Analyzing HSN Data to Improve Disaster Mitigation

Marni, Lahir
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
mlahir1@umbc.edu

Thalisetty, Chetan Sai Kumar
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
tchetan1@umbc.edu

Gunda, Manasa
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
gmanasa1@umbc.edu

***Abstract*— The need for disaster mitigation has been growing rapidly around the world as the number of disasters each year is increasing exponentially. Earthquakes can happen at anytime and anyplace; the human and financial consequences are hard to predict. In-order to achieve an effective mitigation, we need to take action even before the disaster happens. This paper is aimed to implement a service that predicts the occurrence of earthquakes. The occurrence trend and Magnitude Prediction are obtained using MapReduce jobs and Support Vector Classifier (SVC); a web service is also implemented which predicts the earthquakes that might occur. All the features were implemented as services using Service Oriented Architecture to match the objectives of the course (CMSC 668 Service Oriented Computing).**

Keywords- MapReduce, Support Vector Classification, Web Service, WADL, REST, Service Oriented Architecture, Hadoop.,

I. INTRODUCTION

This project idea is a brain child of the paper "Human Sensor Networks for Improved Modeling of Natural Disasters" written by Dr. O. Aulov and Dr. Milton Halem.

Earthquakes are one of the most disaster causing natural events. They affect everyone and can happen everywhere. Earthquake is the result of a rock underground suddenly breaks along a fault creating seismic waves. As, they cannot be stopped, or prevented, but by predicting them before-hand, one can reduce the amount of havoc caused.
The project requirement is to create a web service that:

1) Analyzes the big data
2) Apply machine learning techiniques on big data
3) Hosts the service on a cloud
4) Implements service oriented architecture

We have created a web service that can predict the occurrence of earthquakes all around the world. The user needs to first select the continent of interest in which he wants to check the prediction. The prediction is made based on the analysis from a dataset which has the collection of earthquakes dating back from 1900. Once the continent is selected the prediction service starts running, and points out the coordinates to the most probable location on map where the earthquake might occur using the Support Vector Classifier and various geo-spatial queries. It not only predicts the location, it also predicts the magnitude of earthquake that might occur. By predicting this beforehand, people and organizations could be aware about the disaster, who could help take precautionary steps, so that many millions of lives and property could be saved.

II. RELATED WORK

Numerous works have been done previously on improving disaster mitigation. One of the works includes the work done by Dr. Oleg Aulov and Dr. Milton Halem. Their work was to show how social media (SM) data can be incorporated into the General NOAA Oil Modeling Environment (GNOME) model to obtain improved estimates of the model parameters such as rates of oil spill, couplings between surface winds and ocean currents, diffusion coefficient, and other model parameters. [1] Few of the other works in this area and which is very much in parallel lines to our project are listed below.

A. *Twitter Earthquake Detector:*

People started using social media like Twitter to share information and emotion about the earthquakes that occur. This social media logging enormously increased after the occurrence of disastrous Sichuan earthquake in 2008. The USGS National Earthquake Information Center (NEIC) processes information from about 2000 real time earthquake sensors, where most of them are present in the United States. Which leaves many other parts of the world not available for processing. USGS was surprised with the effectiveness of the Twitter data for disaster detection. USGS started at looking the data for earthquake detection and verification using the APIs provided by twitter based on several assumptions. They considered most of the people use short messages to tweet earthquake information and limited their analysis by considering tweets only containing maximum of seven words. They also recognized that tweets containing magnitude number and link cannot be considered for first hand analysis and they filtered out these data as well. This detector could recognize the earthquakes happening within 2 minutes depending on the tweets. This detector will analyze the data based on multiple languages because while tweeting most of the people use their own language and use different words which provides the same meaning. For example, Chile has two words for earthquakes: terremoto and temblor. Each of these words are based on the intensity of the earthquake.

