, controlling for the source of ground truth data, dataset size, and temporal recency in test data. This analysis identifies a large performance disparity between that reported in the literature and that seen in real-world conditions. To aid reproducibility and future comparison, all implementations have been released in an open source geo-inference package.

Lie et al. [16] declare that Twitter data contains valuable information that has the potential to help improve the speed, quality, and efficiency of disaster response. However, supervised learning algorithms require labeled data to learn accurate classifiers. Unfortunately, for a new disaster, labeled tweets are not easily available, while they are usually available for previous disasters. Furthermore, unlabeled tweets from the current disaster are accumulating fast. Experimental results suggest that, for some tasks, source data itself can be useful for classifying target data. However, for tasks specific to a particular disaster, domain adaptation approaches that use target unlabeled data in addition to source labeled data are superior.

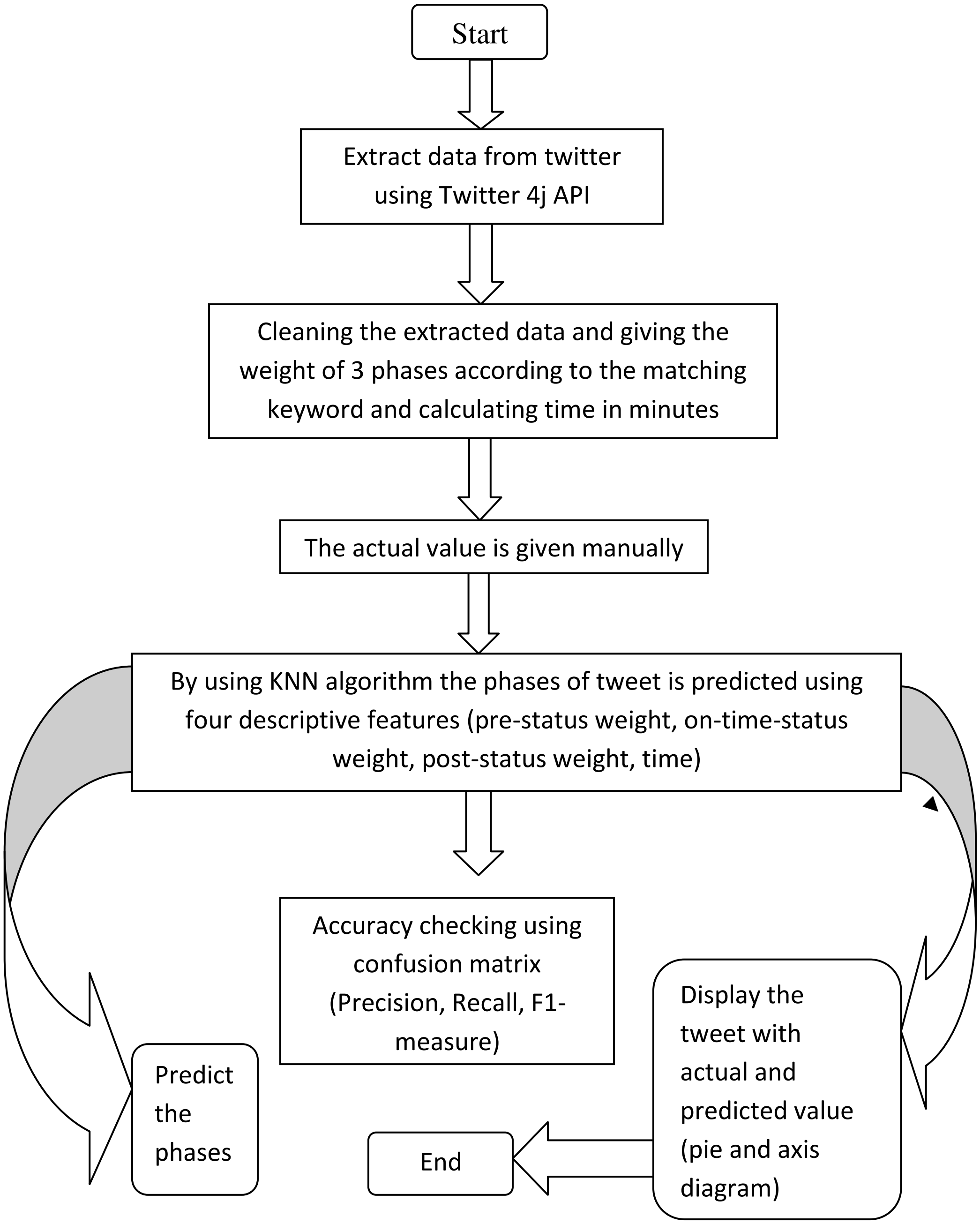
This paper[17] presents a CyberGIS framework that can automatically synthesize multi-sourced data, such as social media and socioeconomic data, to track disaster events, to produce maps, and to perform statistical analysis for disaster management. In this framework, Apache Hive, Hadoop, and Mahout are used as scalable distributed storage, computing environment and machine learning library to store, process and mine massive social media data. The proposed framework is capable of supporting big data analytics of multiple sources. A prototype is implemented and tested using the 2011 Hurricane Sandy as a case study.

Chapter 4

4. Working Procedure

We describe here the working procedure for the solution of the problem we discussed above

4.1 Project Flow Chart



Start
Extract data from twitter using Twitter 4j API
Cleaning the extracted data and giving the weight of 3 phases according to the matching keyword and calculating time in minutes
The actual value is given manually
By using KNN algorithm the phases of tweet is predicted using four descriptive features (pre-status weight, on-time-status weight, post-status weight, time)
Predict the phases
Accuracy checking using confusion matrix (Precision, Recall, F1-measure)
Display the tweet with actual and predicted value (pie and axis diagram)
End

Figure 4.1.1: The overall flow chart of our working procedure**.**

4.1.1. Data Collecting

We collect tweet from twitter using Twitter4j API in JAVA based on Hash-tags. For example, some Hash-tags are like (#disaster, #damage, #flood, #tornado, #cyclone). Using those Hash-tags we collect randomly 3,500 tweet within no bounding-box and no time-bound.

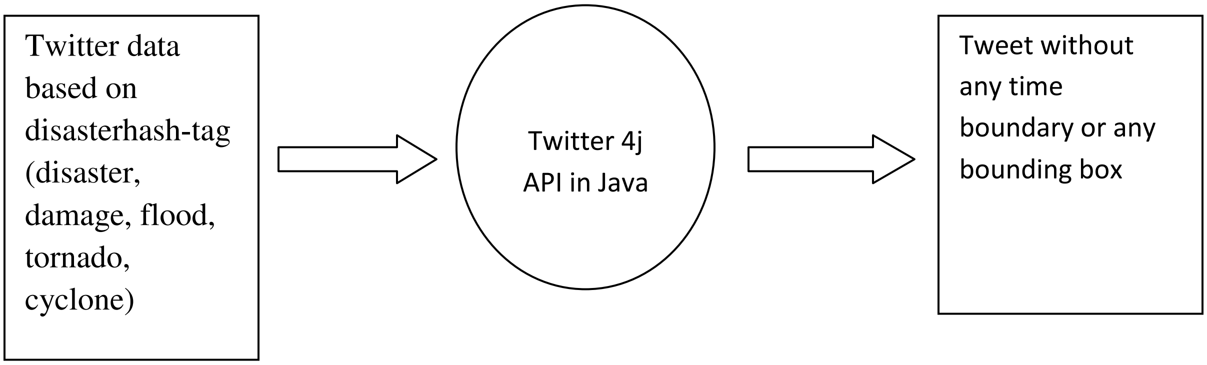

Twitter data based on disasterhash-tag (disaster, damage, flood, tornado, cyclone)
Twitter 4j API in Java
Tweet without any time boundary or any bounding box

Figure 4.1.2: Extracting tweet based on disaster hash-tag with no time boundary and no bounding box.

4.1.2 Keyword Selection

Previously we said that we collect 3,500 tweets which are related to natural disaster. From those tweet manually we select some keyword and manually classify them into 3 phases. We select total number of 65 keywords and classify them in (Pre-Status, On-time-Status and Post-Status)[3 Phases].

Those types of Keywords are:

Status	Keyword	Number Of Keyword
Pre-Status	ready, warning, alerts, preparedness, safety, tornadosafety, staysafe, declaration, plan, prepared, ready, tornadowarning, takecover, awareness, Alert, Beware, prepared, team, EmergencyPreparedness, FloodReady, HaveAPlan, emergency.	**21**
On-time-status	thunderstorm, outbreaks, snowstorm, lightning, blizzard, clouds, "rain", badweather, stormchasers, pray, Drowning, overflow, damaging, wind, sky, flooding, flood, after, stay	**19**
Post-status	disasterrecovery, shelter, donations, recovery, victims, rebuilding,damaged, relief, house, response, responder, help, helping, Disaster Solutions, FLOOD DAMAGE, dead, effects, volunteers, restoration, impacts, survivors, EmergencyRelief, startup, experienced, team, strong	**25**

Table 4.1: Somedisaster relevant keyword

Based on these keywords we remove the noisy data and give the weight of relevant tweet.Sometimes there aresome keywords that are under a phase of three but due to the presence of another word the meaning of the tweet turns into another phase. Example: “Response teams are coming” and “response teams are working” here the 1st tweet belongs to “Pre-Status” and the 2nd tweet belongs to “Post-Status”.

That’s why we keep another checking for this kind of problem, Here the list of this type of keyword:

Keyword	Related Word	Belongs to
Response	Way	Pre
Safety	Damage	Post
House	Will	Pre
Damage	Will	Pre
Help	Can	Pre
Shelter	Will	Pre
Prepared	Was	Post
Rain	Coming	Pre
Cloud	Coming	Pre
Volunteer	Join	Pre
Rain	Stop	Post
Rain	Calm	Post
Safety	Working	Post
Help	Times	On

Table 4.2: Somedisaster relevant keyword with related word

4.1.3 Weight assigning of a tweet

An equation,Weight of a tweet = (maxtime-mintime)*ratio*number_of_matches.

Here, time computed according to minutes.

Number of matches is the number of keyword matches with the words of a tweet. We consider three rules for countingmatches.

