Spring 5-9-2015

# Real-Time Social Network Data Mining For Predicting The Path For A Disaster

Saloni Jain
*Georgia State University*

# REAL-TIME SOCIAL NETWORK DATA MINING FOR PREDICTING THE PATH FOR A DISASTER

by

SALONI JAIN

Under the Direction of Yanqing Zhang, PhD

## ABSTRACT

Traditional communication channels like news channels are not able to provide spontaneous information about disasters unlike social networks namely, Twitter. The present research work proposes a framework by mining real-time disaster data from Twitter to predict the path a disaster like a tornado will take. The users of Twitter act as the sensors which provide useful information about the disaster by posting first-hand experience, warnings or location of a disaster. The steps involved in the framework are – data collection, data preprocessing, geo-locating the tweets, data filtering and extrapolation of the disaster curve for prediction of susceptible locations. The framework is validated by analyzing the past events. This framework has the potential to be developed into a full-fledged system to predict and warn people about disasters. The warnings can be sent to news channels or broadcasted for pro-active action.

INDEX WORDS: Data mining, Disaster computing, Real-time disaster prediction, Early warning system, Prediction path, Regression

# REAL-TIME SOCIAL NETWORK DATA MINING FOR PREDICTING THE PATH FOR A DISASTER

by

SALONI JAIN

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2015

# REAL-TIME SOCIAL NETWORK DATA MINING FOR PREDICTING THE PATH FOR A DISASTER

by

SALONI JAIN

Committee Chair: Yanqing Zhang

Committee: Zhipeng Cai

Yingshu Li

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2015

## DEDICATION

I would like to dedicate this work to my late grandfathers, Jagjot Singh Jain and D. D. Jain. I would also like to dedicate this work to my mother, Dr. Rajni Jain, who has been my biggest inspiration of all time.

## ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Dr. Yanqing Zhang for without him this thesis work would not have been possible. I would like to take this opportunity to thank Dr. Zhang for his guidance, patience and advice throughout my thesis. Without his motivation and moral support, comments and suggestions this study would not have been possible at all. This thesis is entirely due to his interest.

I would also like to thank Dr. Zhipeng Cai who got me interested in the research field of Machine Learning. I would also like to thank Dr. Yingshu Li and Dr. Cai for taking out valuable time from their busy schedules to review my thesis and provide insightful advices and suggestions.

I am extremely thankful to my undergraduate teachers, Dr. Richa Singh and Dr. Mayank Vatsa for introducing me to the field of Pattern Recognition and Machine Learning. I would also extend my gratitude to all my teachers who inspired me on every step of the way.

I wish to thank my mother, Dr. Rajni Jain who supported me at every single step of the way. I cannot express my sincere thanks to her through words. She has given me countless advice and reviewed my thesis work with me. She provided unfaltering support and encouraged me when I felt despair. I would like to thank my father, Mr. Bhushan Kumar Jain, for believing in me and allowing me to continue higher studies. I know it must've been hard for him to send his daughter to a foreign land.

I would also like to thank my sister Saumya, my cousins Anshika and Rishabh, my friends Mannika, Sneha, Apaar, Gaurav, my family and friends who have provided me with their blessings and support. This was clearly a joint effort of so many people. All mistakes that remain are, of course, my errors. Thanks!

(Saloni Jain)

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# 1 INTRODUCTION

Social Media has become a very important tool to stay in touch with friends, to market products and services offered by companies and even to make announcements by government agencies and news channels. One of the social networking sites which has gained vast popularity is Twitter. This research work deals with the data obtained from Twitter which is mined for getting useful information for a real-world scenario, mainly, disaster path prediction. It is further discussed in the next section.

## 1.1 Twitter and its Importance

Twitter is an online social network (OSN) used by millions of people all over the world. It enables people to stay connected with their friends, family and colleagues. With advancement in technology, it has become easier to access Twitter using mobile devices like iPhones and iPads. It enables its users to post messages which are 140 characters or less which are called *tweets*. Users can also *retweet* messages, which is posting the message posted by other users. This can be thought of as email forwarding. These tweets can be displayed to all users or only to the people following the user. A user can follow other users but it is not necessary for the user who is being followed to follow back. This makes the links in Twitter directed. Currently, Twitter has 288 million monthly active users with an average of 500 million tweets being sent per day [1].

Twitter has become an important resource for the field of Data Mining because of its many features. It has a varied variety of users which can represent a sample of the entire population. The revolution of information and communication technology (ICT) has made it possible for billions of people to access social networking sites ensuring that they have a wide reach of people. They can post messages on the go which ensures that the real-time nature of the messages. Compared to emails, this “push” of information is almost instantaneous. Twitter also

has a feature of searching or filtering messages which are interesting to a user using *hashtags*. Many YouTubers create hashtags to communicate with their fans and answer their questions, e.g. #AskSuperwomanLive. Users also have the freedom to follow or join groups that they like. It also caters for security for its users, where they can decide to post tweets publicly or privately. If they decide their tweets to be public, then they can be viewed by anybody in the Twitter network. However, if they are private, then only the people in the user's network can view them. Mostly, people post about their trivial personal experiences but sometimes they post messages which can contain information which will be valuable on mining. This information can be about events like politics, traffic jams, riots, fires, earthquakes, storms, etc. Therefore, Twitter can also act as a non-traditional medium to obtain news as people can tweet information which is newsworthy. They can even create messages which can potentially be news which can be used in early warning detection systems. However, the most important feature for this study is the real-time nature of the information dissipation in the Twitter network. It further becomes useful when 80 per cent of the users are mobile users [1] which can provide us with exact geo-location and more up-to-date information. These users may post several times a day contrary to bloggers who post once every few days. In 2011, when a tsunami hit Japan, Twitter was used as the means of communication when traditional modes of communication went down [2].

### 1.2 Data Mining for Disaster Management

Data Mining plays a crucial role in extracting useful information from Social Media. The reason is because it also contains personal trivial data which is not very enlightening or useful to a large group of people. It is used in many areas for analysis. Companies and organizations can perform sentiment analysis for their products and services [3, 4]. It can also help in detecting and predicting disasters [5] and events such as influenza [6]. This can be the basis of forming early

warning systems, one of which was proposed by Avvenuti et. al [7]. The subsequent section talks about previous works in disaster management using Twitter.

### *1.2.1 Related Work*

Sakaki et al. have mined Twitter data for real-time earthquake detection [5]. They created an application for earthquake reporting system in Japan. This system consists of two parts – event detection and the probabilistic spatiotemporal model of the event. The detection is performed by making a classifier using a Support Vector Machine. The features used are – keywords in a tweet, the number of words in the tweet and the context of target-event words. For creating a probabilistic spatiotemporal model, the authors assumed that the users are social sensors and their tweets are sensory information. This information is noisy because the users will not always tweet about the event. Some sensors can be very active and others might not be. Just like using physical sensors, these social sensors can be used for Kalman filtering and particle filtering. These are used for estimating the location in ubiquitous computing. They were able to detect 96 per cent of earthquakes reported by Japan Meteorological Agency.

Avvenuti et al proposed a novel architecture for an early warning system and validated it with an implementation in [7]. They made use of social sensing where a group of people or a community provides similar information that might be obtained from a single sensor. The authors dealt with the issue of earthquake detection in Italy. The main steps involved in their study are – Data Acquisition, Data filtering, Event Detection, Damage Assessment and Early Warning. The keywords selected were Italian words for “earthquake” and “tremor”. They used Streaming API of Twitter for up-to-date tweets. The filtering phase reduces noise by discarding retweets, replies, tweets containing blacklisted words and tweets by official channels. A more sophisticated filtering was done by classifying tweets as useful and not useful. The features used

are: URL, mentions, words, character, punctuation and slang/offensive words. Events are detected by temporal and spatial analysis. For temporal analysis, they created a novel burst-detection method which observes a peak of the number of messages in a time window. They extracted location from the content of the tweet for spatial analysis using TagMe [8]. Damage Assessment was done by using a bigger set of general keywords, images and videos. The results obtained from the experiment were checked with official data to show that earthquakes with a magnitude equal or greater than 3.5 on Richter scale can be timely detected with 10 per cent False Positives.

There is a need for an automated disaster management system which can recommend suitable action patterns in case of a disaster. These can deal with informing about the shelter in case of a disaster emergency or maybe how to travel from one place to another. One of the methods was suggested by Nguyen et al. [9]. They built an earthquake semantic network using human activity on Twitter based on Web Ontology Language. A Twitter activity was defined by five attributes, namely – action, object, location, time and actor. The network is connected by the relationships – *Next* and *BecauseOf*. They also created automatic data for the network. This network was further used to recommend suitable actions in the face of a disaster. They found out that their learning model Conditional Random Field (CRF) outperformed the baseline method (which used syntactic parser with the linguistic pattern for training data) and the previous extraction method [10]. One of the problems with the works of Banerjee et al. [10] was that the list of actions and objects had to be prepared before extraction.

Another focus for disaster management is to make sure that information is spread as widely as possible. Social media is the fastest way of information diffusion where general population as well as government agencies can respond to requests for assistance, information and

announcements. Retweeting on Twitter is the most efficient way to spread an original message beyond the author's network. Zhu et al. built a predictive model for finding the retweeting decision of a user [11]. They have found out the factors affecting the retweet decision. The features can be classified into three categories – contextual influence, network influence and time influence from which a set of features are found. A Monte-Carlo simulation was also performed for finding how the information propagates in Twitter network. Even though the information on social media is important for spreading awareness, credibility of the information might be a problem. Kongthon et al. analyzed the content of Twitter messages and the characteristics of Twitter users which can be used for better disaster management [12]. They used keyword analysis and rule based approach for classifying tweets into five categories – Situational announcements and alerts, support announcements, requests for assistance, requests for information and other. They also classified users on the basis of number of followers and retweets. It is of utmost importance to make sure that the information being used for disaster preparedness and response is current and true. Hence, all the factors must be considered.

Disaster Management is also useful for topical analysis. In the aftermath of the 2011 Japan tsunami, there was a commotion due to damage to conventional communication networks and power outages. Family members wanted to confirm safety of each other. There was a need to exchange demands and opinions. In [13], Murakami et al. discussed text mining techniques on tweets. They are used to find areas where there are a shortage of supplies and other peoples' needs. Two approaches are used; one that uses simple keyword matches, and another that uses syntactic pattern dictionaries. In the keyword match approach, the system found areas needing supplies, but there was noise in some of the tweets found in which the tweet contained a keyword but was unrelated to the disaster. In the syntactic approach, a syntactic pattern dictionary was

used to identify things that were in short supply. For example, “cannot buy <noun>,” and “<noun> is sold out” are patterns that were used. Using this approach, the most frequent nouns that had shortages were water, battery, rice, gasoline, and toilet paper.

### 1.3 Real-Time Data Mining for Predicting the Path for a Disaster in Twitter

The present work has aimed to take another step forward in the direction of disaster management on Twitter. Disaster detection is not enough. After the disaster has been detected, we should be able to predict the real-time trajectory of the disaster (like the path a tornado or a fire) can take. This can enable officials to spread warnings to susceptible people and take preventive measures. This framework can be further extended to detect time-sensitive requirements such as food, shelter, medicines, etcetera for the victims based on some keywords used in the tweets. The main steps in this work are as follows – (1) Extracting tweets and user information from Twitter using suitable keywords, (2) data preprocessing of the tweets, (3) extracting locations, geo-coordinates and time from a tweet, (4) clustering of locations to further filter the data points, (5) finding a polynomial curve to draw an approximate trajectory, (6) extrapolating the trajectory to get a path and finally (7) validating the extrapolated path by analyzing it against real data points.

# 2 EXPERIMENT

This chapter presents the details of the experiment to trace the trajectory of a disaster and use it for issuing the forewarning to the susceptible people. The main steps followed are – Data acquisition, Data preprocessing, Extracting information from tweets, Clustering data points for filtering, Curve fitting to get the trajectory for the disaster, Extrapolation of the graph, Validation of the extrapolated graph. The details of the experiment are explained further below.