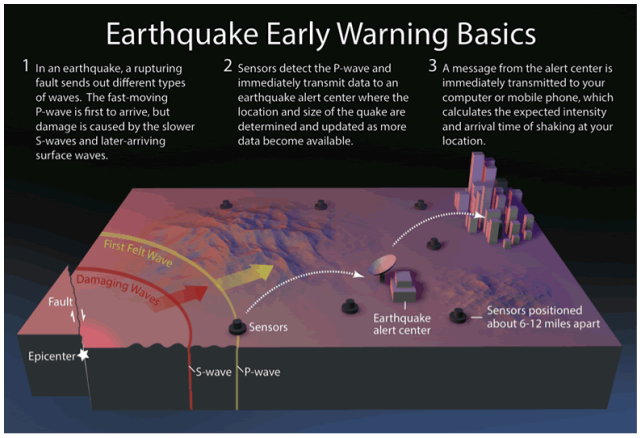

Fig. 1. Image showing architecture of Earthquake early Warning

Fig. 2. A snapshot of the single data element

USGS used the twitter data to check whether their seismic alarms are correct or not. If the seismic alarm happened at a crowded city but there are no tweets that means the alarm was false. But this detector can only use the tweets to log the occurrences. Our system is aimed at predicting the future possible earthquake locations based on the tremors occurred along the fault line which can save more lives.

B. *Shake Maps:*

Shake maps are products developed by U.S. Geological Survey Earthquake Hazards Program collaborating with regional seismic network operators. Shake Maps provide real time maps for ground motion and shake intensity allowing the USGS officials to identify the impact and intensity of an earthquake. These maps can be used by any public or private organizations like post disaster management teams etc. for post-earthquake recovery and response. These maps cannot predict, post event occurrence they mention the intensity of event in terms of shake and ground motion.

C. *Earthquake Early Warning:*

Earthquake Early Warning (EEW) uses earthquake science and sensor network to generate alerts to the post disaster management teams present. Sensors are situated on the surface, which will continuously wait for so called P-waves. In an earthquake, a rupturing event sends out various signals among them the fast-moving waves P-waves reach the earth surface first. S-waves and surface waves which cause major destruction follows later. So, the main idea is to detect the P-waves before other wave reach surface. Once the sensor detects the P-waves using multi-hop communication all the detected sensors sends the information to the base station present around. Base-station validates and processes the information before sending the alert information to the respective post disaster management organizations. Fig 1, shows the brief working of a EEW.

In February of 2016 the USGS, along with its partners, rolled-out the ShakeAlert early warning test system in California. The system includes geographically distributed servers, and allows for automatic fail-over if connection is lost. This next-generation system will not yet support public warnings but will allow selected early adopters to develop and deploy pilot implementations that take protective actions triggered by the ShakeAlert warnings in areas with sufficient coverage. To make this network efficient, every place on the world should have sensors and base stations deployed this is one of the major drawback with this system. If malfunction of any of the sensors deployed may not provide the base station with proper data. Even this system cannot predict the occurrence of an earthquake early. This system can maximum provide information prior occurrence of an earthquake in order of minutes which is not sufficient to vacate the people in most of the cases.

III. DATASET

Two sources of data-sets are used for the project,

1) Dataset from USGS
2) Twitter feed data.

The primary dataset is acquired from United States Geological Survey (USGS) which contains information about all the earthquakes that had occurred from years 1900 to till date. A WebCrawler is written in python to get the data from USGS site, as manually downloading 20 million entries is a very tedious and close to impossible. The WebCrawler built with wget Linux tool is used to get the data from the USGS site and automatically populates the dataset in an xml format. In the later sections of the paper you will see more about how this data is processed to attain required results. A snapshot of the single data element is shown in the Fig.2, for reference.

The second dataset is the continuous twitter data feed that is used to estimate the occurrence of the earthquake based on a model created using the previous dataset. The main challenge here is to find the reliability of data which is explained in detail in Section VI.

IV. ARCHITECTURE AND TECHNOLOGIES

The architecture mainly consists of a HTML user interface, NODE JS is used to implement the RESTful service, MongoDb is used as the database to perform queries, dataset is processed using Hadoop and Machine learning algorithms

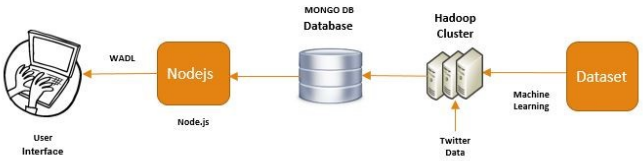

Fig. 3. A brief overview of the Service

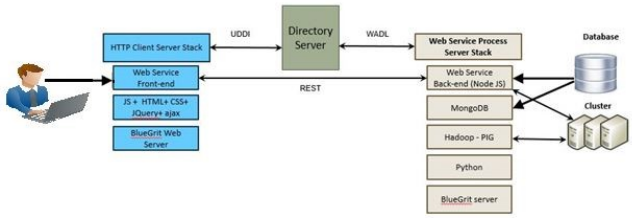

Fig. 4. Detailed Architecture of the Service

are implemented on the data and then the data is hosted on the Bluewave cluster. The data from twitter is also fed into the bluewave cluster. A brief design of the architecture can be shown in the Fig. 3.