1. Directly and equally the keyword and a word of tweet is matches, example=("rain"[Keyword] and "rain"[A word of tweet]).
2. If there is any last letter like '.'(dot), ','(comma), '?'(intarogation symbol) and '!'(Exclamatory symbol) then we ignore it and check without it, example=("rain"[Keyword] and "rain."[A word of tweet]).
3. If a keyword matches completely but the word_of_tweet_length - Keyword_length = 1 then we accept that as a matches. ("wind"[Keyword] and "winds"[A word of a tweet]).

Here Ratio=0.75

This valueisfindingout by using **ternary search** manually

Lower bound of ternary search is 0.0 and upper bound of ternary search is 1.0

Here the calculation of ternary search,

L= 0, R=1.0

Mid1= 0.33	88.095	83.673	87.8048	89.361	90.697	Avg:87.9262
Mid2=0.67	87.5	87.93	87.5	86.4864	92.6829	Avg:88.41986

L = 0.33, R =1.0

Mid1=0.55	86.0	78.84	87.5	87.755	95.1219	Avg:87.043
Mid2=0.77	83.673	90.566	91.489	83.018	95.833	Avg:88.915

L =0.55, R =1.0

Mid1 = 0.7	97.5609	87.23404	94.736	90.556	90.243	Avg:92.065
Mid2=0.85	94.736	82.875	84.444	89.583	87.755	Avg:87.875

L = 0.55, R = 0.85

Mid1 = 0.65	89.361	88.636	88.888	91.304	88.235	Avg:89.284
Mid2=0.75	87.8084	91.111	97.8723	98.0	90.196	Avg:92.997

L = 0.65, R = 0.85

Mid1=0.72	91.667	91.489	92.0	89.361	90.0	Avg:90.03
Mid2=0.78	88.888	85.416	91.667	89.473	91.667	Avg:89.422

L = 0.65, R = 0.78

Mid1=0.69	90.264	86.36	90.476	84.444	92.5	Avg:88.808
Mid2=0.73	94.736	87.8048	95.8337	92.105	92.158	Avg:92.327

L = 0.69, R = 0.78

Mid1=0.72	91.667	91.489	92.0	89.361	90.0	Avg:90.905
Mid2=0.75	87.8048	91.111	97.8723	98.0	90.196	Avg:92.997

L = 0.72, R = 0.78

Mid1=0.74	90.697	91.2280	97.727	89.36170	88.888	Avg:91.580
Mid2=0.76	88.636	86.7294	86.956	90.909	90.566	Avg:88.759

L = 0.72, R = 0.76

Mid1=0.73	94.736	87.8048	95.8337	92.185	92.158	Avg:92.325
Mid2=0.75	87.8048	91.111	97.8723	98.0	90.196	Avg:92.997

L = 0.73, R = 0.76

Mid1=0.74	91.667	91.489	88.888	89.361	96.0	Avg:91.2280
Mid2=0.75	87.8048	91.111	97.8723	98.0	90.196	Avg:92.997

We know the value of 0.73, 0.74, 0.75, 0.76, from them we get the best accuracy from 0.75[Accuracy = 92.997]

4.1.4 Giving the Ground Value

We give the ground value of all tweets.We do it manually by reading all cleanedtweets and determine their original value.

4.1.5 Predict the Tweet

Now we use KNN [K=5] to predict the phase of a tweet. There are 67% data using as a training-set and 33% using as a test set of a datafile. There are 4 descriptive features in dataset [Pre-Status, On-time-Status, Post-Status, Time] and 3 target features, and for an example the weight of the Pre-status are higher than the other 2 target features would be [1, 0, 0].

Python codeselects 5 nearest training datasetby usingEuclidian distance.

The three phases point is counted from these 5 data and returns the phase which gets the maximum number of vote and predictsthe selected phases on which type of phases it is.

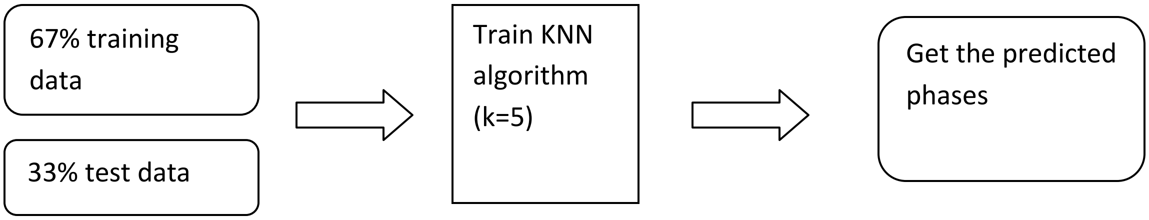


Figure4.1.3: Training KNN algorithm to get the predicted phases.

4.1.6 Accuracy Checking

Now we compare the predicted value with the actual ground value. We build a confusion matrix based on the average of 5 iterations.

4.1.7 **Phase Detection**

We take the last 10% data whose predicted value and actual value is equal. Then count which phase occurs most from these tweets. Then select that**Phase.**

4.2. Pseudo code of our project

4.2.1. Pseudo code of weight and clean

```
CLEAN_TWEET(tweet_data_set)
pre[1...pre_len] be a new array
on[1...on_len] be a new array
post[1...post_len] be a new array
declare a map structure confusion
keyword[1...n] be a new array
number_of_word=tweet_data_set.len
initializepre_weight to zero
initializeon_weight to zero
initializepost_weight to zero
struct data{
  initializepre_point,on_point,post_point to zero
  initialize time to zero
  initializelog,lat to zero };
initializemini_time to maximum_value
initializemaxi_time to minimum_value
for x=1 to number_of_word
  If tweet[x] == end_mark_of_tweet //end_mark_of_tweet
       If pre_weight==0 and on_weight==0 and post_weight==0
            continue
       Else
            data random // declare a 'data' type variable
            random.pre_point=pre_weight
            random.on_point=on_weight
            random.post_point=pre_weight
            random.time=a
            randon.lon=b
            randon.lon=c
            push_back random into tweet //(clean_tweet)
            pre_weight=0
            on_weight=0
            post_weight=0
  If tweet[x] == time_mark_of_tweet
       a=calculate_the_time(year, month, date, hours, minutes,
       secound)
       maxi = max(a, maxi_time)
       mini = min(a, mini_time)
       b=logitude
       c=lattitude
  for i=1 to pre_len
       If pre[i]==tweet[x]
            add one to pre_weight
  for i=1 to on_len
       If on[i] == tweet[x]
            add one to on_weight
  for i=1 to post_len
```

```
       If post[i] == tweet[x]
            add one to post_weight
for each x (- tweet.data
       pre_preference
       on_preference
       post_preference
       x.time = x.time-mini
       x.pre_point = (maxi_time-mini_time) * ratio * x.pre_point
       x.on_point = (maxi_time-mini_time) * ratio * x.on_point
       x.post_point = (maxi_time-mini_time) * ratio * x.post_point
       initializepre_preference, on_preference, post_preference to zero
       If x.pre_point == x.on_point and x.pre_point == x.on_point
            pre_preference=on_preference=post_preference=1
       Else If x.pre_point == x.on_point and x.pre_point>x.post_point
            pre_preference=on_preference=1
       Else If x.pre_point == x.post_point and x.pre_point>x.on_point
            pre_preference=post_preference=1
       Else If x.post_point == x.on_point and x.post_point>x.pre_point
            post_preference=on_preference=1
       Else If x.pre_point>x.on_point and x.pre_point>x.post_point
            pre_preference=1
       Else If x.on_point>x.pre_point and x.on_point>x.post_point
            on_preference=1
       Else If x.post_point>x.pre_point and x.post_point>x.on_point
            on_preference=1
       printx.pre_point,x.on_point,x.post_point,x.time, pre_prefenence,on_preference,post_preference,x.lon,x.log
```

4.2.2. Pseudo Code of KNN, phase detection and accuracy checking

```
KNN(dataset_of_CleanData)
declare a structure array testSet
declare a structure array trainingSet
declare a string array phase
initailize split to 0.67
Split_Data(dataset_of_CleanData,split,trainingSet,testSet)
initialize k to 5
declare a structure array neighbours
declare a string variable response
initialize match to zero
for x to number_of_test_data
       neighbours<-Get_Neighbours(trainingSet,testSet,k)
       response<-Get_Response(neighbours)
       If actual == prediction
             insert into phase[]
             add one to match
reverse the array of phase
phase_length = phase.len
initialize p1,p2,p3 to zer0
for x =1  to phase_length*0.10
       If phase[x] == "Pre-status"
             add one to p1
       If phase[x] == "On-time-status"
             add one to p2
       If phase[x] == "Post-status"
             add one to p3
multiply p2 with 2
If p1 and p2 and p3 is maximum
       print "Phase: Confuse"
If p1 and p2 is maximum
       print "Phase: Either Pre-Status or On-time-Status"
If p1 and p3 is maximum
 print "Phase: Either Pre-Status or Post-Status"
If p2 and p3 is maximum
 print "Phase: Either On-time-Status or Post-Status"
If p1 is maximum
       print "phase : Pre-status"
If p2 is maximum
       print "phase : On-time-status"
If p3 is maximum
       print "phase : Post-status"
declare a double variable accracy
accuracy<-Get_Accuracy(phase_length,testSet.len)
print accuracy
printpie_chat_actual
printpie_chat_predict
printaxis_chat_actual
printaxis_chat_predict
printearth_map_chat_actual
printearth_map_chat_predict
```

4.3. Visualization of Our Project

Now, In Queensland, Australia recently a Cyclone named Debbie has been occurred. This disaster hits the Queensland in the evening of 28th March, 2017. We collect tweet based on two hash-tags[eg. #Cyconedebbie, #disaster] and also based on 3 steps and this steps are [26th march to 27th march], [28th march to 29th march] and [30th march and 31th march].

We collect 1000 tweets in 1st step, 300 tweets in 2nd step and 700 tweets in 3rd step. Total 2000 tweets.

Now we create 4 data-files, three filescarry these 3 steps and the 4thfile carries all of the tweets of these 3 steps.