## 2.1 Event – Alabama Tornado October 2014 [14]

Alabama was hit by a series of small tornadoes on 13th October 2014. The counties that were directly hit by the tornadoes are Colbert, Lamar and Marion. Some of the states where tornado or strong winds were reported are Alabama, Louisiana, Florida, Georgia, Tennessee, Arkansas, Missouri and Illinois. More information is provided in the appendix about the tornadoes which occurred on 13th October 2014 in Alabama. Tornado reports are available on NOAA's National Weather Service Storm Prediction Center [15].

## 2.2 Data Acquisition

The role of collecting data is extremely important for any experiment as all further operations are performed on the data obtained in this step. The data should be able to represent all the information that is required for the cause, which in proposed experiment, is the prediction of the path of the tornado that hit Alabama in October 2014. Therefore, it becomes crucial to not lose any key data to prevent loss of information. Any errors that creep in at this step, will propagate throughout the system in all sequential steps. There are two phases in this step – identifying an apt set of keywords and collecting data from Twitter using an appropriate interface.

### *2.2.1 Keyword Selection*

Twitter produces around 6000 tweets per second [16], thus increasing the probability of noise in the dataset. To have the relevant data for the experiment, the keywords should be selected carefully. They should neither be very specific nor too general. If the keywords are very specific to a certain event, it might limit the tweets which are extracted. For example - #*AlabamaOctoberTornado* is very specific for our event. Not everybody will tweet using these keywords and hence informational tweets might not be captured by our system. On the other hand, using very general keywords like, *#StrongWinds* will lead to too much noise in the collected dataset as this keyword does not necessarily mean a tornado or a storm. Users can just be talking about a weather change in the day. Therefore, it becomes extremely crucial to choose the correct set of keywords. The next phase is 'Data Collection which is discussed in the next sub-section.

### *2.2.2 Data Collection from Twitter*

Twitter has provided developers with two APIs – Twitter Search API and Twitter Streaming API. As the name signifies, the Search API allows users to query against the indices of recent or popular tweets [17]. However the search API does not get all the tweets and only the relevant ones. This might result in a loss of some tweets from the results. Generally the tweets from the past week are extracted. It has a rate limit of 180 requests/query per 15 minutes. For more information, the documentation provided by Twitter can be referred [17]. In contrast to the Search API, the Streaming API can provide the user with all tweets for maintaining the completeness of the dataset [18]. It needs a persistent connection open for streaming. The main benefit of this API is getting the real-time stream of tweets. However, this API cannot obtain the tweets that were published before opening the connection. For my experiment, I chose the Search

API because I wanted past tweets as the disaster had already occurred. However, the limitations of the Search API was not a problem for the system because a sufficiently large dataset was obtained.

The system made use of Twitter Search API through the interface developed by Martin Hanksey called Twitter Archiving Google Spreadsheet TAGS v5.1 [19]. It requires authentication which has been mandated by Twitter for all its APIs. However, since it uses the Search API, there are some limitations. It can over-represent the more influential users which might lead to some bias in the data. Also, the API can access only a subset of all the tweets but we obtained a large number for performing the experiments. Streaming API would have given a more complete dataset which would have given a more accurate result because the tweets obtained would be in real-time. However, I wanted to build a framework, for which, the Search API suited just fine. Also, as mentioned before, the event had already occurred rendering the Streaming API of not much use for the system. It will be better for implementation for real-world applications.

### 2.3 Data Preprocessing

Data collection is done by querying keywords mentioned in the previous section which gave a large selection of tweets. A set of more than 4000 tweets was used to run the experiment. This set can obviously have tweets which won't be useful for this study. It is important to eliminate those to save computation power and prevent noise in the data. The dataset should only contain English tweets as the event in consideration, i.e. the Alabama Tornado in October 2014, occurs in United States. Even if people from around the world tweeted about it, we will want to eliminate those if they are in another language because they will not provide us with much

information. However, my system is just an initial framework, which can be extended to multiple languages.

The first step was to remove all non-English tweets. This was done by checking the language feature of a tweet. If it was not '*en*' it was discarded. Along with getting the language of the tweet, we also obtained the geo-coordinates for those tweets since the system already made a call to the Twitter API. Location extraction is discussed in the next section. The next step was to remove spam tweets. A real-world application can have the full spam detection feature in place. A list of 165 spam words was compiled using numerous blacklists. The complete set of the spam words is presented in Appendix A. The tweets which contained spam words were removed. The final dataset was stored as a *json* object. The fields stored are mentioned in Table 2.1.

**Table 2.1 Tweet Features of the database and their Meaning**

| Field Name | Meaning |
|---|---|
| tweet_id | A unique number for the tweet |
| iso_language_code | The language of the tweet. English is represented by 'en' |
| user_id | A unique number for the user who posted the tweet |
| text | The content of the tweet |
| created_at | UTC time when the tweet was created |
| time | Local time when the tweet was created |
| geo-coordinates | Latitude and Longitude from where the tweet was published |
| entities | Contains details about hashtags, URLs and symbols |

The tweets might be retweets and messages to certain users but I decided to keep these tweets as they give weight to the tweets. For example – consider a tweet, '*I hope people of Alabama are able to withstand the storm*'. Now if this tweet is retweeted, we can get more weight for the same tweet. This implies that Alabama is definitely a location where the storm has hit. The method to extract location and time is given in the next section.

## 2.4 Time and Location Extraction

For predicting the path of a disaster, it is safe to assume that the disaster has at least already started. Even so, for creating a complete application for disaster management, there are many state-of-the-art solutions which can detect an event [7, 5]. Once occurrence of the event is made certain, the possible path of the disaster needs to be traced from the dataset. This can be done using the extraction of name and location from the tweets. These two are the most important fields for the experiment of predicting the path. The location field can determine the *where* aspect and time field can determine the *when* aspect of a disaster. Thus, it is important to do temporal and spatial analysis which mainly deal with getting the time and the most accurate locations from a tweet.

### *2.4.1 Time Extraction*

It is fairly straight-forward to get the time the tweet was created. As seen in Table 2.1, the system has two types of times – created-at and the local time of the tweet. For the experiment, I decided to use the local time at which the tweet was created for understanding the association between the actual event and the predictions by the system. The UTC time can just as well be used. In fact, it will be much more useful if the event being considered is a global event. In this work, local time is selected for further experiment because the event under consideration was a local event.

### *2.4.2 Location Extraction*

Twitter provides its users with the option to geo-code their tweets which can offer the exact locations, especially in mobile devices. But, less than 1% of the data that was collected had geo-locations associated with them. This leaves a very small number of tweets to work with. Sakaki et al. proposed a workaround to this solution. He proposed that the location of the Twitter

account can be used to get an approximate location [5]. However, this is just an approximate location. It is possible that the user may not even be in the same state he was when he first created his Twitter account. The best way to get the most information from a tweet about its location is through the content of the tweet [7] like in example, "*Widespread severe weather outbreak…right now Tornado Warnings in Illinois, Tennessee, Kentucky, Louisiana, Alabama & Mississippi"*. If the example can be properly used, we can obtain locations Illinois, Tennessee, Kentucky, Louisiana, Alabama and Mississippi. In the present work, the TagMe API developed at University of Pisa, Italy to get the "spots" which can potentially be location names [8]. The TagMe API can annotate very fast to give accurate results. It also has a parameter which can take care whether the text to be annotated is a tweet or not as tweets have special features like URLs and hashtags. It returns a *json* object with all the spots from the tweets. It has a very descriptive documentation which can be referenced for more details.

For the proposed experiment, location is very important. Hence, location was extracted from the three sources for a single tweet – (1) geo-coordinates, (2) the content of the tweet and finally (3) the location of the Twitter account. If a tweet has geo-coordinates associated with it, they were also extracted while checking if the tweet is in English during the preprocessing step. This was done to save computation power as there is a rate limit on the number of calls to Twitter API. If a tweet has geo-coordinates associated to it, they will be the most accurate location for the tweet. The next best method for extracting location is through the content of the tweet.

The text of the tweet is passed to a Python library called *geograpy.* It can extract countries, regions and cities from a URL or text. Along with the text, the *spots* which were extracted using the TagMe API are also passed to identify if they represent a location. For example, the tweet can be "*GA under Tornado warning*". The TagMe API will result *GA* as a spot with the

annotation as *Georgia.* This spot is then passed to *geography* which will identify it as a location. The list of locations is then passed to another Python library called *geopy.* It can locate the coordinates of places and therefore if the spots from the list are identified as locations, it can geocode them to return a pair of coordinates, i.e. the latitude and the longitude of the location. Similarly, *geopy* library can extract the coordinates for the location associated to the Twitter account. For all the geo-coordinates, a location was also saved by reverse geocoding for keeping a track of the places. The next step is to filter out noise and keep only the samples which will be useful for getting the curve for trajectory prediction.

## 2.5 Clustering the Data for Filtering

It is important to find the useful points that will lead us to the solution. Clustering can help us achieve that. It will group nearby locations in the same clusters. The clusters which lie far away from other clusters can be eliminated reducing the noise and leading to a better polynomial curve. Clustering of areas has many other potential uses like identification of areas which need help and what help, identification of areas which need to be evacuated depending upon the intensity of the disaster, etcetera. In this study, the clustering algorithms mainly serve the purpose of removing noisy data points which might hinder further mining of the data. Two clustering algorithms were tested – Agglomerative Clustering and Density-Based Spatial Clustering of Applications with Noise (DBSCAN).

### *2.5.1 Agglomerative Clustering*

Agglomerative clustering uses a bottom up hierarchical clustering where each point is initially its own cluster. Clusters are merged successively until the desired number of clusters are achieved. The merging can be done based on three linkages – ward, complete and average. The experiment observes the results with all three and as expected, ward linkage gives the most

distinctive clusters. Another parameter is connectivity which merges clusters only if they are adjacent to each other. Number of clusters, *n* is user defined. For experimental purposes, values of n was varied between 3 and 30. It was observed that 3 will be the most optimum number of clusters. However, agglomerative clustering has no concept of core samples and non-core samples which is why DBSCAN clustering is beneficial. Identification of core samples can help us eliminate unwanted points and therefore DBSCAN was considered for the rest of the experiment.

### *2.5.2 DBSCAN Clustering*

In DBSCAN clustering, the clusters are based on the density of data points. They are areas of high density which are separated by areas of low density. It will have a set of core samples and a set of non-core samples. Core samples are those points which have at least a given number of minimum samples, *min_samples* within a specified distance, *eps*. Non-core samples are close to the core samples but they do not meet the condition for core-samples. For ensuring high density, the values for *eps* should be low and *min_samples* should be high. This algorithms is beneficial for the experiment as it can eliminate outliers and also perform sampling in the process. Clusters which have lesser number of data points as compared to others should be removed. Also, data points which are not part of any clusters are removed as noise.

## 2.6 Curve Fitting and Extrapolation for the Trajectory of the Disaster

Latitude and longitude of the locations which were struck by disaster or will possibly be hit by disaster can be visualized as functions of time as the disaster progresses. They are represented as scatter plots in Figure 2.1 and Figure 2.2.

In the proposed work, experiment was performed with two methods. Results from all the experiments suggest a similar curve, confirming the validity of the curve. The aim was to

find a polynomial curve which will satisfy the existing points and also give a good prediction for the future points based on obtained values. The first approach was to use least squares polynomial fit. This was accomplished by using the *polyfit* function of the python library, *numpy* and *cftool* in MATLAB. In the experiment, it was assumed that latitude and longitude are functions of time as the disaster progresses. For the simplicity of the equation, latitude and longitude were taken as independent of each other. Therefore, two polynomial functions of time are needed, one for latitude and one for longitude. It was experimented with linear regression and non-linear regression with degrees two, three and four. Higher degrees were also tried but they did not change the curve significantly. The points were also assigned weights depending upon its frequency. But that did not bring much changes and therefore, the results were found without weights.