Technologies used to implement this project have been listed below.These technologies are mapped with architecture diagram and shown in the Fig. 4.

1) JS, HTML, CSS - front-end user interface
2) Node JS (Express) - RESTful API implementation
3) PIG - To process dataset using Map-Reduce
4) Support Vector Classifier - It approximates the earthquake occurrence zone and magnitude. (Machine Learning)
5) MongoDB - used as a Database to store database and to run all geo-spatial queries.
6) Python - To get data from twitter, to write web-crawler and other scripts.
7) Json - To parse data (strings) to HTML from RESTful API.

A detailed explanation of each block and why the particular technology is chosen is explained in detail in the next section and the fig. 4 , shows a detailed architecture of the block.

V. SERVICE IMPLEMENTATION

A. *Twitter Data:*

Twitter data has to be taken continuously and this data is analyzed for predicting the future occurrences of earthquakes in any location. For this purpose, by using the APIs provided by the Twitter we collect data continuously and process them in chunks using map-reduction techniques which will be explained in the following sections.

Getting twitter data from Twitter APIs: API stands for Application Programming Interface which makes the interaction with computer programs and web services easy. Many web services provide APIs to developers to interact with their services and to access data in programmatic way.

Step 1: To get Twitter API keys

We need to follow the following information in order to get the keys provided by the Twitter.

1) API key
2) API secret
3) Access token
4) Access token secret

Certain steps should be followed to receive the API key and secret information along with access tokens which is mentioned below.

1) Create a twitter account.
2) Go to https://apps.twitter.com/ and log in with your credentials.
3) Click ”Create New App”
4) Click ”Create your Twitter application” by filling out the form and agreeing to the terms and conditions.
5) In the next page, under ”API keys”, your ”API key” and ”API secret” are present.
6) Once you click on ”Create my access token”, your ”Access token” and ”Access token secret” will be created.

Step 2: Connecting to Twitter Streaming API and downloading data using python

Python library called Tweepy is used to connect to Twitter Streaming API and to download the data. To get the data from the Twitter you must be a genuine user which is checked by the Twitter using your access_token, access_token_secret, consumer_key, and consumer_secret provided during the authentication phase.

Step 3: Filtering the data obtained

Data obtained from the twitter API is very huge and contains earthquake related tweets as well as other tweets. So, the data related to earthquakes can be filtered using the hashtags like #earthquakes, #quakes, #catastrophe etc., all this filtered data must be further processed for location, magnitude and accuracy of an earthquake occurrence which is done using the map reducing techniques which is explained in detail in next section.

B. *Pre-processing the USGS and twitter Data:*

We have done Map-reduce to pre-process the data, as the data is huge and it takes enormous amount of time to process the data. PiG scripts were written which enhances the and makes the map-reduce processes easy. Apache PIG is a platform which can be used to analyze larger data sets in Hadoop. To write data analysis programs, PIG provides a high-level language called PIG Latin. This language allows programmer to develop their own functions for reading, writing as well as for processing data. All the scripts written using the PIG Latin are converted to Map and Reduce Tasks. PIG Engine provided by the Apache PIG takes this PIG Latin

```
_id: ObjectID('5841e71d28de5dfcee6542a1')
time: "1930-01-18T07:04:02.000Z"
latitude: -4.61
longitude: 153.176
magnitude: 6.5
year: "1930"
month: "01"
▼ loc: Array[2]
    0: 153.176
    1: -4.61
```


Fig. 5. Figure showing a document in the MongoDB

Fig. 6. All earthquakes that had occurred with magnitude more than 6

scripts as inputs and convert those scripts in to map reducing jobs. Apache PIG uses multi query approach which helps to reduce the length of the code and provides nested data types like tuples, bags and maps that are not present in MapReduce. PIG has many advantages when compared to other languages like HIVE etc.,

Preprocessing is performed on the USGS datasets, all the unwanted data is removed and feature extraction of 20million rows is done using the PIG scripts which helped in reducing the processing time by a very huge amount. Also, twitter data is very huge, where almost Gigabytes of data is obtained for every week and for preprocessing of this data to obtain trivial information is done using the PIG scripts. All this huge data is present in the Hadoop file system (HDFS) from where pig scripts preprocesses and stores them in a CSV file which is then stored in MongoDB.