Then we clean the data. In file1 [Step 1] we get 158 relevant data, in file2 [Step 2] we get 79 relevant data and in file3 [Step 3] we get 193 relevant data and finally in file4 [all tweet] we get 426 relevant data.

Then we collect theactual value of all these tweets manuallyand trained the KNN algorithm to predict the phase of these data.

4.3.1.File 1:

We can manually say that file 1 has Pre-Status tweet. There is some comparison:

4.3.1.1. Pie chart

Actual

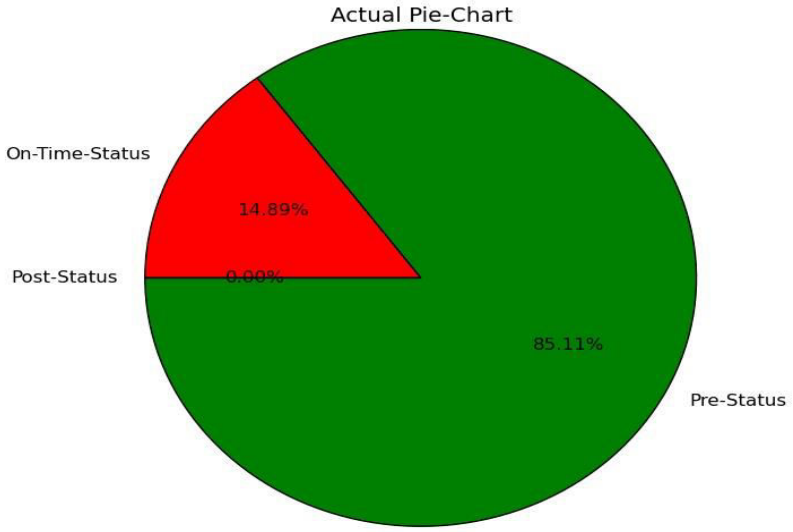


Figure 4.3.1: Pie-chart of file 1 (Actual)

Here we visually represent the percentage ofthe actual value of phase1 using pie chart. From here we can see that in pre-status phase the percentage of data is 85.11, in on-status phase the percentage is 14.89, whereas in post-status phase there is 0.00% data. As in pre-status phase the percentage is higher than another two phases so we can say that this chart indicates that this data is under pre-status phase. When we collect data from Cyclone-Debbie there we see that whenever people tweet for the awareness before the cyclone in that time cyclone has been began and so during the pre-status phase people was posting on-status tweet. So we get 14.89% tweet in on-time-status. From this chart we can observe that here is no tweet in post-status phase and we know in pre-status phase generally people does not post any type of tweet which are related with post-status phase.

Predicted

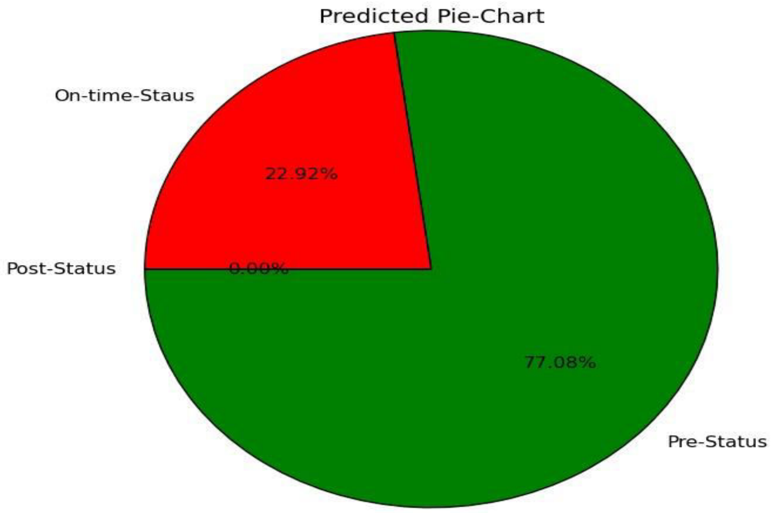


Figure 4.3.2: Pie-chart of file 1 (Predicted)

This figure describes the visual representationofthe predicted value of phase1 using pie-chart**.** Thepercentage of the three phases is pointing out in this figure. From this figure we can see that the percentage of predicted data in pre-status phase is 77.08, in on-status phase the predicted percentage is 22.92, whereas in post-status phase it is 0.00%.We can predict that in pre-status phase the percentage is also higher and here also we were able to predict the on-status phase. So we correctly predict that this is in pre-status phase. Like actual data we could predict that here is no post-status phase.

4.3.1.2. Axis-chart:

Actual

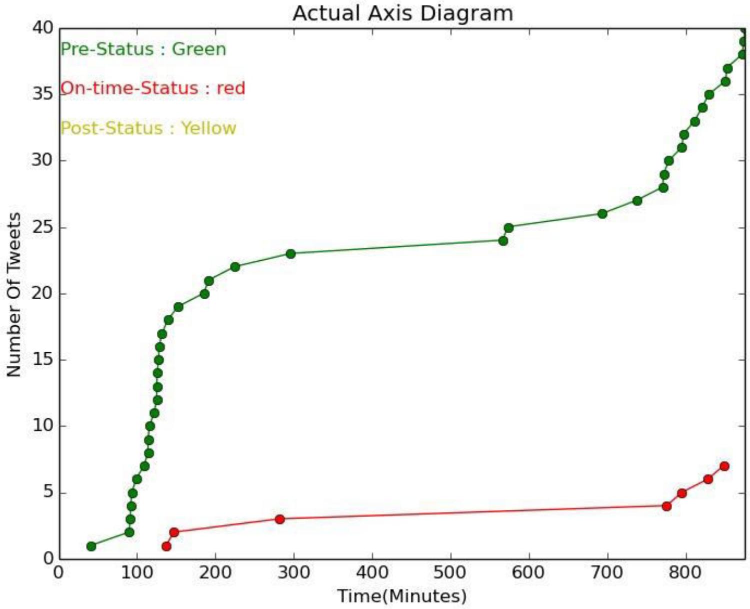


Figure 4.3.3: Axis diagram of file1 (Actual)

In this figure, the actual data of phase1 is graphically represented using axis diagram. Here we visualize three phases using three colors (green for pre, red for on, yellow for post). In this figure the x-axis represent the Time in minutes and the y-axis is the Number of Tweets. The circle point indicates the data of each phases including time represented by the color. From this diagram we can understand that in pre-status phase there are a lot of data and in on-status phase there are little bit data and here is no yellow data so post-status phase is empty. By observing the data we can understand that this is pre-status data because the number of tweet is higher in pre-status phase.

Predicted

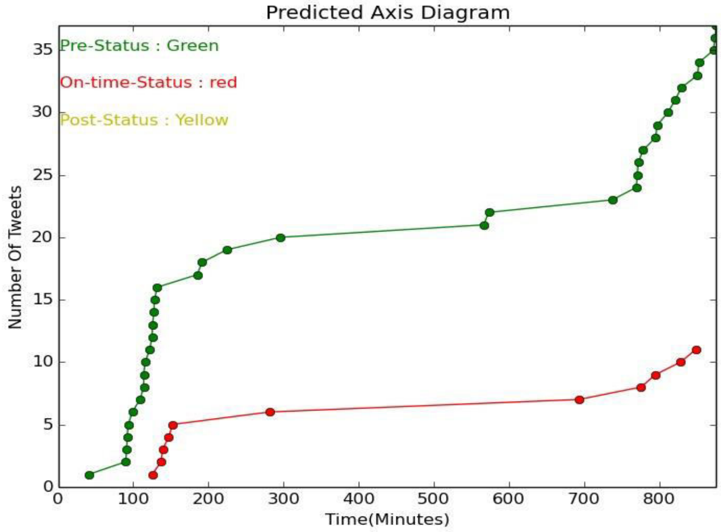
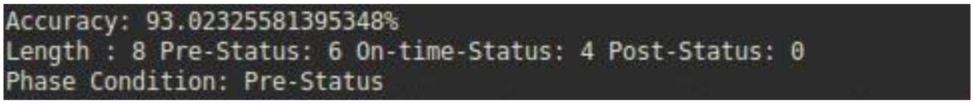


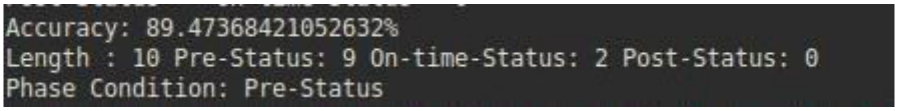
Figure 4.3.4: Axis diagram of file1 (Predicted)

In this figure, the predicted data of phase1 is graphically represented using axis diagram. Here we visualize three phases using same three colors (green for pre, red for on, yellow for post) as actual. This figure represents Time (minutes) vs. Number of Tweets. From this diagram we can say that we were able to correctly predict the disaster phases as a pre-status phase because the data (represented by circle point) in pre-status phase is higher than the on-status phase. As like actual data here is no post-status tweet.

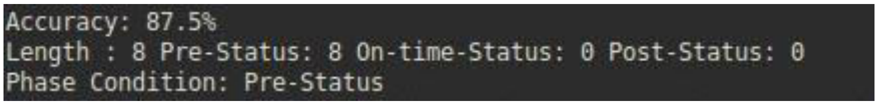
4.3.1.3. Accuracy and phase detection of file1:



```
Accuracy: 93.02325581395348%
Length : 8 Pre-Status: 6 On-time-Status: 4 Post-Status: 0
Phase Condition: Pre-Status
```



```
Accuracy: 89.47368421052632%
Length : 10 Pre-Status: 9 On-time-Status: 2 Post-Status: 0
Phase Condition: Pre-Status
```



```
Accuracy: 87.5%
Length : 8 Pre-Status: 8 On-time-Status: 0 Post-Status: 0
Phase Condition: Pre-Status
```

By observing the result we can say that this is Pre-status because Pre-status is higher than other two. The accuracy is 89.473%.