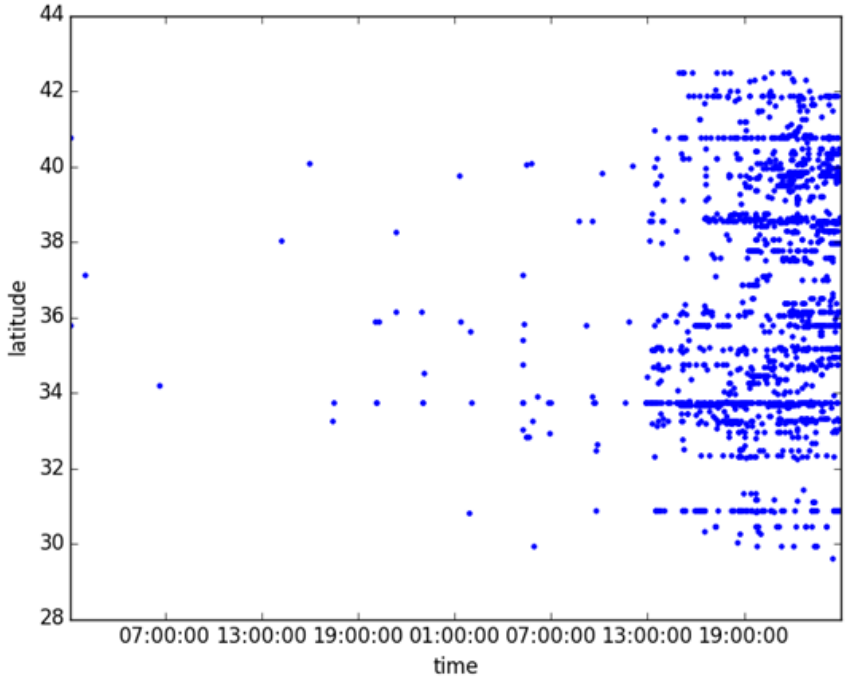


**Figure 2.1 Latitude vs. Time as disaster progresses on October 13th 2014.**

The curve was first found for the data for dates 12th and 13th October 2014 separately for latitude and longitude. For finding the future locations that might be susceptible, the curve was extrapolated for future time values. A confidence level of 95% was kept and prediction bounds were also found. Validation was done by plotting the actual points of the future date, in my case 14th October 2014 and plotting a curve using that data. The plots are discussed in the results section. Using the extrapolated curve, future value pairs are obtained which are then reverse-geocoded to obtain the actual location addresses which will tell us the places which should be warned.

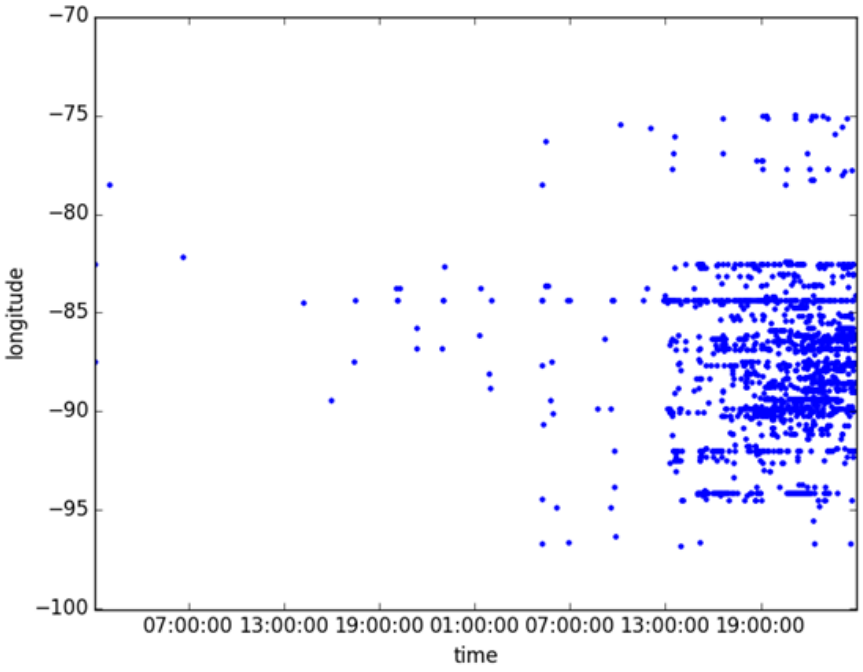


**Figure 2.2 Longitude vs. Time as disaster progresses on October 13th 2014.**

## 2.7 Conclusion

A framework for predicting the trajectory of a disaster has been discussed in this chapter. It consists of acquiring data, preprocessing it, extracting time and geo-location from the tweets, clustering using DBSCAN for filtering and finally curve formation and its extrapolation for

finding the susceptible locations with 95% confidence. The results are further discussed in the next chapter.

# 3 RESULTS

The location was extracted in three ways to give an accurate picture of the real-world scenario – from the geo-coordinate of the tweet, from locations mentioned in the content of the tweet, from location/geo-coordinate of the user. Times were also extracted to form a location-time tuple. These locations have to be clustered together to remove noise or outliers. A curve is formed using the obtained points to give an approximate path of the disaster. Further, we can extrapolate the path to get the susceptible locations. This can give the officials an idea for taking preventive measures and get equipped for the disaster. The sub-sections discuss the results obtained at each step of the experiment.

## 3.1 Data Acquisition

As mentioned in Section 2.2, keyword selection is very important to capture all the relevant data. For this study, I selected three keywords – '*Tornado*', '*AlabamaTornado*' and '*storm'*. The type of event can generally serve as good keywords since they'll capture more data and prevent the loss of relevant data. For my study, the training data was taken for the dates 12$^{th}$ and 13$^{th}$ October 2014. A total of 4147 tweets were extracted for these days. The fields that were extracted are – *id_str, from_user, text, created_at, time, geo-coordinates, iso_language_code, to_user_id_str, from_user_id, source, profile_image_url, status_url* and *entities_str*. Similarly, future data which acted as the validation data was taken for the date 14$^{th}$ October 2014. A total of 2920 tweets were collected with the same fields. Tweets from 15$^{th}$ to 17$^{th}$ October 2014 were also collected but data till 14$^{th}$ October was observed as sufficiently enough for the experiment to prove the validity of the framework.

## 3.2 Data Preprocessing

The data was processed for getting only the English tweets and removing the spam tweets. The training data resulted in 3909 tweets which was a big enough number to work with. On the other hand, the validation data, that is, the data for 14th October 2014 resulted in 2390 tweets. The next step is to extract time and location from both the datasets.

## 3.3 Time and Location Extraction

Using the locations and time obtained, a tuple is created containing the name of the location for readability, geo-coordinates for the exact point on earth, and local time. The extracted points are displayed on a world map in Figure 3.1. Figure 3.2 displays the map of the reports that were released by SPC for October 13, 2014 [15]. We can see that the actual affected points are also present on the world map which was generated using locations from the tweets. However, there are some more points which do not reflect the actual data. There is a need to remove these points which is achieved by filtering.

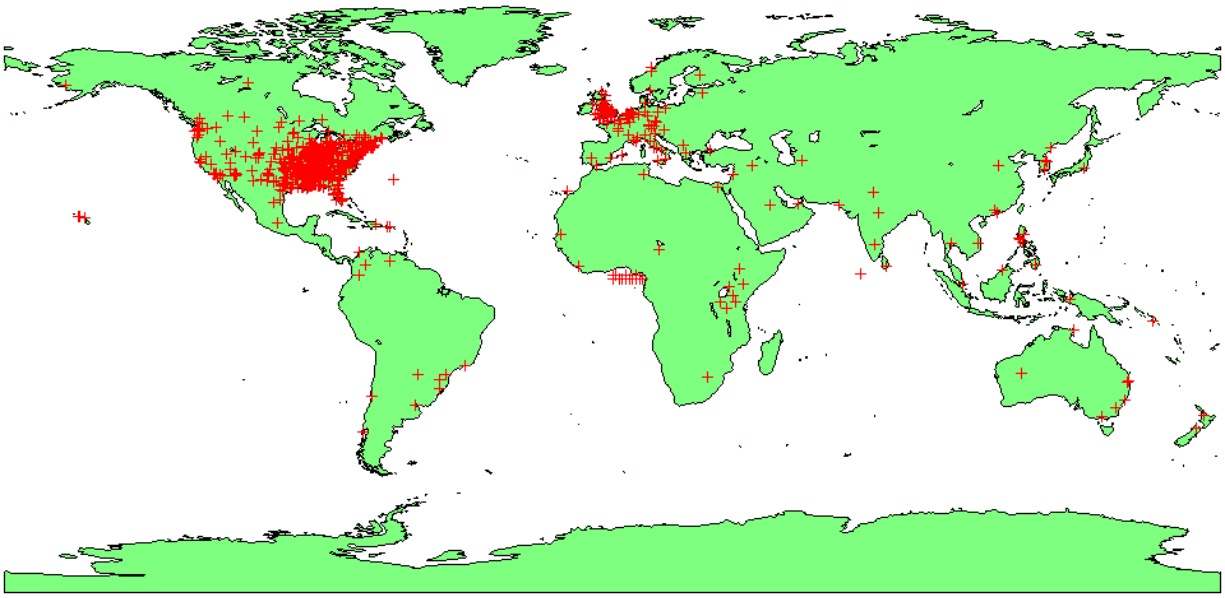

**Figure 3.1 A world map containing all the locations that were extracted from the tweets containing the keywords**

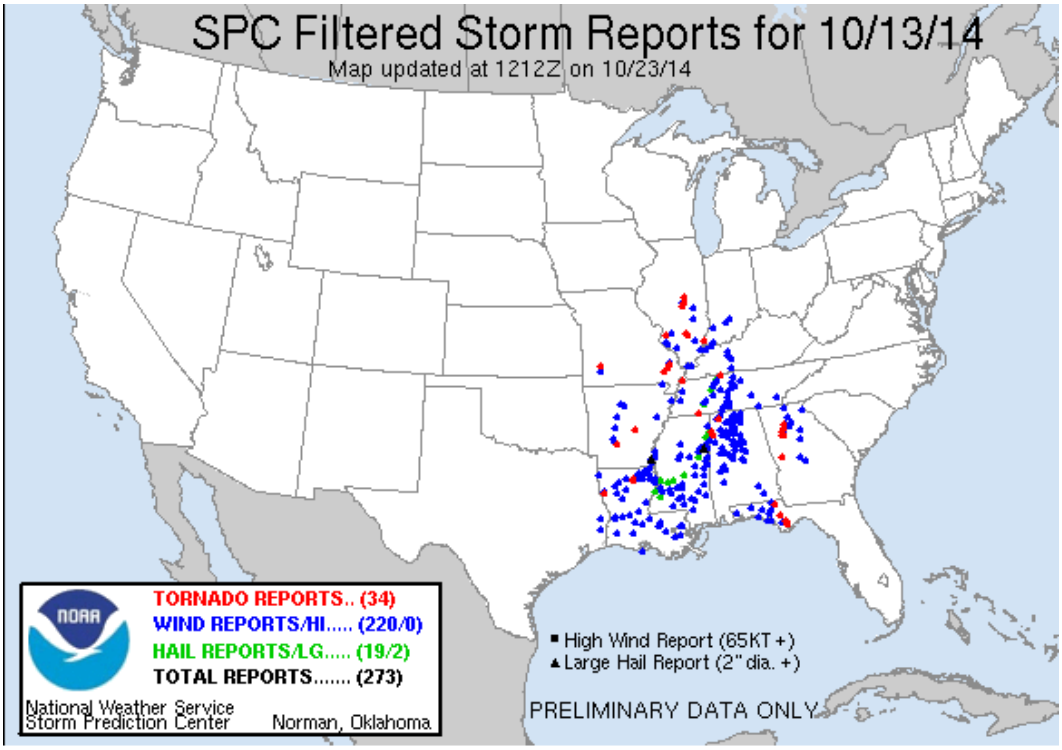


**Figure 3.2 SPC Reports for the Tornado on 10/13/2014**

## 3.4 Clustering the Data for Filtering

The locations that we obtained in the previous step are varied all over the globe. We know that the tornado can only have a limited reach. It is clear on comparing Figure 3.1 Figure 3.1 A world map containing all the locations that were extracted from the tweets containing the keywordsand Figure 3.2 that there are a lot of locations which are not required for building the curve for disaster prediction. On experimentation, it was found that the best number of clusters to obtain is 3 where the cluster with minimum number of samples can be removed. This was obtained by keeping the *min_samples* value as 160 and *eps* = 2.8. Two experimental clusters are shown in Figure 3.3 and Figure 3.4.