Data obtained from both USGS and the preprocessed twitter data is stored to the MongoDB where further processing takes place. Data processing in MongoDB is explained in further sections.

C. *Geo-spatial queries with MongoDB:*

Data obtained from Twitter and USGS is pre-processed to get the location (latitude, longitude) and magnitude fields. All this data is stored in the mongoDB for further analysis. The document stored in the mongoDB is shown in fig. 5. List of all the points that are in DB with a magnitude greater than ”6” and on which the analysis is to be performed is plotted on the map and shown in fig. 6.

Our concentration was to find the minor tremors occurred in a year and earthquakes above 6 magnitude . To calculate this, we need to find the distance between two locations. Since earth is not flat we cannot simply calculate the Euclidian distance. We need to calculate haversine great circle distance between two geo locations obtained from the twitter and USGS data. MongoDB provides geo-spatial queries to calculate haversine distance between these locations. The queries are performed over the complete database and an updated table is again stored in the MongoDB. This table is read from python scripts to implement Machine Learning algorithm on it, which is discussed in the next topic. These calculations could also be done using the pig scripts.

D. *Support Vector Classifier (SVC):*

SVC is supervised learning methods used for classification. It is a very memory efficient technique. In he project, we use ”sklearn” library from python. SVC is used in the present project mainly for two purposes, and their implementations

1) Classifying the given location to potential earthquake threat or not,
2) If it is threat, then classifying the intensity based on magnitude.

To classify the given location to potential earthquake threat or not, we take the probability of earthquake to occur on any co-ordinate on earth to be equal to zero. So, in the start all the earth is assumed to be perfect Faultline-less land with a zero probability of earth quake occurrence. Now we start noting the faults in the crust by taking into consideration of all the major earthquakes. Once this is noted, We get the analyzed data from MongoDB, that has all the minor earthquakes caused in the previous couple of weeks that lead to the occurrence of earthquakes. These form the dimensions for the SVC. The SVC is trained with this data and when and the twitter data is continuously being given to this trained model. Depending on the present occurrence and the dimension and data from the present data, it classifies the input data to be an yes or no for the occurrence of the earthquake. Now the each earthquake sample is also classified against a magnitude at which it had occurred, its value ranges from 0 to 10. Explaining in a short note, two classifications are done basing on the same samples, one is, the pattern the minor earthquakes or tremors occur that result in major earthquakes and the second is the magnitude with which they occur. So, two classifications are made with the same sample data. The python program reads the current data from mongodb which in-turn gets it from the twitter and classifies the data into a potential earthquake and also assigns a magnitude to it. After classifying, all the potentially could occur earth quakes are converted into a json string with co-ordinates and magnitude as parameters and stored locally in the Bluewave server. These json files are always being updated basing on the live

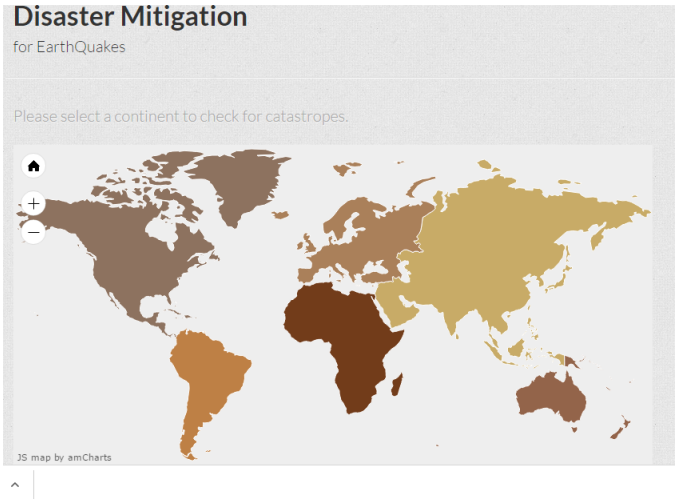

Fig. 7. Figure showing the web-page of The user interface

twitter thats is being input to the SVC classifier trying to improve the probability of prediction and the scale of disaster it could cause.