4.3.2. File 2:

Similarly we can manually say that file 2 has on-time-Status tweet. There is also some comparison:

4.3.2.1. Pie chart:

Actual

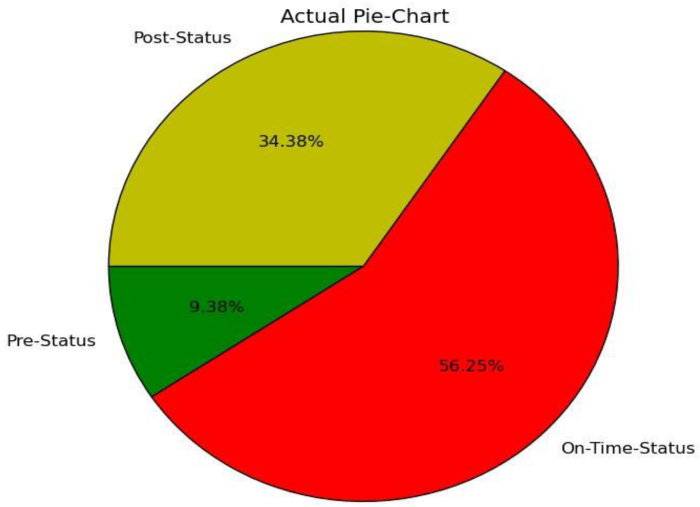


Figure 4.3.5: Pie-chart of file 2 (Actual)

We visually represent the percentage ofthe actual value of phase2 using pie chart. From here we can see that in pre-status phase the percentage of data is 9.38, in on-status phase the percentage is 56.25, whereas the percentage in post-status phase is 34.38. As in on-status phase the percentage is higher than another two phases so we can say that this chart indicates that this data is under on-status phase. When we collect data from cyclone-debbie there we see that whenever cyclone was occurringat that time because of the flow of water flood also occurred and the flood is post-status phase. So during the on-time-status phase people was also posting on-status tweet. Though this is under on-status phase we see a little bit pre-status phase because when disaster is happening in an area the neighbor area may also be affected by disaster. So during a disaster event an awareness tweet may also be post for the neighborhood area. From this chart we can observe that here is no tweet in post-status phase and we know in pre-status phase generally people does not post any type of tweet which are related with post-status phase.

Predicted

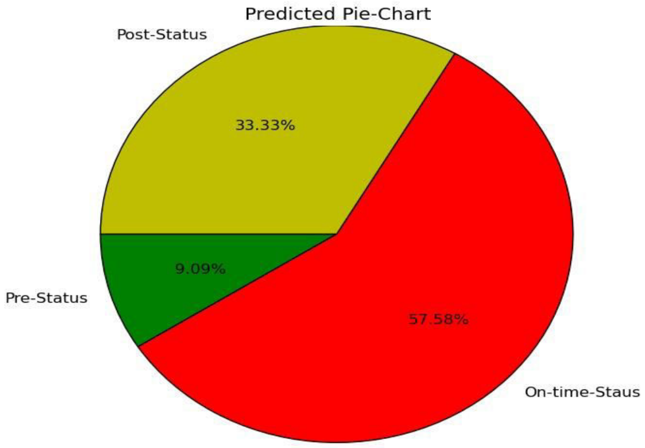


Figure 4.3.6: Pie-chart of file 2 (Predicted)

This figure describes the visual representation of the predicted value of phase2 using pie-chart**.** The percentage of the three phases is pointing out in this figure. From this figure we can see that the percentage of predicted data in pre-status phase is 9.09, in on-status phase the predicted percentage is 57.58, whereas in post-status phase the predicted percentage is 33.33. We could predict that in on-status phase the percentage is higher than the two and could predict that here is also post-status phase and a little bit pre-status phase though the percentage of the actual data with predicted data is not same. After all of the observation we could say that we correctly predict that this is in on-status phase.

4.3.2.2. Axis-chart:

Actual

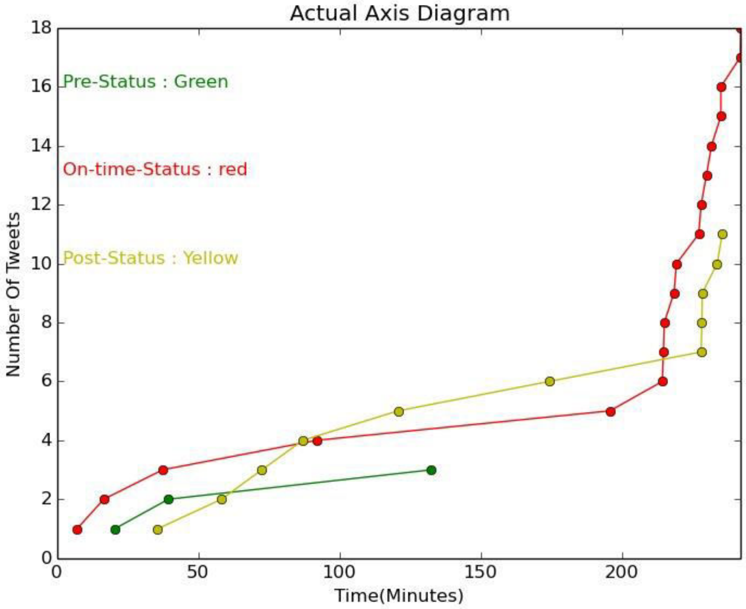


Figure 4.3.7: Axis diagram of file2 (Actual)

In this figure, the actual data of phase2 is graphically represented using axis diagram. Here we visualize three phases using three colors (green for pre, red for on, yellow for post). In this figure the x-axis represent the Time in minutes and the y-axis is the Number of Tweets. The circle point indicates the data of each phases including time represented by the color. From this diagram we can understand that in on-status phase there are a higher data than in on-status phase and a little bit of data in pre-status phase. By observing the data we can understand that this is in on-status phase though there are post-status data because of the flooding during cyclone.

Predicted

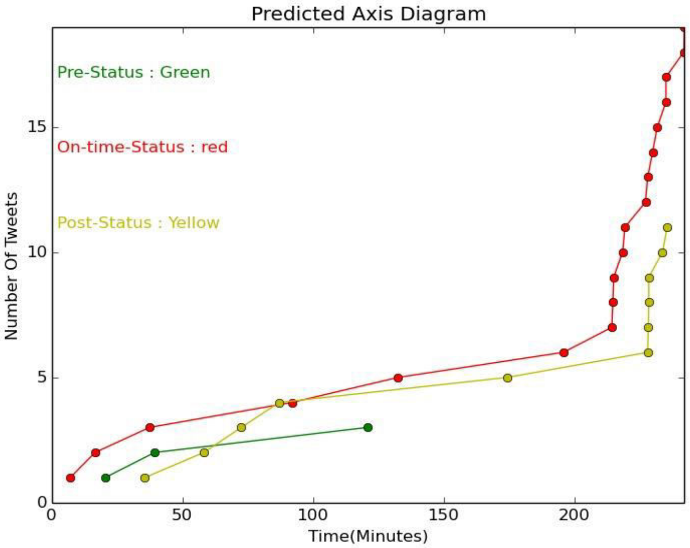
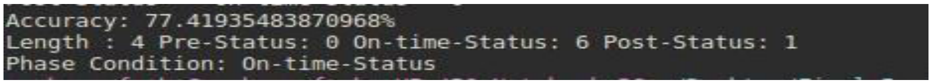


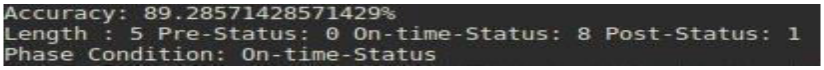
Figure 4.3.8: Axis diagram of file2 (Predicted)

In this figure, the predicted data of phase2 is graphically represented using axis diagram. Here we visualize three phases using same three colors (green for pre, red for on, yellow for post) as actual. This figure represents Time (minutes) vs. Number of Tweets. From this diagram we can say that we were able to correctly predict the disaster phases as an on-status phase because the data (represented by circle point) in on-status phase is higher than the post-status phase. Here we can predict 3 data as a pre-status data for alerting neighborhood area of disaster located area.

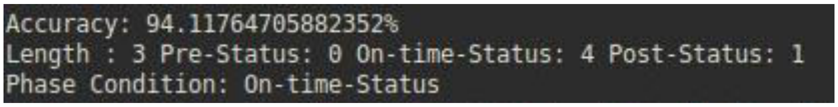
4.3.2.3. Accuracy and phase detection of file2:



```
Accuracy: 77.41935483870968%
Length : 4 Pre-Status: 0 On-time-Status: 6 Post-Status: 1
Phase Condition: On-time-Status
```



```
Accuracy: 89.28571428571429%
Length : 5 Pre-Status: 0 On-time-Status: 8 Post-Status: 1
Phase Condition: On-time-Status
```



```
Accuracy: 94.11764705882352%
Length : 3 Pre-Status: 0 On-time-Status: 4 Post-Status: 1
Phase Condition: On-time-Status
```

By observing the result we can say that this is On-status because on-status is higher than other two. The accuracy is nearly 89.0%.

4.3.3.File 3:

Again we can manually say that file 3 has Post-Status tweet. And there is some comparison:

4.3.3.1. Pie chart

Actual

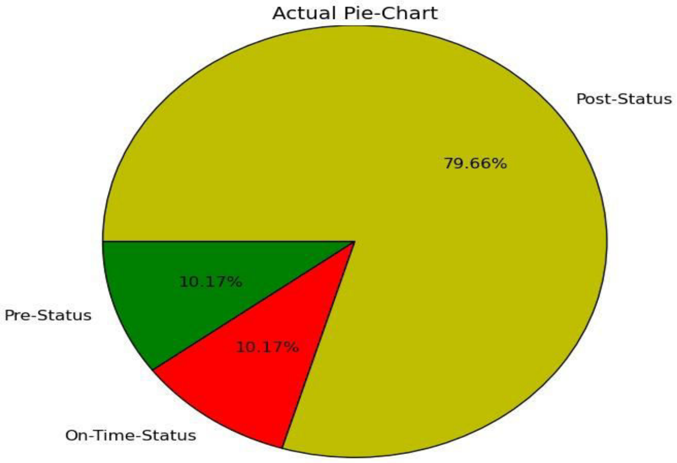


Figure 4.3.9: Pie-chart of file3 (Actual)

We visually represent the percentage ofthe actual value of phase3 using pie chart. From here we can see that the percentage of pre-status and the on-status phase is same (10.17), whereas the percentage in post-status phase is 79.66. As the post-status phase's percentage is higher than another two phases so we can say that this chart indicates that this data is under post-status phase. Though this is under post-status phase we also see a little bit pre-status and on-status phases because whenevercyclone has been finished there also some tweets which indicated that another disaster can be occur (pre-status) and some type of tweet like rain is still falling which is under on-status.