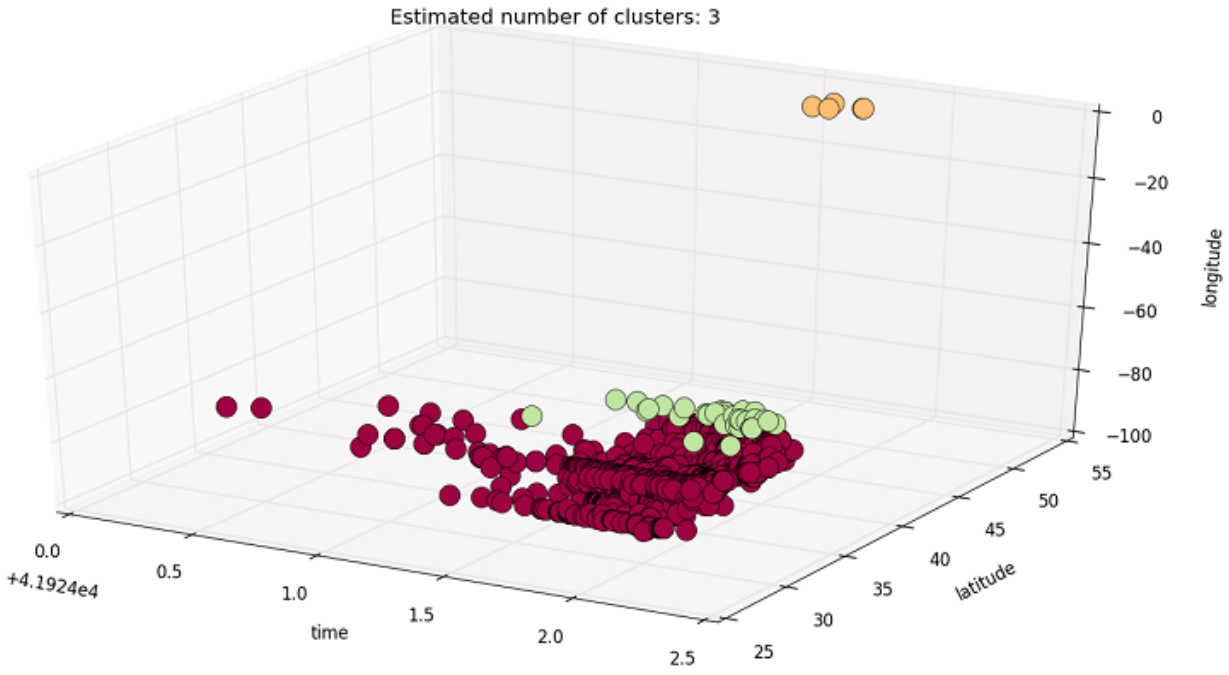


**Figure 3.3 DBSCAN Clustering with *min_samples* = 160 and *eps* = 2.8 resulting in 3 clusters containing only core-samples.**

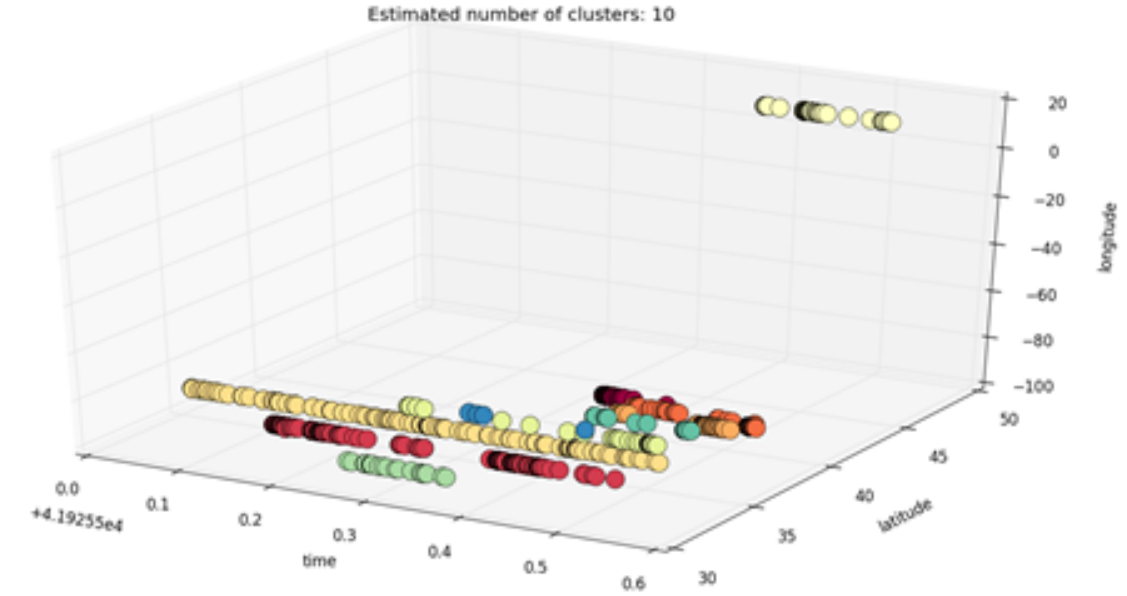


**Figure 3.4 DBSCAN Clustering with *min_samples* = 70 and *eps* = 0.3 resulting in 10 clusters containing only core-samples.**

It is clear in the 3 clusters of Figure 3.3 that the orange clusters are the outliers. Therefore, filtering was performed by removing the orange cluster points. However, it does not represent a complete data set of all the locations. In Figure 3.3 and Figure 3.4, only core samples

are shown. Figure 3.5 shows all the data points with non-core samples being in black. They might be a part of a cluster but they lie on its outskirts. Therefore, we can eliminate them as they are represented by more important core samples. Using this method, the data points were reduced from 4018 to 2144 data points.

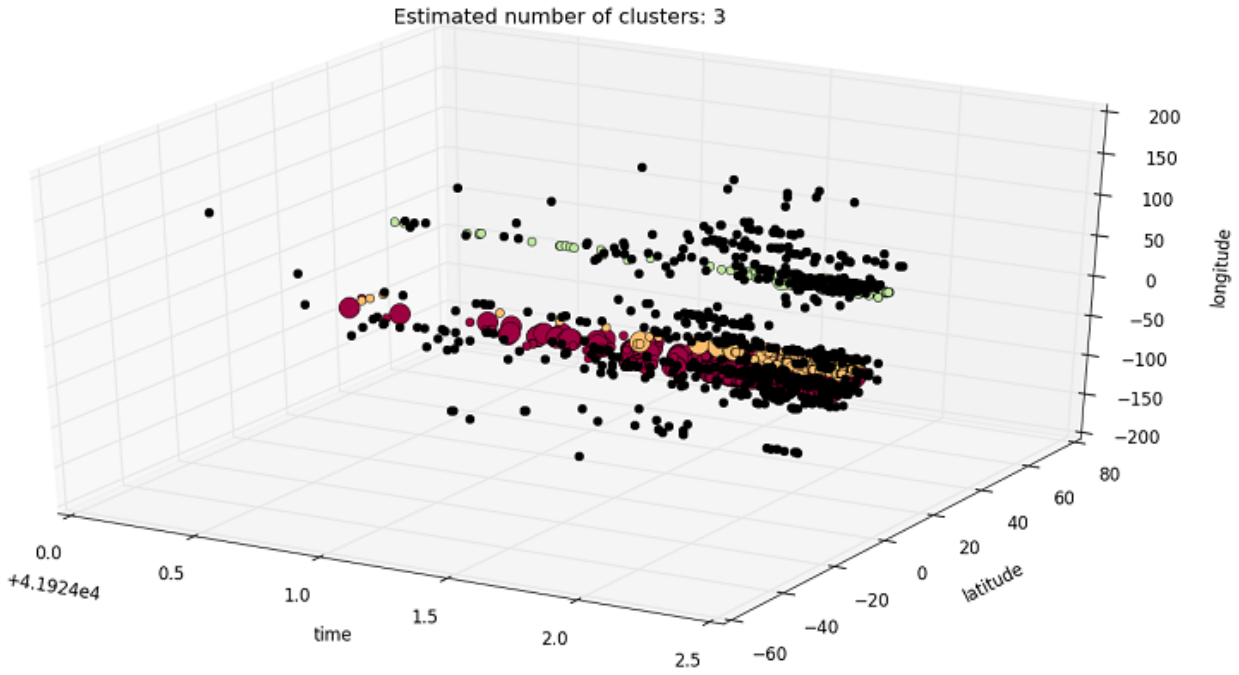


**Figure 3.5 DBSCAN Clustering with *min_samples* = 160 and *eps* = 2.8 resulting in 3 clusters with non-core samples in black.**

The three clusters are plotted on a world map to see where these points lie in Figure 3.6. It is clear that the desired cluster which was on the south-east coast of America is still there and a lot of unwanted points have been eliminated when we compare it to the original data set of points from Figure 3.1. Upon further filtering, we removed the cluster with minimum points to reduce the noise. This is represented in Figure 3.7. These points are used further for curve fitting.