E. RESTful API and User Interface:

Node JS service is written over the express framework to implement the RESTful APIs. Before knowing the RESTful implementation, it is good to understand the User interface and the possible queries it could post to the REST service. The User Interface consists of a map of the world and is implemented in HTML/JavaScript with an attractive and user-friendly CSS, when clicked on any continent a request is sent with a continent name requesting a json string containing the list of potential earthquake locations that could occur. The Javascript in HTML takes the string response sent by the server, parse the content and displays the locations on the screen.

Once the continent is clicked, a request is prepared for the respective continent and Sent as a call to REST API, the node js running a REST service, gets the request and then validates the request, It reads the json file according to the request sent, and sends the respective json back to the HTML which will parse and display the data. A RESTful implementation is described using a WADL file which is attached in the Appendix for further references.

VI. CHALLENGES

Few challenges were encountered during the project implementation and are briefly explained:

1) USGS provides a web service which can be used to retrieve the earthquakes information, but the data that can be fetched is limited to only 20,000 records for any single request. To retrieve the data for 20 million entries, we had to use a web crawler to download all 20 million entries of data at a time to an XML file
2) The REST Service has to authenticate every user from any platform and using any browser, So explicit authentication headers had to be added for universal deployment of the service
3) The continents are surrounded with international waters that doesnt belong to any continent, to include latitude and longitude of these locations into a continent has become a huge issue, a python service is implemented by writing our own geo-spatial queries so that we could even predict the earthquakes in that location.
4) Since earth is spherical, euclidean distance does not result in the accurate distance between two points. To solve this we have used the geo-spatial queries provided by MongoDB. AS mongoDB has internal mappers and reducers, very complex calculations could be performed in very short amounts of time.
5) Twitter was used to obtain the social media data and twitter API streams a lot of data which is not relevant for our usage and cause a setback in model. to avoid unwanted data we used certain relevant keywords to filter data and the reliability of data is assured by making sure it is from reliable sources.

VII. RESULTS

The HTML web page takes the continent as an input by clicking on the map. Then the service runs and displays the results generated according to user selected continent. We point the predicted location with the latitude and longitude obtained and also display the predicted magnitude.

As this experiment depends on the occurrence of earthquakes which is a very indefinite phenomenon, we simulated the environment by passing the twitter data that has been collected for almost two months before the occurrence of the earthquake in New Zealand (North Canterbury) on 14 Nov 2016. This earthquake had a magnitude of 7.8 on momentum magnitude scale. And our service could predict the occurrence of that earthquake with almost the magnitude of similar scale. Please refer to the figures 8 and 9,for actual and predicted location on the map as shown in Result 1. After the demonstration for the project has been given, there was another earthquake that had occurred in marina island on 21st December, 2016,which was also predicted by our model. A few more major earthquakes had happened in the past month of which our model had fairly detected almost all of them with almost the very accurate magnitude as in shown in figures 10 and 11 in result 2.

VIII. FUTURE SCOPE

The mode and accuracy can be improved by increasing the number of features on which the analysis is done. This nothing but, increasing the number of dimensions of the sample in the SVC. This not only increases the accuracy but also helps analyze the precautions to be taken and the area that can be affected by the earthquake. Instead of Social Media stream (twitter), more reliable data (USGS data, New Zealand Earthquake Report or Geoscience Australia Earthquakes) can be used to get better and more accurate prediction of the earthquakes.