Predicted

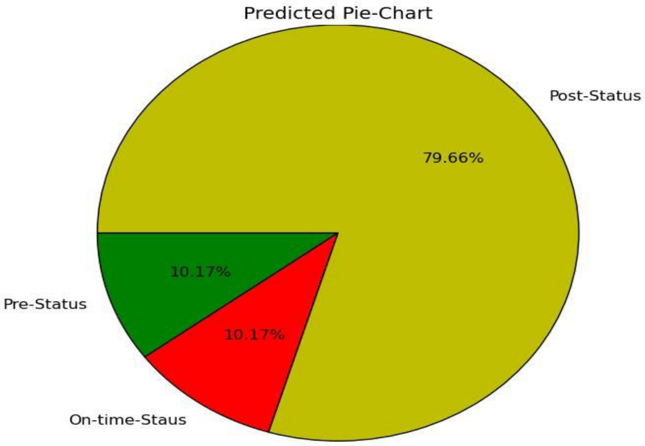


Figure 4.3.10: Pie-chart of file3 (Predicted)

This figure describes the visual representation of the predicted value of phase3 using pie-chart**.** The percentage of the three phases is pointing out in this figure. From this figure we can see that the percentage of predicted data in pre-status and on-status phase is same (10.17) as in actual, whereas in post-status phase the predicted percentage is 79.66. We could predict that in post-status phase the percentage is higher than the two and could predict that here is also pre-status phase and on-status phase. Here the percentage of the actual data and the predicted data is equal. After all of the observation we could say that we correctly predict that this is in post-status phase and our prediction is 100% correct as the actual and predicted value is same.

4.3.3.2. Axis chart:

Actual

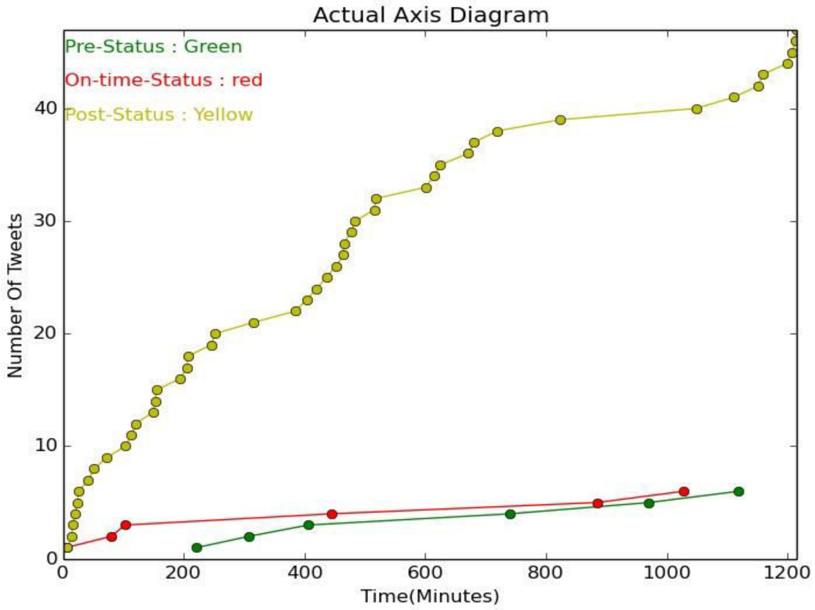


Figure 4.3.11: Axis diagram of file3 (Actual)

In this figure, the actual data of phase3 is graphically represented using axis diagram. Here we visualize three phases using three colors (green for pre, red for on, yellow for post). In this figure the x-axis represent the Time in minutes and the y-axis is the Number of Tweets. The circle point indicates the data of each phases including time represented by the color. From this diagram we can see that yellow circle point indicates the higher than red and green. As the yellow color represent the post-status phase, so by observing the data we can understand that this is in post-status phase.

Predicted

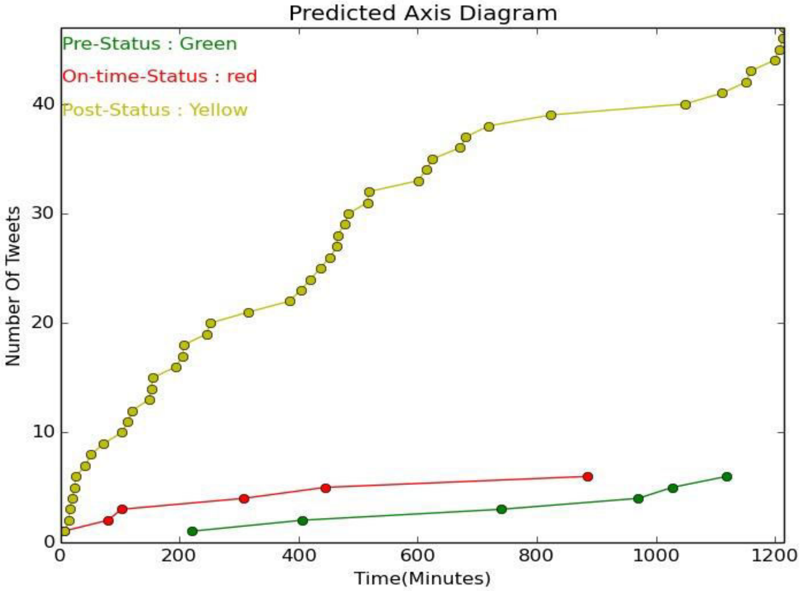
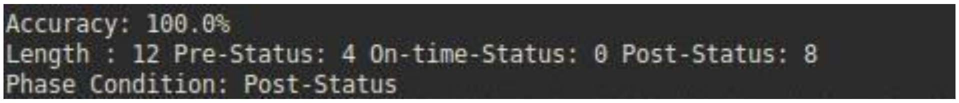


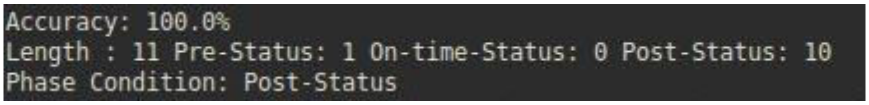
Figure 4.3.12: Axis diagram of file3 (Predicted)

In this figure, the predicted data of phase3 is graphically represented using axis diagram. Here we visualize three phases using same three colors (green for pre, red for on, yellow for post) as actual. This figure represents Time (minutes) vs. Number of Tweets. From this diagram we can say that we were able to correctly predict the disaster phases as a post-status phase because the data (represented by circle point) in post-status phase is higher than the 2 phases.

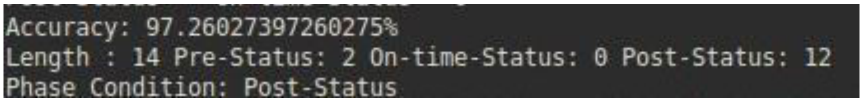
4.3.3.3. Accuracy and phase detection of file3:



```
Accuracy: 100.0%
Length : 12 Pre-Status: 4 On-time-Status: 0 Post-Status: 8
Phase Condition: Post-Status
```



```
Accuracy: 100.0%
Length : 11 Pre-Status: 1 On-time-Status: 0 Post-Status: 10
Phase Condition: Post-Status
```



```
Accuracy: 97.26027397260275%
Length : 14 Pre-Status: 2 On-time-Status: 0 Post-Status: 12
Phase Condition: Post-Status
```

By observing the result we can say that this is post-status because post-statusis higher than other two. The accuracy is nearly 96.0%.

4.3.4. File 4:

We can manually say that file 4 has all tweets of three states. There some comparison:

4.3.4.1. Pie chart:

Actual

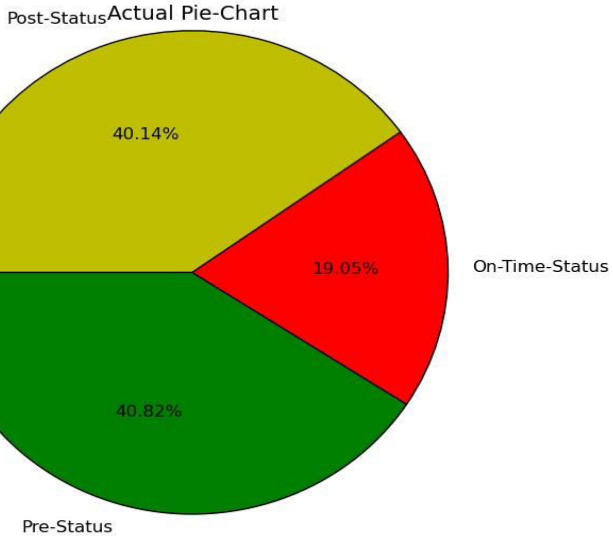


Figure 4.3.13 Pie-chart of file4 (Actual)

We visually represent the percentage ofthe actual value of all phases using pie chart. From here we can see that the percentage of pre-status is 40.82, the on-status phase is same 19.05, whereas the percentage in post-status phase is 40.14. Here the contribution of all phase's percentage is given. We can observe from the percentage that pre-status phase is higher though the pre-status and post-status percentage is almost same but the on-time status is lower than the two because during the disaster mostly people posted tweet rarely.