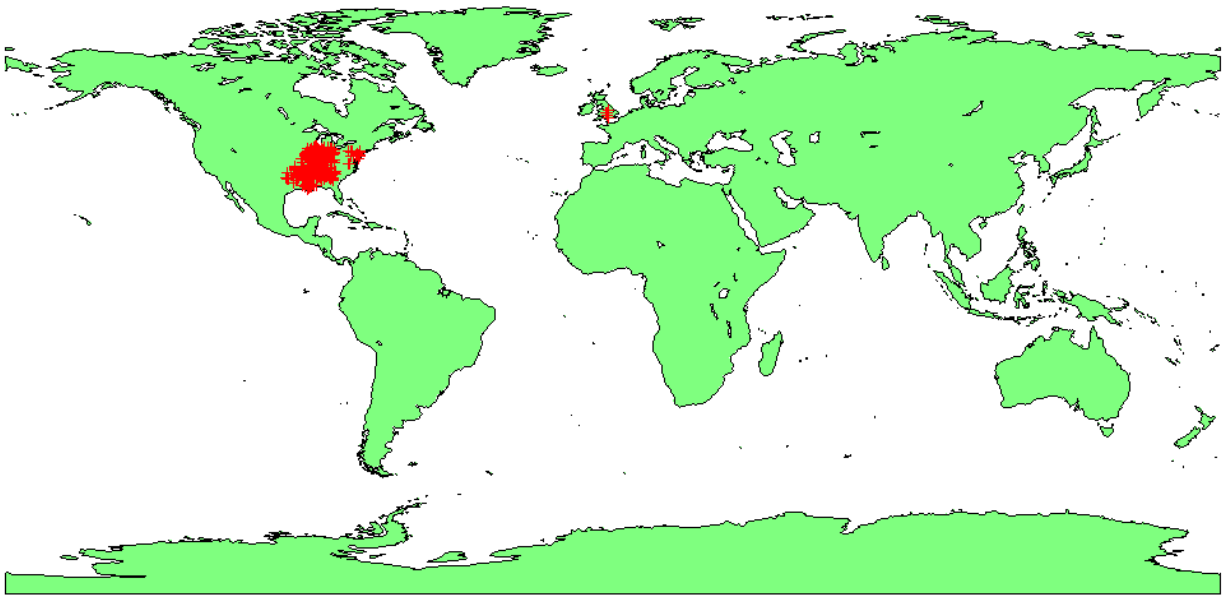

**Figure 3.6 Three clusters obtained after DBCSN clustering represented on a world map.**

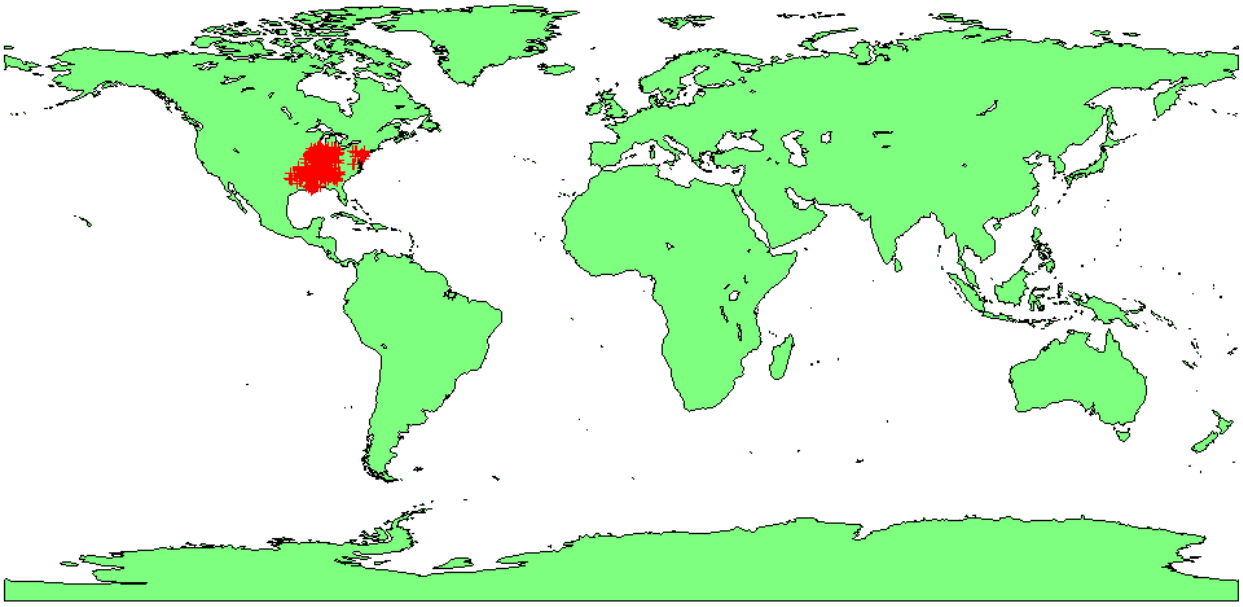

**Figure 3.7 Points after complete filtering on a world map.**

## 3.5 Curve Fitting and Extrapolation for the Trajectory of the Disaster

As reported in section 2.6, results using linear and non-linear regression for extrapolation of the trajectory are discussed below. Since, latitude and longitude were assumed to be independent of each other, there are separate curves for them.

### *3.5.1 Linear Regression Results*

Linear regression will give us a linear equation which will be the best fit for given data. Figure 3.8 shows the plot for Latitude vs. Time with prediction bounds with 95 % confidence bounds. We can see that almost all the points are getting included in the prediction bounds. Similarly, in Figure 3.9, we can see that the prediction bounds of the curve encompass major points for longitude and time. The details about the curve is presented in Table 3.1 where p1 and p2 are the coefficients for the linear polynomial equation.

**Table 3.1 Linear Model for y vs. Time.**

| y | coefficients with 95% confidence bounds | | | | | | Goodness of fit | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p1 | | | p2 | | | | | | |
| | lower | exact | upper | lower | exact | upper | SSE | R-square | Adjusted R-square | RMSE |
| Latitude | 2.646 | 3.323 | 3.999 | -1.68E+05 | -1.39E+05 | -1.11E+05 | 1.86E+04 | 0.04176 | 0.04131 | 2.953 |
| Longitude | 0.009029 | 0.1594 | 0.3099 | -88.2 | -88.05 | -87.9 | 2.66E+04 | 0.002028 | 0.001558 | 3.538 |

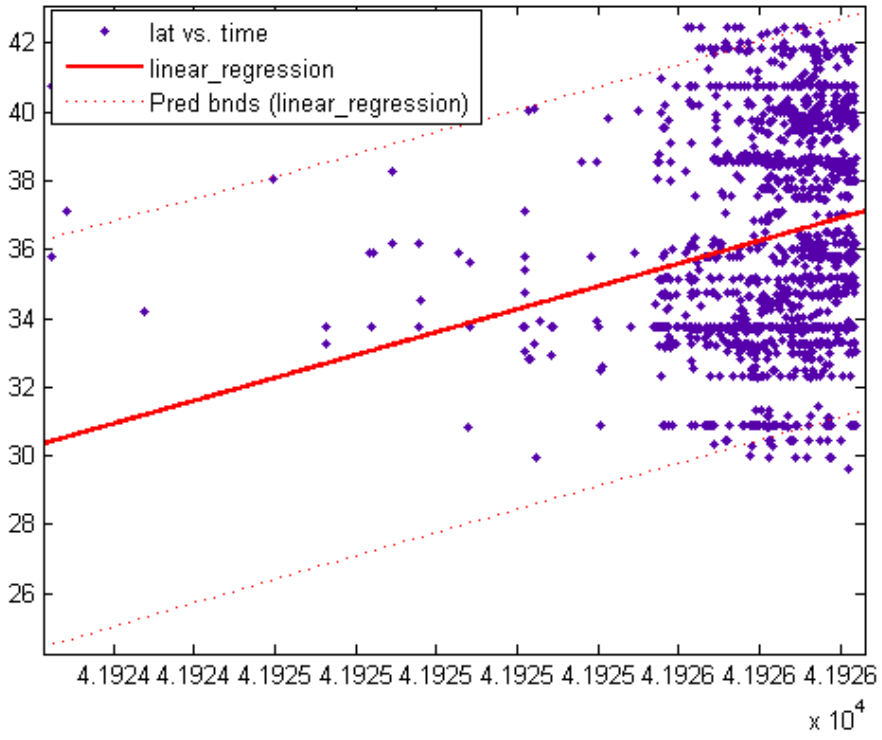


**Figure 3.8 Linear regression for Latitude vs. Time for training data**

On extrapolation, we observe that the curves can represent most of the points of the validation data, i.e. the data of 14th October 2014. This can be seen in Figure 3.10 and Figure 3.11. The blue dots represent the original data and the green dots represent the validation data. Further, I've included the curves with 95% prediction bounds without the data points in Figure 3.12 and Figure 3.13 for a clearer picture of the functions.

Therefore, using the equation parameters of Table 3.1, given a value for time, we can find the values of corresponding latitude and longitude with a confidence of 95%. These values are then reverse-geocoded to result out the locations. Some of the locations which were obtained were places in Texas, Arkansas, Missouri, Indiana, Kentucky and Michigan. These locations are the future values which need to be warned for the disaster. However, as a more future value of time is taken, the locations start coming out to be in Canada.

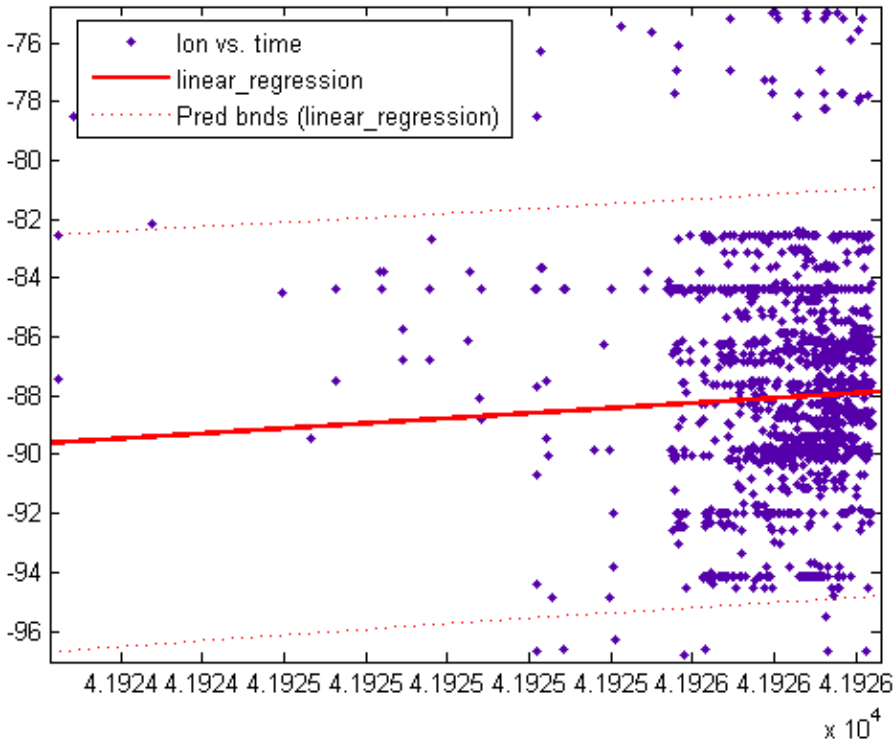


**Figure 3.9 Linear regression for Longitude vs. Time for training data**

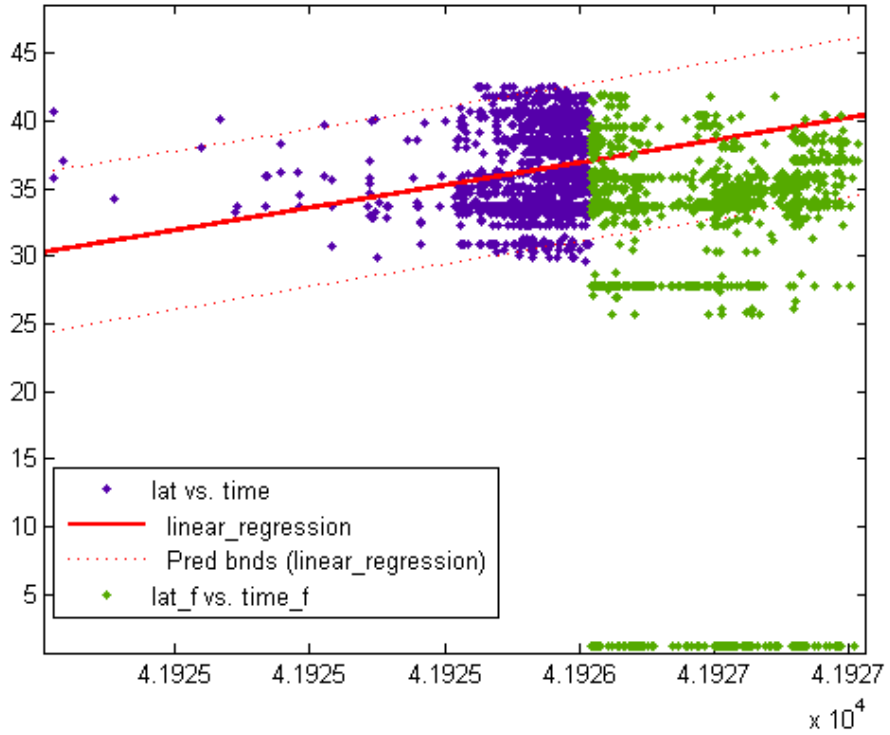


**Figure 3.10 Linear curve for training data extrapolated for Latitude vs. Time**

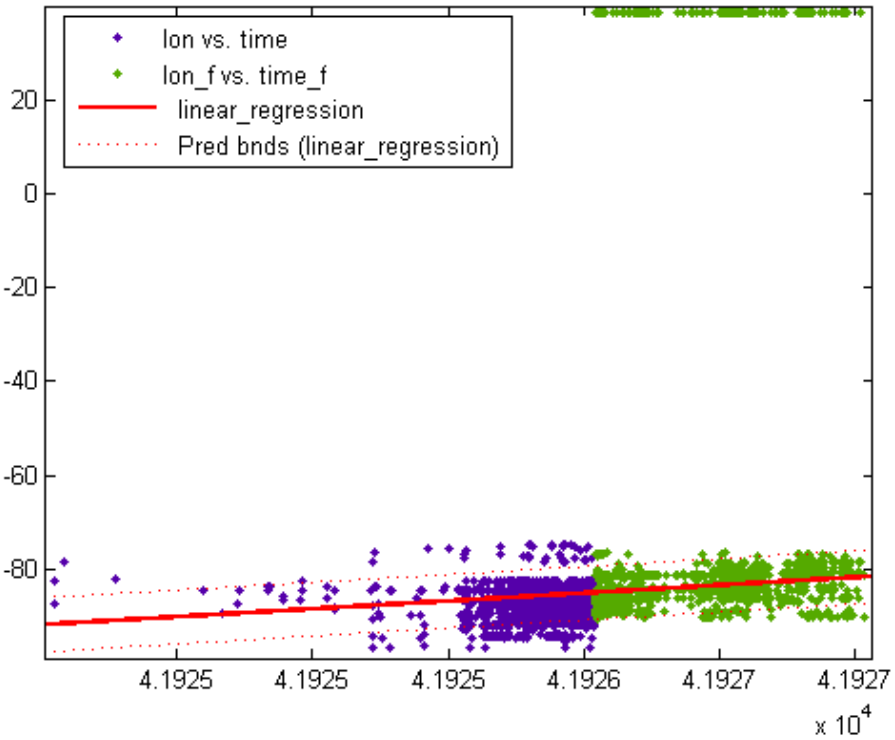


**Figure 3.11 Linear curve for training data extrapolated for Longitude vs. Time**

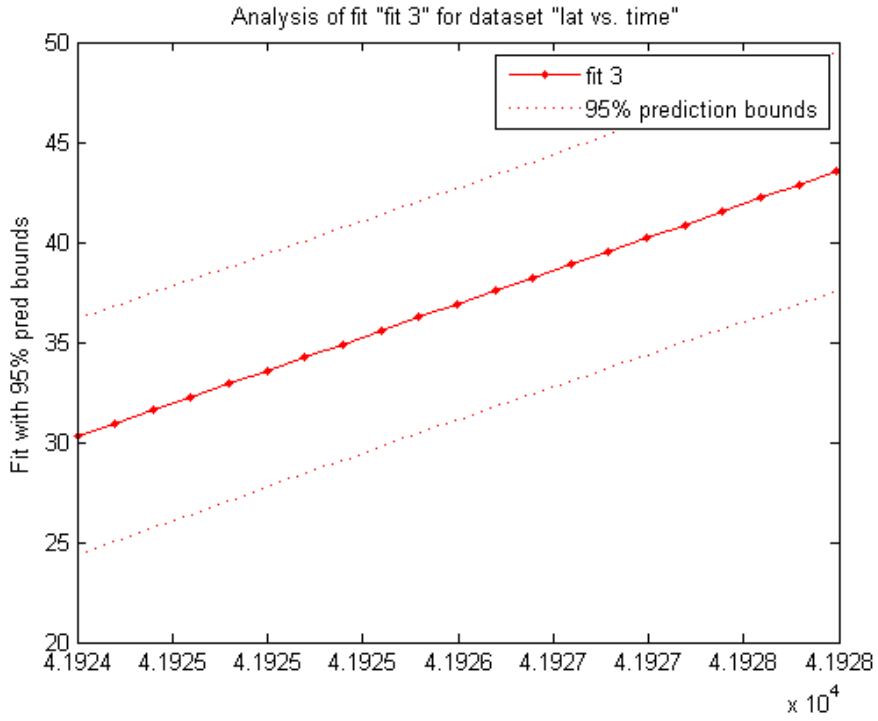


**Figure 3.12 The linear curve with prediction bounds at 95% confidence for latitude vs. time.**

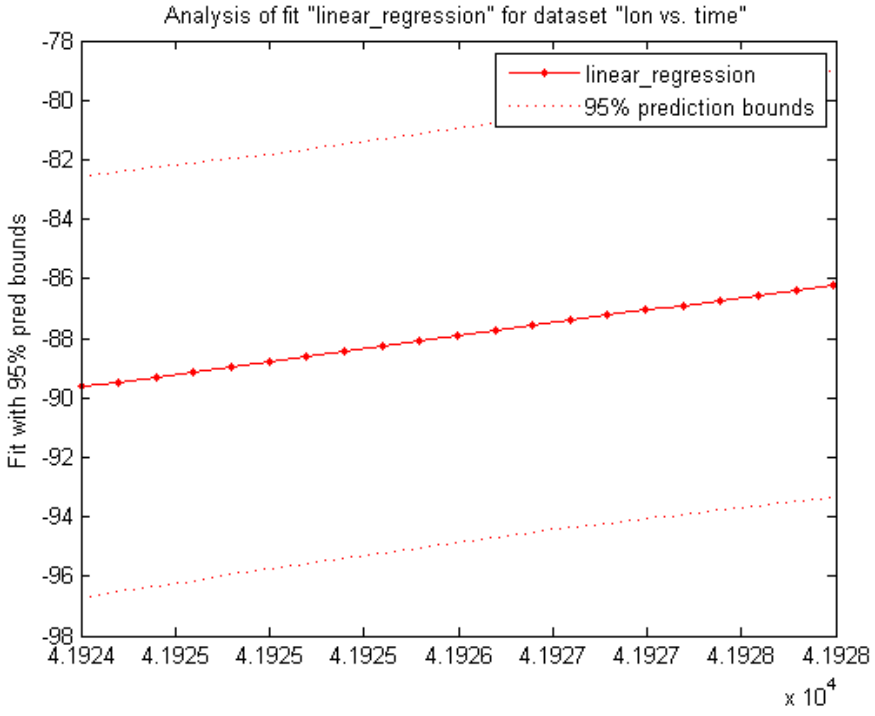


**Figure 3.13 The linear curve with prediction bounds at 95% confidence for longitude vs. time.**

### *3.5.2 Quadratic Regression*

As mentioned in the experiment section, for non-linear regression, I chose to fit a quadratic equation. Figure 3.14 shows the plot for Latitude vs. Time with prediction bounds with 95 % confidence bounds. The quadratic curve for Longitude vs. Time with prediction bounds with 95% confidence bounds is shown in Figure 3.15. The details about the curves is presented in Table 3.2 where p1, p2 and p3 are the coefficients for the quadratic polynomial equation. We can see that the curves are just as good as the linear curves for the training data as they provide a good fit for the given training data. The curves are represented in Figure 3.16 and Figure 3.17. However, we can see on extrapolation and comparing the values with the validation data in Figure 3.18 and Figure 3.19, the values which are obtained by extrapolation of the curve do not coincide much with the actual values.

To validate these curves further, they are used to get latitude and longitude values corresponding to future times. These values are then reverse-geocoded which returned very few addresses in Texas, Arkansas, Missouri, Indiana and Michigan. Most of the values returned Quebec, Canada and it further returned points which lied somewhere in the ocean which is why they were unnamed.

On comparing linear and non-linear (quadratic) regression, we can say that linear gave much more accurate results as they are comparable to the locations marked in Figure 3.2 which reflect official government records. On visualizing the data, we can see that the points are linear in nature and hence it intuitively makes sense that the linear curve is a better fit. Degrees higher than 2, further worsened the fit of the curve and hence were not studied further.

**Table 3.2 Quadratic Model for y vs. Time.**

| y | coefficients with 95% confidence bounds | | | | | | | | | Goodness of fit | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | p1 | | | p2 | | | p3 | | | | | | |
| | lower | exact | upper | lower | exact | upper | lower | exact | upper | SSE | R-square | Adjusted R-square | RMSE |
| Latitude | 0.1025 | 0.1389 | 0.1753 | 0.8986 | 1.071 | 1.242 | 36.14 | 36.26 | 36.39 | 1.81E+04 | 0.06632 | 0.06544 | 2.916 |
| Longitude | 0.1283 | 0.1719 | 0.2155 | 0.5157 | 0.7214 | 0.9272 | -88.37 | -88.22 | -88.07 | 2.59E+04 | 0.0293 | 0.02839 | 3.49 |