Predicted

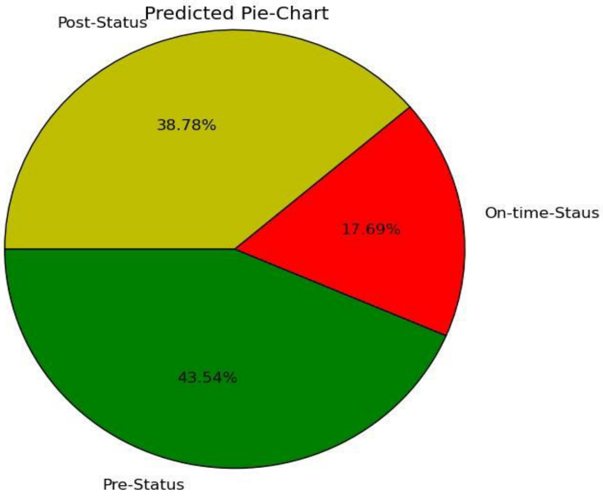


Figure4.3.14 Pie-chart of file4 (Predicted)

This figure describes the visual representation of the predicted value of all phases using pie-chart**.** The percentage of the three phases is pointing out in this figure. From this figure we can see that the percentage of predicted data in pre-status is 43.54, on-status phase is 17.69, whereas in post-status phase the predicted percentage is 38.78. There is a little bit difference between the actual and predicted value. So we could predict the three phases tweet almost accurately.

4.3.4.2. Axis-chart

Actual

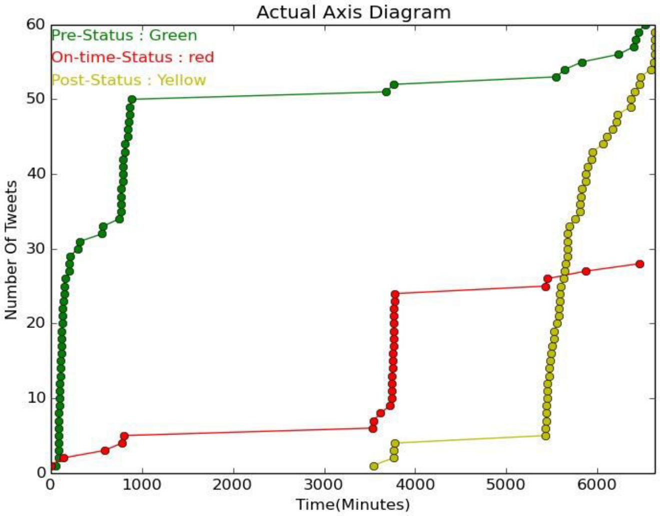


Figure 4.3.15: Axis diagram of file4 (Actual)

In this figure, the actual data of all phases is graphically represented using axis diagram. Here we visualize three phases using three colors (green for pre, red for on, yellow for post). In this figure the x-axis represent the Time in minutes and the y-axis is the Number of Tweets. The circle point indicates the data of each phases including time represented by the color. From this diagram we can see that green and yellow circle point are higher than red that is in on-time-status the tweet is lower.

Predicted

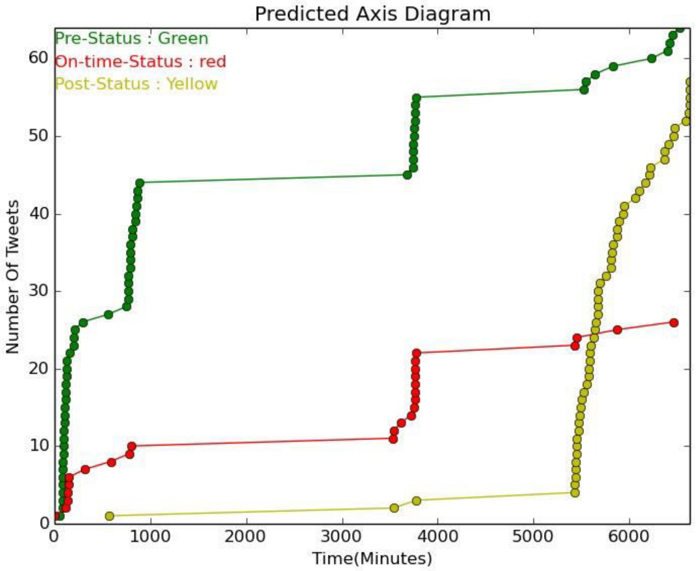
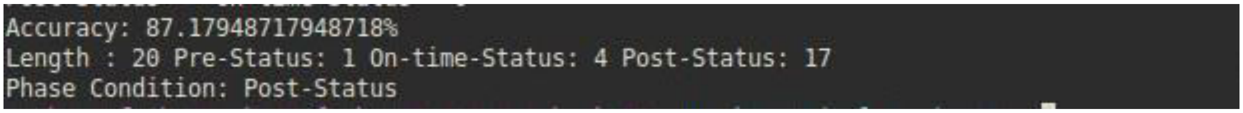


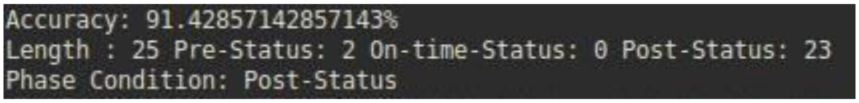
Figure 4.3.16: Axis diagram of file4 (Predicted)

In this figure, the predicted data of all phases is graphically represented using axis diagram. Here we visualize three phases using same three colors (green for pre, red for on, yellow for post) as actual. This figure represents Time (minutes) vs. Number of Tweets. From this diagram we can say that we were able to almost accurately predict the disaster phases. In our prediction red circle point is lower like as actual that is on-time status lower.

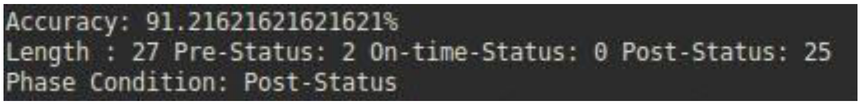
4.3.4.3. Accuracy



```
Accuracy: 87.17948717948718%
Length : 20 Pre-Status: 1 On-time-Status: 4 Post-Status: 17
Phase Condition: Post-Status
```



```
Accuracy: 91.42857142857143%
Length : 25 Pre-Status: 2 On-time-Status: 0 Post-Status: 23
Phase Condition: Post-Status
```



```
Accuracy: 91.21621621621621%
Length : 27 Pre-Status: 2 On-time-Status: 0 Post-Status: 25
Phase Condition: Post-Status
```

The accuracy result is: 90.0%

4.4. Earth Map

4.4.1.Earth map of file1

Actual

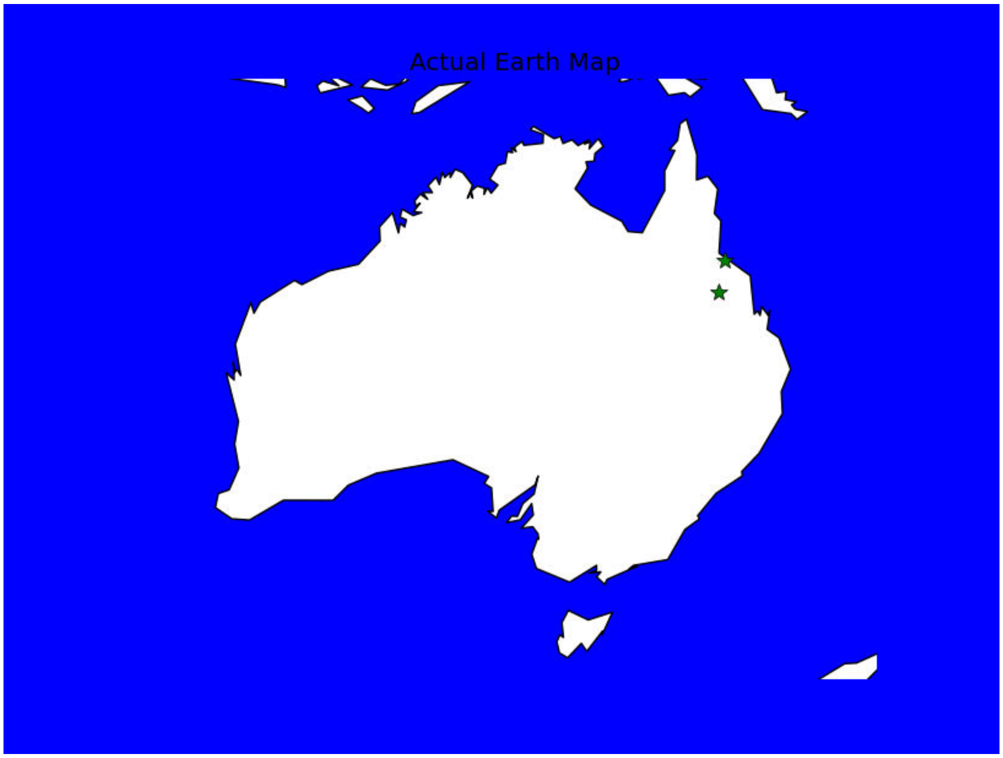


Figure 4.4.1: Earth Map of file1 (Actual).

From this figure we can see that two tiny green star pointing out the affected area. As this is green star and we represent the green color as a pre-status so we can say that this is under pre-status phase. Here we could point out only two points as a pre-status because during the disaster event people rarely open the GPS when they tweet. So we find only few tweets with GPS location.

Predicted

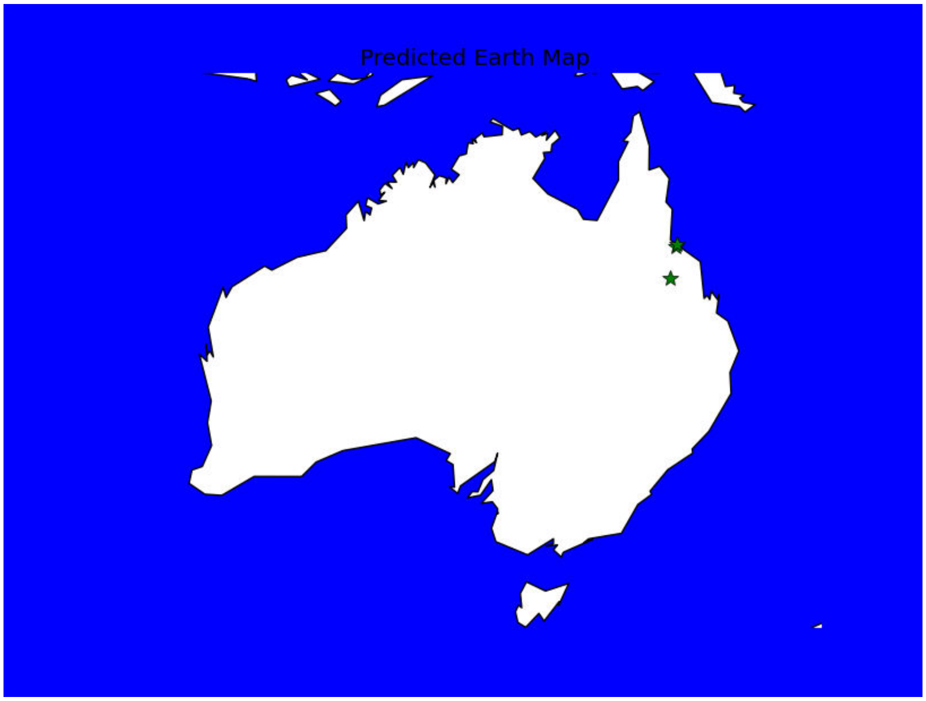


Figure 4.4.2: Earth Map of file1 (Predicted)

From this figure we can see that we were able to correctly predict two tiny green stars pointing out the affected area. As this is green star and we represent the green color as a pre-status so we can say that we could predict that this is under pre-status phase. Here we find only few tweets with GPS location.