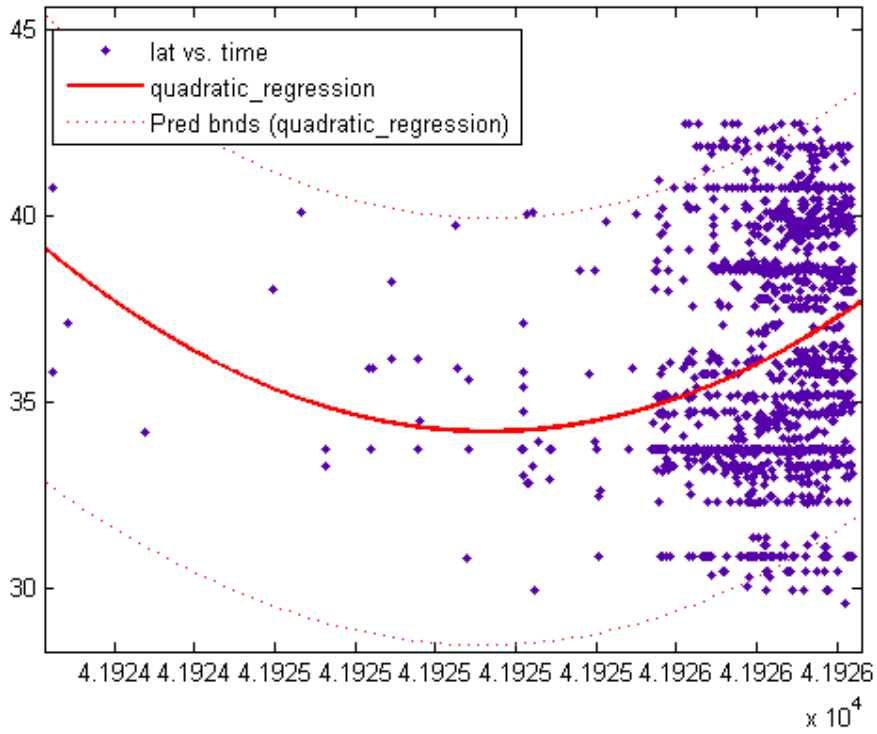


**Figure 3.14 Quadratic regression for Latitude vs. Time for training data**

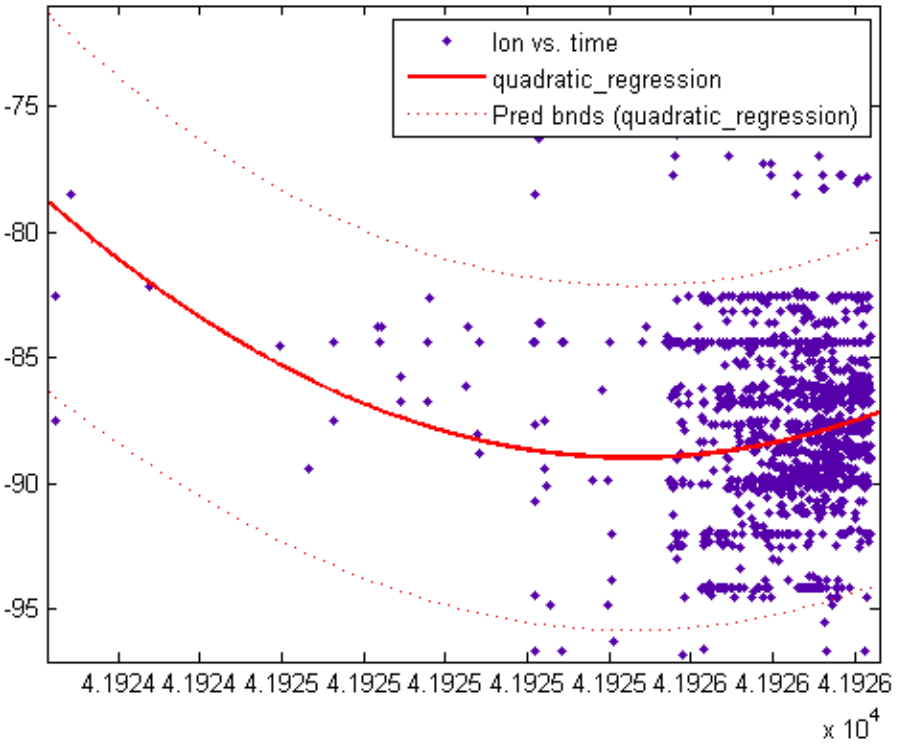


**Figure 3.15 Quadratic regression for Latitude vs. Time for training data**

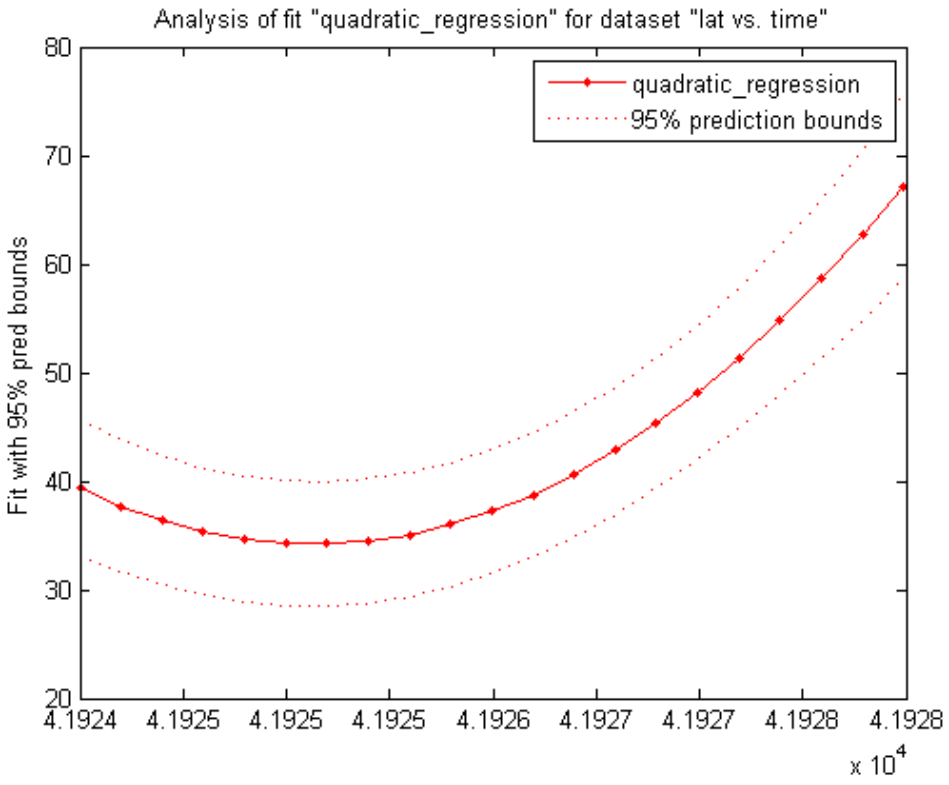


**Figure 3.16 The quadratic curve with prediction bounds at 95% confidence for latitude vs. time**

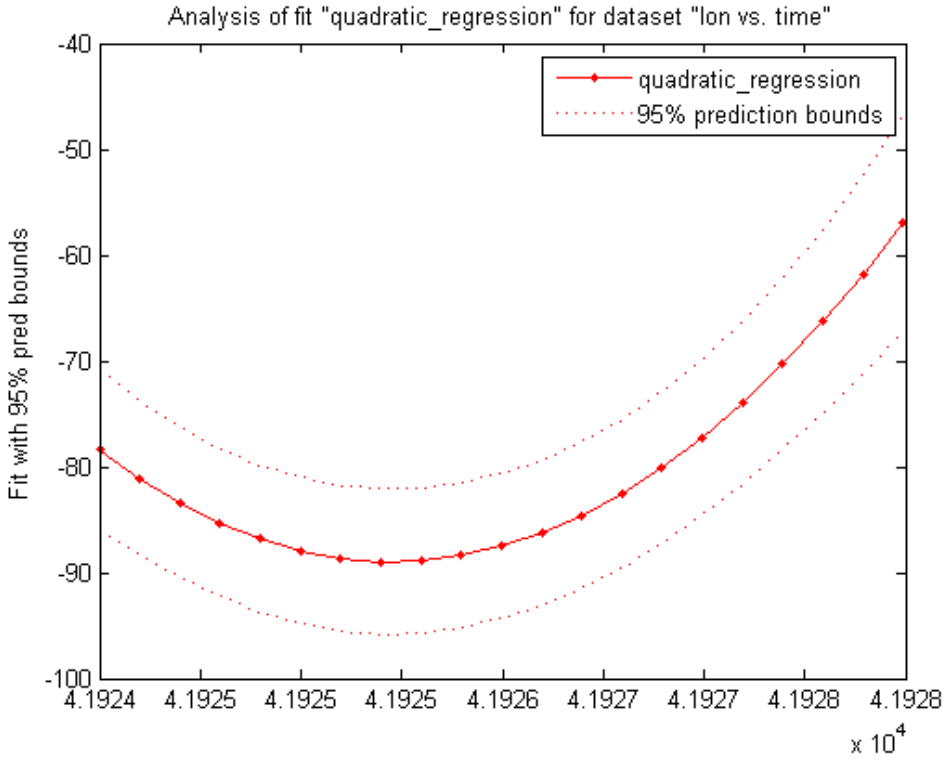


**Figure 3.17 The quadratic curve with prediction bounds at 95% confidence for longitude vs. time**

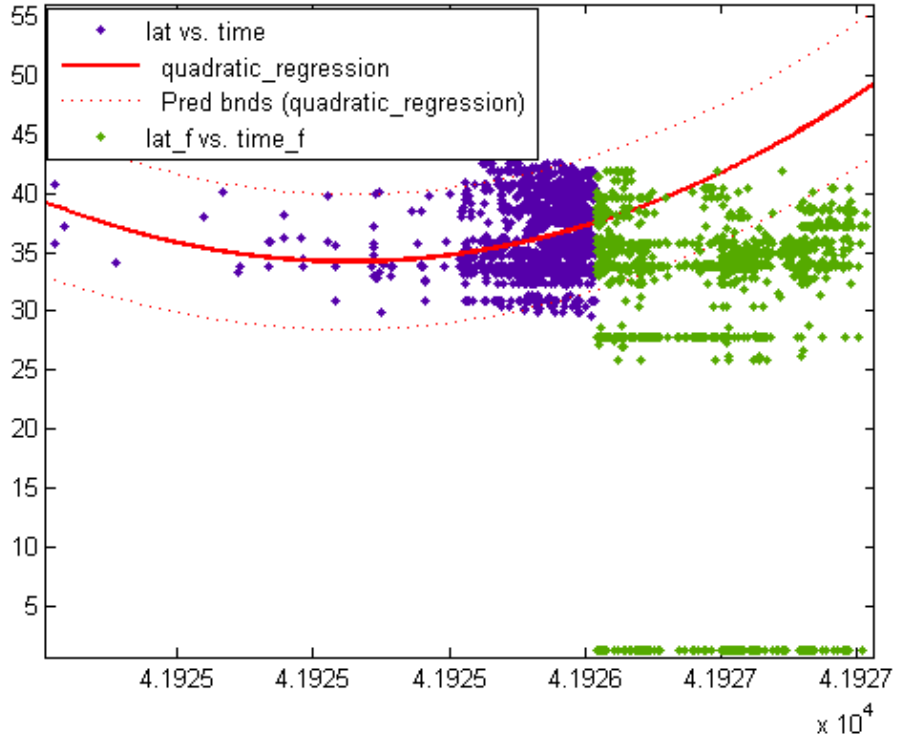


**Figure 3.18 Quadratic curve for training data extrapolated for Latitude vs. Time**

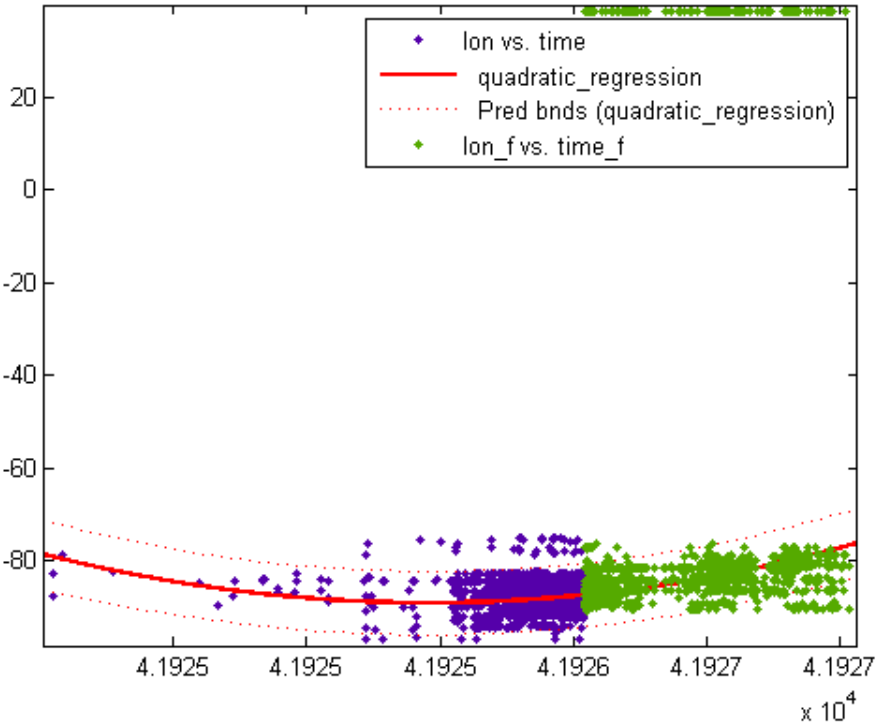


**Figure 3.19 Quadratic curve for training data extrapolated for Longitude vs. Time**

## 3.6 Conclusion

In this chapter, we saw the results that were obtained after each step of the experiment. The last step which was to find a curve, extrapolate it and then validate it against actual data was observed for two different types of curves – linear and quadratic. It was clearly seen that linear curves gave better results and can be accurately used for prediction of the trajectory. In conclusion, the proposed framework was successfully validated for the given dataset for path prediction.

# 4 CONCLUSIONS

Twitter plays a major role in information dissipation in today's world of information and technology. A large amount of data on Twitter comes through the mobile devices which ensures that the data is recent and geo-coded. This aspect can be exploited in a variety of ways to give out important information from Twitter data. One of the most emerging technologies to do so is through Data Mining. A lot of research work has been done on exploiting social network data for data mining for various applications across a bunch of domains. This research work made use of Twitter data for getting a step closer for disaster management in the digital age. Specifically, the problem was to validate the proposed framework for finding the trajectory of a disaster for real-time data.

## 4.1 Summary

Earlier works in disaster management on Twitter dealt with disaster detection, early warning prediction system, dealing with aftermath of a disaster, etcetera. However, the problem of finding the trajectory has not been looked into much by the research community. This research work proposes a framework for finding out the trajectory of a disaster for real-time data extracted from Twitter. The steps that were involved are – 1) Data Acquisition, 2) Data Preprocessing, 3) Time and Location Extraction, 4) Filtering of Data using Clustering, 5) Curve Fitting and Extrapolation for finding the trajectory of the disaster. Finally, the framework was validated by looking at the validation data and comparing the locations obtained by the extrapolation of the curves.

The keywords for data collection were chosen which could completely capture the informative points. I chose to work with the Twitter Search API which had some limitations but it worked well for my experiment since the disaster had already occurred. The Streaming API

would've worked for capturing the tweets as they are posted. Data acquisition provided me with a training set which was used after the operations for getting the time functions for latitude and longitude. There was also a validation data set which was obtained for verifying the results obtained.

Data preprocessing was an important step to get rid of unwanted tweets which can show up later as noise. Only English tweets were considered and tweets containing spam words were eliminated. This cleaned up the two datasets. This framework can be extended to include multiple languages using multilingual data mining. A more sophisticated approach for spam detection can be used for a real-time system.

One of the most important aspects of this experiment is the time and location extraction from the tweets obtained in the data collection stage. My research implemented a fresh approach of getting the location in three ways – 1) The place where the tweet was posted from, 2) the content of the tweet and 3) the location of the Twitter account. I considered all available locations for a tweet. Time and location tuples were sent for filtering in the next step.

. Some of the literature works have also made use of classification techniques for filtering. In my research, I have made use of data clustering for filtering the data. Namely, DBSCAN clustering gave good results as I was able to utilize its concept of core and non-core samples for removing the outliers as well as for sampling. However, if computation power is not an issue, non-core samples can be included which can then be examined for finding out the trajectory to see if a more accurate picture can be obtained.

Finally, linear regression gave good functions of time for latitude and longitude. They were extrapolated for getting the future values of latitude and longitude. These values were

reverse geo-coded to get the location addresses. These were further verified with official government data and the validation dataset for the final validation step.

In conclusion, this research work was able to prove the validity of the proposed framework. It can be tried with a disaster of higher impact to test its validity for higher impact disasters. It can also be used on its own or in a full-fledged disaster management system to benefit mankind. Some of the ways it can be extended and implemented are given in the next subsection.