4.4.2. Earth map of file2

Actual

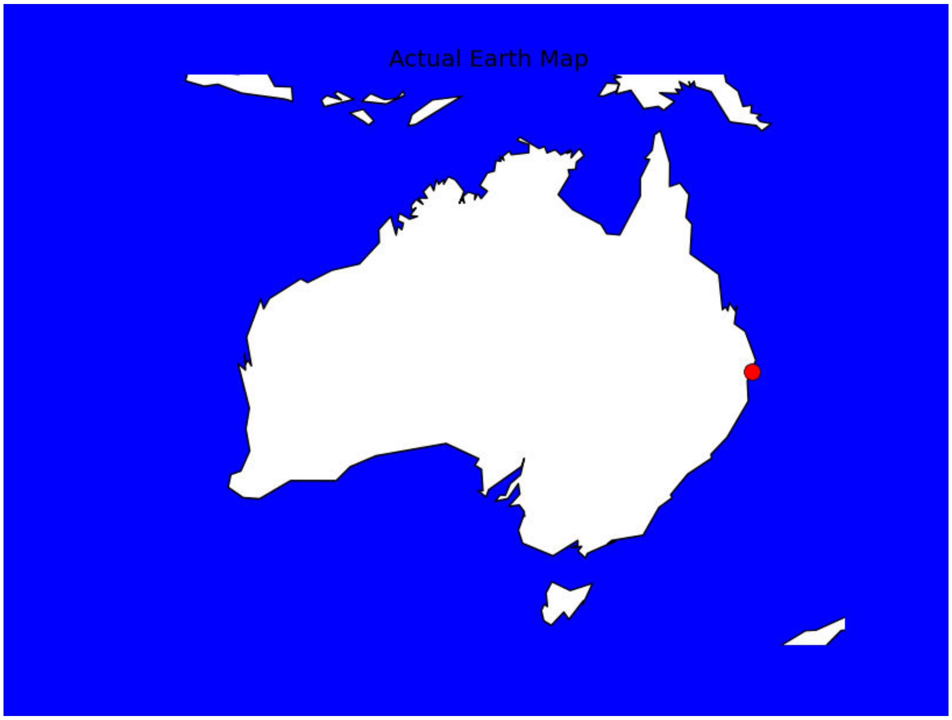


Figure 4.4.3: Earth Map of file2 (Actual)

From this figure we can observe that here is a tiny red circle pointing out the affected area. As the red color indicates the on-status phase so we can say that this is on-status phase. We could find only one area under on-status phase because finding the data with GPS is very tough because people generally does not open GPS whenever tweeting. Besides during disaster people tweet a little.

Predicted

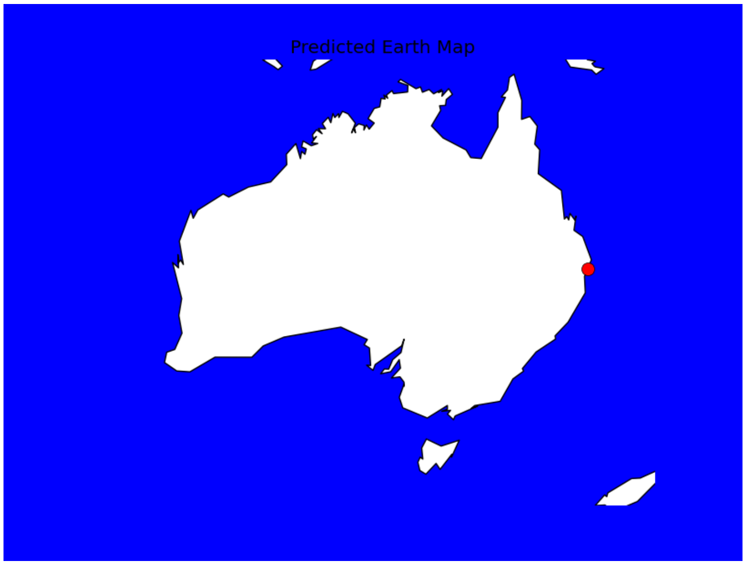


Figure 4.4.4: Earth Map of file2 (Predicted)

From this figure we can see that we were able to predict a tiny red circle pointing out the affected area (Queensland, Australia). As this is red circle and we represent the red color as on-status so we can say that we could predict that this is under on-status phase.

4.4.3. Earth map of file3

Actual

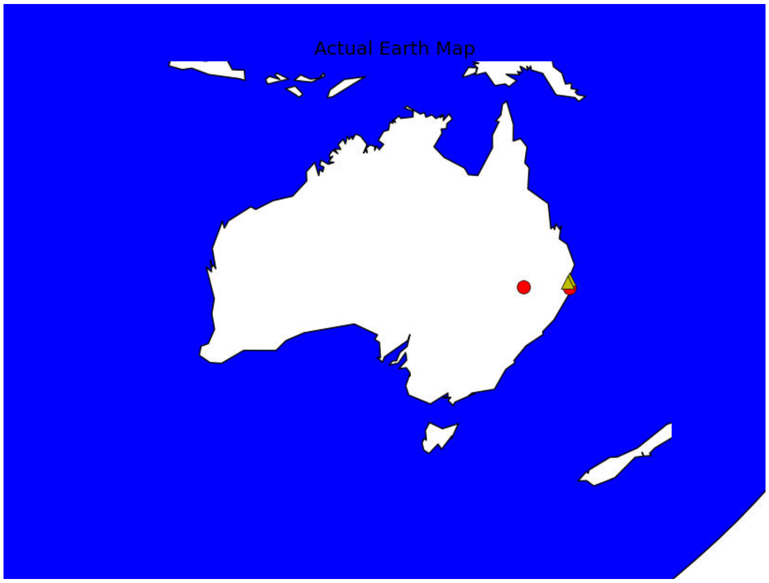


Figure 4.4.5: Earth Map of file3 (Actual)

From this figure we can observe that here is two tiny red circlesand two yellow triangles pointing out the affected area (Queensland, Australia). The red color indicates the on-status phase and the yellow color indicates the post-status phase. Here we can see both of the post-status and the on-status tweet.

Predicted

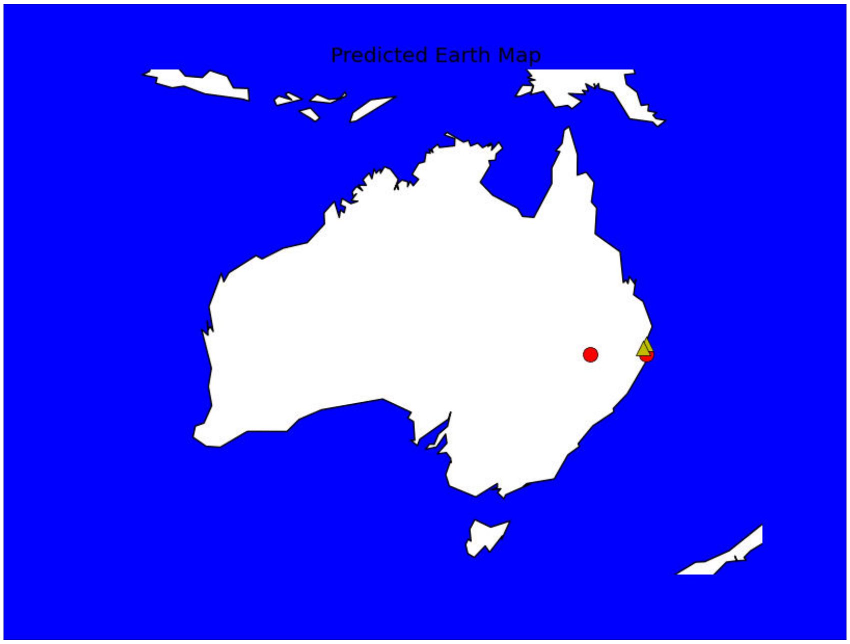


Figure 4.4.6: Earth Map of file3 (Predicted).

We could predict two tiny yellowtriangles (yellow for Post-Status) and twored circles(red for On-time-Status) pointing out the affected area (Queensland, Australia) under the Cyclone-Debbie

Chapter 5

5. Experimental Resultand Performance Evaluation

Here we calculate some comparison result in all four file.

5.1. Confusion matrix based on file1:

We take the accuracy and result of file1 for 5 time and show the average result in Confusion Matrix

Actual	**Predicted**			
		Pre-Status	**On-time-Status**	**Post-Status**
	Pre-Status	**43**	**5.2**	**0.0**
	On-time-Status	**0.0**	**4.4**	**0.0**
	Post-Status	**0.0**	**0.0**	**0.0**

Table 5.1: Confusion-Matrix of file1

In this confusion matrix, from the 48.2 actual pre-status tweets, the system can predicted that 43 were pre-status tweets and 5.2 were on-status tweets, and from the actual 4.4 on-status tweets, the system can predicted 4.4 on-status tweets correctly and here is no post-status tweets. As the number of pre-status tweets is higher than the on-status and post-status tweetsso we can say that this table of file 1 indicates the confusion matrix of pre-status tweets. The accuracy of file1 we get is**90.11%**.