## 4.2 Future Work

This framework can be implemented in disaster management systems by refining some of the techniques discussed in the paper. A complete independent system can be implemented which will take care of detecting an event, predicting the location and then providing relief for the disaster struck areas. If a disaster has struck, messages can be broadcasted to use a particular *hashtag,* for examples, in case of a hurricane, officials can ask the people to post tweets using *#HurricaneSandy*. This will help the system to obtain data easily. But spammers might misuse this hashtag for spamming which is why sophisticated spam detection systems should also be put in place. The data will be captured efficiently and completely by making use of Twitter Streaming API in case of a real-world application.

Another way to make the architecture more global is to include multilingual support. This would enable the system to mine tweets which are in languages other than English. This will be able to capture global events as well. Twitter features like retweets, URLs, images, videos, etcetera can also be analyzed to form features which can give weights to the locations extracted from those tweets. The locations can also be assigned weights based on how accurate they can be, i.e., locations which are obtained from geo-tagged tweets are given the most weight.

Locations which are obtained from the content are ranked next followed by the location of the Twitter account. These weights can be used in regression to give a better fitting and accurately predicting function.

In this research, clustering is only used for filtering out the data for removing noise and further for sampling. However, it can also be used for figuring out areas which are in need of help, be it in the form of shelter, food, water or medicines. Areas which have similar properties in terms of the damage done will require similar sort of assistance forming them into one cluster. Government agencies can send help according to the needs of the areas which can ultimately lead to saving human lives. Another way that the government can be more prepared is if they know the speed with which a disaster can approach. This is one more aspect that can be looked at in the future. Clearly, the domain of disaster management on Twitter has a wide scope for research which will, in the end, benefit all mankind.

# REFERENCES


[1] "About," 18 January 2015. [Online]. Available: https://about.twitter.com/company.

[2] C. Taylor, "Twitter Users React To Massive Quake, Tsunami In Japan," 10 March 2011. [Online]. Available: http://mashable.com/2011/03/10/japan-tsunami/. [Accessed 7 January 2015].

[3] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment analysis of Twitter data," in *LSM '11 Proceedings of the Workshop on Languages in Social Media*, 2011.

[4] E. Kouloumpis, T. Wilson and J. Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!," in *Proceedings of the Fifth International Conference on Weblogs and Social Media*, Barcelona, 2011.

[5] T. Sakaki, M. Okazaki and Y. Matsuo, "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors," in *WWW2010*, Raleigh, 2010.

[6] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu and B. Liu, "Predicting Flu Trends using Twitter Data," in *The First International Workshop on Cyber-Physical Networking Systems*, 2011.

[7] M. Avvenuti, S. Cresci, M. N. L. Polla, A. Marchetti and M. Tesconi, "Earthquake Emergency Management by Social Sensing," in *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, Budapest, 2014.

[8] P. Ferragina and U. Scaiella, "Fast and Accurate Annotation of Short Texts with Wikipedia Pages," *IEEE Software,* vol. 29, pp. 70-75, 2012.

[9] T.-M. Nguyen, K. Koshikawa, T. Kawamura, Y. Tahara and A. Ohsuga, "Building Earthquake Semantic Network by Mining Human Activity from Twitter," in *IEEE International Conference on Granular Computing*, 2011.

[10] N. Banerjee, D. Chakraborty, K. Dasgupta, A. Joshi, S. Mittal, S. Nagar, A. Rai and S. Madan, "User Interests in Social Media Sites: An Exploration with Micro-blogs," in *CIKM*, 2009.

[11] J. Zhu, F. Xiong, D. Piao, Y. Liu and Y. Zhang, "Statistically Modelling the Effectiveness of Disaster Information in Social Media," in *IEEE Global Humanitarian Technology Conference*, 2011.

[12] A. Kongthon, C. Haruechaiyasak, J. Pailai and S. Kongyoung, "The role of Social Media during a natural disaster: A case study of 2011 Thai Flood," in *Proceedings of PICMET '12: Technology Management for Emerging Technologies*, 2012.

[13] A. Murakami and T. Nasukawa, "Tweeting About the Tsunami? - Mining Twitter for Information on

the Tohoku Earthquake and Tsunami," in *WWW 2012*, Lyon, France, 2012.

[14] "Alabama Tornado Database," National Oceanic And Atmospheric Administration (NOAA), [Online]. Available: http://www.srh.noaa.gov/bmx/?n=tornadodb_2014. [Accessed 28 February 2015].

[15] "SPC Storm Reports - 20141013's Storm Reports (1200 UTC - 1159 UTC)," NOAA's National Weather Service, [Online]. Available: http://www.spc.noaa.gov/exper/archive/event.php?date=20141013. [Accessed 3 4 2015].

[16] "Twitter Usage Statistics," Internet Live Stats, [Online]. Available: http://www.internetlivestats.com/twitter-statistics/. [Accessed 26 February 2015].

[17] "The Search API," Twitter, [Online]. Available: https://dev.twitter.com/rest/public/search. [Accessed 26 February 2015].

[18] "The Streaming APIs," Twitter, [Online]. Available: https://dev.twitter.com/streaming/overview. [Accessed 27 February 2015].

[19] M. Hanksey, "Twitter Archiving Google Spreadsheet TAGS v5," 15 February 2013. [Online]. Available: https://mashe.hawksey.info/2013/02/twitter-archive-tagsv5/. [Accessed 20 October 2014].

# APPENDICES

## Appendix A

Alabama Tornados on October 13th, 2014.

**Tornado 1:**

**EST of the Tornado**: 1929

**County(s) affected**: Colbert

**Damage Scale**: F1

**Path length (Miles)**: 3.48

**Max Path Width (Yards)**: 200

**Fatalities**: 0

**Injuries**: 1

**Location**: 0.2 NW Tuscumbia - 1.1 SSW Wilson Dam

A tornado initially touched down in downtown Tuscumbia , near the intersection of 6th Street and Main Street, where two roofs were peeled off nearby businesses. The brick facade of a small retail building was torn off across 5th Street, and its roof was partially torn off and roofing material vaulted into nearby power lines. The path continued northeast to High Street between 3rd and 4th Streets, where a very old pecan tree was uprooted and fell onto an historic home causing injury to the resident. In this residential area, multiple trees were snapped and uprooted or had large limbs broken off. Several power poles were snapped. Many more old trees were snapped an uprooted across the road along Commons Street between Mulberry and Hickory Streets. At this point, the tornado appeared to weaken somewhat as it crossed a large corn field between Commons Street and King Avenue. It snapped branches upon reaching King Avenue and toppled Bradford pear trees on the grounds of Shoals Hospital along Billy Bowling Drive. As the tornado moved across Grand, Pasadena, and Ford Avenues and Ford Street in Muscle Shoals, it did significant damage to multiple trees throughout the neighborhood. Very minor damage was noted to a couple of houses, but nearly all damage was to trees or power poles. The tornado intensified as it crossed

Ford Street and moved into a commercial district along Woodward Avenue near 2nd Street. There was widespread structural damage to businesses and a church in this area. Numerous awnings were damaged or destroyed, roofs were partly torn off of strip malls, a few windows were broken, and several large signs along the road were blown down. Roofing was peeled off of a used car dealership and gas station and vaulted across the street into a wooded area and another nearby gas station. Power poles were snapped, and two metal power poles were bent, damaging the traffic lights at the intersection. The tornado continued across 2nd Street onto the Tennessee Valley Authority Reservation, where it snapped multiple trees along Garage Road. The tornado lifted before reaching Reservation Road, and little to no damage was noted thereafter.

Start: 34.7318/-87.7029

End: 34.7661/-87.6580

**Tornado 2:**

**EST of the Tornado**: 1848

**County(s) affected**: Lamar

**Damage Scale**: F0

**Path length (Miles)**: 1.08

**Max Path Width (Yards)**: 150

**Fatalities**: 0

**Injuries**: 0

**Location**: 5.7 NNE Beaverton-6.7 NNE Beaverton

National Weather Service meteorologists surveyed the damage in extreme northeast Lamar County and have determined that the damage was the result of a brief EF-0 tornado. The weak tornado touched down in extreme northeastern Lamar County north of Beaverton. The tornado touched down just to the southwest of Henson Springs Road and continued to the northeast. The only damage observed was along Henson Springs Road where 20-30 trees were either uprooted or snapped. Most of the trees were facing

northeast but there were indications of convergent damage. No structural damage was observed in the area. The tornado continued to the northeast over mostly forest and open farm land. The tornado lifted just to the southeast of the Pikeville Country Club before entering Marion County. The start and end points are approximate due to limited road access in the area.

Start: 34.0087/-88.0010

End: 34.0209/-87.9893

**Tornado 3:**

**EST of the Tornado**: 1643

**County(s) affected**: Marion

**Damage Scale**: F0

**Path length (Miles)**: 6.01

**Max Path Width (Yards)**: 50

**Fatalities**: 0

**Injuries**: 0

**Location**: 3.1 NE Bexar-3.9 ENE Shottsville

National Weather Service meteorologists surveyed the damage in western Marion County and have determined that the damage was the result of an EF-0 tornado. A weak tornado touched down north of Interstate 22/US Highway 78 northeast of the intersection of Taylor Road and County Road 11. Along County Road 11, ten to twenty pine trees were uprooted in a convergent pattern, with damage caused to a church by falling trees. The tornado paralleled County Road 11 briefly, causing mainly sporadic tree damage, and then continued northeast over open fields and forest. As it crossed County Road 157 north of County Road 56, several trees were uprooted and tin was pealed back on a farm building. It continued northeastward with very minor damage, crossing County Road 309 and Reid Road. The last damage along the path occurred on County Road 13 just west of Alabama Highway 19, where several pine trees were uprooted. The tornado dissipated rapidly beyond this point.

Start: 34.2080/-88.1077

End: 34.2878/-88.0658

**Appendix B**

List of Spamwords

| | | |
|---|---|---|
| [url= | casino | gambling |
| [/url] | dating | roulette |
| thx | payday | top-site |
| sex | rental | mortgage |
| byob | ambien | pharmacy |
| nude | holdem | dutyfree |
| loan | adipex | ownsthis |
| debt | booker | duty-free |
| poze | youtube | insurance |
| bdsm | myspace | ringtones |
| soma | advicer | blackjack |
| visa | flowers | hair-loss |
| hotel | finance | bllogspot |
| paxil | freenet | baccarrat |
| anime | =-online | thorcarlson |
| naked | shemale | jrcreations |
| poker | meridia | credit card |
| coolhu | cumshot | macinstruct |
| cialis | trading | hydrocodone |
| incest | adderall | leading-site |

slot-machine
carisoprodol
ottawavalleyag
cyclobenzaprine
discreetordering
acetaminophen
augmentation
enhancement
phentermine
doxycycline
citalopram
cephalaxin
vicoprofen
lorazepam
oxycontin
oxycodone
percocet
propecia
tramadol
cymbalta
lunestra
fioricet
lesbian
lexapro
valtrex
titties
xenical
levitra
vicodin
ephedra
lipitor
breast
cyclen
viagra
valium
hqtube
ultram
clomid
vioxx
zolus
pussy
porno
xanax
bitch
penis
pills
male
porn
dick
cock
tits
fuck
shit
gay
ass
gdf
gds
4u
baccarat
car-rental-e-site
car-rentals-e-site
casinos
chatroom
coolcoolhu
credit-card-debt
credit-report-4u
cwas
dating-e-site
day-trading

debt-consolidation

debt-consolidation-consultant

equityloans

facial

femdom

fetish

flowers-leading-site

freenet-shopping

fucking

gambling-

health-insurancedeals-4u

holdempoker

holdemsoftware

holdemtexasturbowilson

homeequityloans

homefinance

hotel-dealse-site

hotele-site

hotelse-site

insurance-quotesdeals-4u

insurancedeals-4u

mortgage-4-u

mortgagequotes

online-gambling

onlinegambling-4u

palm-texas-holdem-game

poker-chip

rental-car-e-site

ringtone

roulette

shoes

texas holdem

texas-holdem

top-e-site

trim-spa

valeofglamorganconservatives