The precision, recall and F1-measure of file 1:

Phases	**Precision**	**Recall**	**F1-measure**
Pre-Status	**1.000**	**0.892**	**0.942**
On-Time-Status	**0.458**	**1.000**	**0.628**
Post-Status	**Ambiguous**	**Ambiguous**	**Ambiguous**

Table 5.2: Some measurement of file1

From this table we can observe that, the precision, recall and F1-measure of pre-status tweets is high. We also get some measurement from on-status phases but the result of these three measures of post-status tweets is ambiguous that means we actually cannot understand that this status is under which phase.The precision of pre-status tweets is higher than two. As the measurement of pre-status tweets is higher than two so the file 1 indicates the measurement of pre-status tweets.

5.2. Confusion matrix based on file2:

We take the accuracy and result of file2 for 5 time and show the average result in Confusion Matrix

Actual	**Predicted**			
		Pre-Status	**On-time-Status**	**Post-Status**
	Pre-Status	**1.8**	**0.0**	**0.0**
	On-time-Status	**0.6**	**14.8**	**0.2**
	Post-Status	**0.2**	**0.0**	**6.2**

Table 5.3: Confusion-Matrix of file2

In this confusion matrix (Table 5.3), from the 15.6 actual on-time-status tweets, the system can predicted that 14.8 were on-time-status tweets, 0.6 were pre-status phases and 0.2 were post-status tweets, and from the actual 6.8 post-status tweets, the system can predicted 6.2post-status tweets. As the number of on-status tweets is higher than the pre-status and post-status tweetsso we can say that this table of file 2 indicates the confusion matrix of on-time-status tweets. The accuracy of file2 we get is **95.70%**.

The precision, recall and F1-measure of file 2:

Phases	**Precision**	**Recall**	**F1-measure**
Pre-Status	**0.692**	**1.000**	**0.817**
On-Time-Status	**1.000**	**0.949**	**0.973**
Post-Status	**0.968**	**0.968**	**0.968**

Table 5.4: Some measurement of file2

From theTable 5.4 we can observe that, the precision, recall and F1-measure of on-status tweets is high. We also get some measurement from pre-status tweets and post-status tweets. The precision of on-time-status tweets is higher than the two. As the measurement of on-status tweets is high so the file 2 indicates the measurement of on-time-status tweets.

5.3. Confusion matrix based on file3:

We take the accuracy and result of file3 for 5 time and show the average result in Confusion Matrix

	Predicted			
		Pre-Status	On-time-Status	Post-Status
Actual	Pre-Status	0.0	0.0	0.0
	On-time-Status	0.0	0.8	0.0
	Post-Status	5.4	2.8	49.2

Table 5.5: Confusion-Matrix of file3

In this confusion matrix (Table 5.5), from the 57.4 actual post-status tweets, the system can predicted that 49.2 were post-status tweets, 5.4 were pre-status tweets and 2.8 were on-time-status tweets, and from the actual 0.8 post-status tweets, the system can predicted 0.8post-status tweets. Here is no pre-status tweet. As the number of post-status tweets is higher than the pre-status and on-time-status tweetsso we can say that this table of file 3 indicates the confusion matrix of post-status tweets. The accuracy of file 3 we get is **85.90%.**

The precision, recall and F1-measure of file 3:

Phases	Precision	Recall	F1-measure
Pre-Status	0.000	Ambiguous	Ambiguous
On-Time-Status	0.222	1.0	0.360
Post-Status	1.000	0.857	0.922

Table 5.6: Some measurement of file3

From theTable 5.6 we can observe that, the precision, recall and F1-measure of post-status phase is high. We also get some measurement from on-status phases but the result of recall and f1-measures of pre-status tweet is ambiguous that means we actually cannot understand that this status is under which phase. The precision of post-status tweets is higher than two. As the measurement of post-status tweets is higher than two so the file 3 indicates the measurement of post-status tweets.

5.4. Confusion matrix based on file4:

We take the accuracy and result of file4 for 5 time and show the average result in Confusion Matrix

	Predicted			
		Pre-Status	**On-time-Status**	**Post-Status**
Actual	**Pre-Status**	**46.4**	**4.6**	**0.6**
	On-time-Status	**5.4**	**19.4**	**0.4**
	Post-Status	**4.0**	**0**	**55.6**

Table 5.7: Confusion-Matrix of file4

In this confusion matrix (Table 5.7), from the 51.6 actual pre-status tweets, the system can predicted that 46.4 were pre-status tweets, 4.6 were on-time-status tweets and 0.6 were post-status tweets, and from the actual 25.2on-time-status tweets, the system can predicted 19.4 on-time-status tweets, 5.4 were pre-status and 0.4 were post-status tweets. From 59.6 post-status tweets the system can predict 55.6 post-status and 4.0 pre-status tweets.The accuracy of file4 is**89.0%**.

The precision, recall and F1-measure of file 4:

Phases	**Precision**	**Recall**	**F1-measure**
Pre-Status	**0.8315**	**0.899**	**0.8639**
On-Time-Status	**0.8083**	**0.769**	**0.7882**
Post-Status	**0.9823**	**0.9328**	**0.9569**

Table 5.8: Some measurement of file4

From theTable 5.8we can observe that, the precision, recall and F1-measure of pre-status, on-time-status and post-status of tweets. The precision of post-status tweets is higher than two. As the table shows the measurement of all status tweets so the file 4 indicates the measurement of all-status.

By observing our experimental result we can say that our experiment performs well because it gives the higher accuracy and the result of precision, recall and F-measure is also good.

Chapter 6

6. Conclusion and future work

As social media messages are rich in content capturing people's activity, behavior, it is very easy to track public messages with their locations. Twitter is a popular social media and during natural disaster or any other crisis this source is very helpful for tracking the disaster data to alert people or helps the affected people. In our work we track this twitter data with latitude and longitude using text mining technique during disaster, Cyclone Debbie in Queensland, Australia (26th march to 31th march 2017, three steps). A machine learning classifier technique is introduced here to categorize these messages. First of all we cleaned the noisy data for extracting valuable information. Our aim was to measure and classify these tweets into three phases (pre, on, post) and detect the result in which phase is it now. We also visualize the actual and predicted value of each category phases using pie-chart and axis-chart and seeing this chart we can easily compare the actual data and the predicted data and the percentage of each phases and also visualize our experimental result by showing the earth map and pointing out these three phases on that map. We try to do our work as much as accurate we can and so in our result we see that there is increasing number of classification accuracy and we calculate this using confusion matrix. We also take the screenshot of the accuracy of each category phases and the all the phases to show our result how accurate it is.

In future work, it is useful for the people for alerting them before any disaster event occurs so that they can prepared to face the disaster. Besides the most damaged area of a disaster can be pointing out by using our work so that by seeing the pointed map affected people can get help, food, and donation by the response team. A technique should be applied so that disaster related data can automatically be classified into different phases. Our paper is also helpful for examining the disaster tweet with their geo-location by mapping. In future, more risky zone of a disaster can also be finding out so that people can alert and make a recovery plan or leave the risky area for reducing the damage. Also it can be possible to predict the nature and the danger level during natural disaster. It also possible to visualize the live disaster moment in future and will be very helpful for disaster responder. By visualizing the live disaster event it will be very easy to find out the area under the disaster event.

References

[1]. P. Meier, C. Castillo, M. Imran, S. M. Elbassuoni, and F. Diaz. Extracting information nuggets from Disaster-related messages in social media. In 10th International Conference on Information Systems for Crisis Response and Management, 2013.

[2]. Ashktorab, Z.; Brown, C.; Nandi, M.; Culotta, A. Tweedr: Mining twitter to inform disaster response. In Proceedings of the 11th International ISCRAM Conference, University Park, PA, USA, 18–21 May 2014.

[3]. S. Kumar, G. Barbier, M. A. Abbasi, and H. Liu. Tweet Tracker: an analysis tool for humanitarian and disaster relief. In ICWSM'11, 2011.

[4]. Geographic Situational Awareness: Mining Tweets for Disaster Preparedness, Emergency Response, Impact, and Recovery QunyingHuang , and Yu Xiao

[5]. Geotagging One Hundred Million Twitter Accounts with Total Variation Minimization
Ryan Compton, David Jurgens, David Allen

[6]. M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier. Practical extraction of disaster-relevant information from social media. In Proceedings of the 22nd internationalconference on World Wide Web companion, WWW '13 Companion, page 10211024, Republic and Canton of Geneva, Switzerland, 2013.

[7]. Neal, D.M. reconsidering the phases of disaster. Int. J. Mass Emerg. Disasters 1997, 15, 239–264.

[8]. www.csee.umbc.edu/~tinoosh/cmpe650/slides/K_Nearest_Neighbor_Algorithm.pdf
(siddharthdeokar CS8751)

[9]. http://faculty.smu.edu/tfomby/eco5385_eco6380/lecture/Confusion%20Matrix.pdf

[10]. Geolocation for Twitter: Timing Matters Mark Dredze, Miles Osborne, Prabhanjan Kambadur Bloomberg L.P. 731 Lexington Ave, New York, NY 10022.

[11]. Text-Based Twitter User Geolocation Prediction Bo Han, Paul Cook, Timothy Baldwin.

[12]. twitter4j.org/en/ (twitter 4j API for collecting data from twitter based on hashtag).

[13]. http://matplotlib.org/index.html (visualization platform)

[14]. https://en.wikipedia.org/wiki/Ternary_search

[15]. Geolocation Prediction in Twitter Using Social Networks:A Critical Analysis and Review of Current Practice David Jurgens, Tyler Finnethy, James McCorriston, Yi TianXu, Derek Ruths.

[16]. Twitter Mining for Disaster Response: A Domain Adaptation ApproachHongmin Li Nicolais Guevara, Nic Herndon DoinaCaragea, Kishore Neppalli Cornelia Caragea,AnnaSquicciarini Andrea H. Tapia.

[17]. DisasterMapper: A CyberGIS framework for disaster management using social media data Qunying Huang1 , Guido Cervone2 , Duangyang Jing1 , Chaoyi Chang1

[18]. Confusion Matrix-based Feature Selection, Sofia Visa, Brian Ramsay, AncaRalescu, Esther van der Knaap.