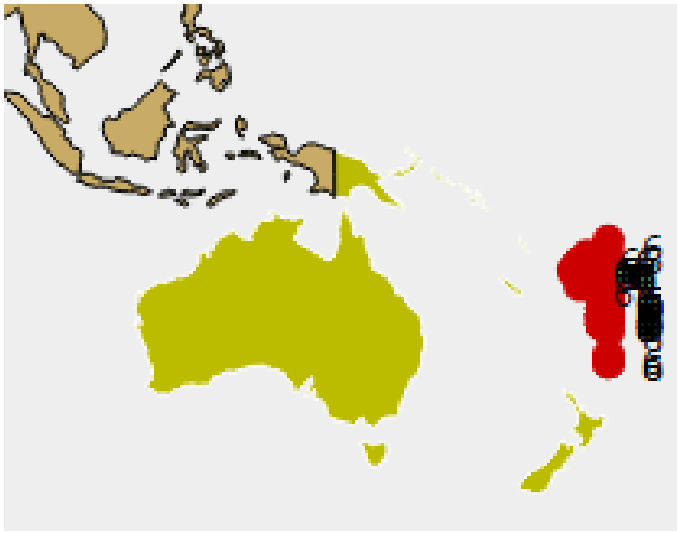

Fig. 8. Image showing the Predicted Earthquake location in Result-1(Sample Data)

Fig. 9. Image showing the actual Earthquake location in Result-1(Sample Data)

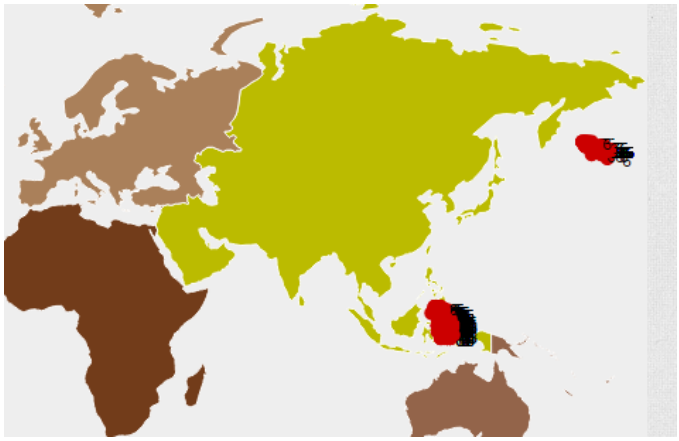

Fig. 10. Image showing the Predicted Earthquake location in Result-2(Live Data)

Fig. 11. Image showing the Predicted Earthquake location in Result-2(Live Data)

IX. CONCLUSIONS

Earthquakes is one of the most disastrous natural calamity. It causes massive loss of life and property. By predicting this disaster, it could help people from saving their lives and property. Hence, we have created a web service which helps in predicting earthquakes that might occur. It helps the organizations that could help minimize the loss of both life and property. It could also help scientists and geologists in studying the occurrences of earthquakes which in-turn could improve our model.

ACKNOWLEDGMENT

We would like extend our gratitude towards Dr. Milton Halem for his constant guidance and Support, without whom this project would still remain a dream. We would also like to thank the TA, Yin Huang for his extended help in installing and solving all issues related to Bluewave computing platform.

REFERENCES

[1] O. Aulov and M. Halem, "Human Sensor Networks for Improved Modeling of Natural Disasters," in Proceedings of the IEEE, vol. 100, no. 10, pp. 2812-2823, Oct. 2012. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6235982

[2] Aulov, O., Halem, M., & Price, A. AsonMaps: A Platform for Aggregation Visualization and Analysis of Disaster Related Human Sensor Network Observations. Available: http://www.iscram.org/legacy/ISCRAM2014/papers/p97.pdf

[3] Shubhendu S. Shukla, Disaster Management: Managing the Risk of Environmental Calamity, in International Journal of Scientific Engineering and Research (IJSER), vol.1 Issue 1, September 2013. Available: http://www.ijser.in/archives/v1i1/MDMxMzA5MTE=.pdf

[4] Andrew Hooper, Senior Member IEEE, Fred Prata, and Freysteinn Sigmundsson, Remote Sensing of Volcanic Hazards and Their Precursors, in Proceedings of the IEEE, vol. 100, no.10, October 2012. Available: http://www.citg.tudelft.nl/uploads/media/Hooper_et_al_Proc_IEEE.pdf

[5] Takeshi Sakaki, Makoto Okazaki, Yutaka Matsuo, Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors, Available: http://www.ymatsuo.com/papers/www2010.pdf

[6] USGS. Available: http://earthquake.usgs.gov/

[7] SVC. Available: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[8] MONGODB. Available: https://www.mongodb.org/
[9] REST. Available: http://en.wikipedia.org/wiki/Representational_state_transfer
[10] PIG. Available: https://pig.apache.org/docs/r0.7.0/tutorial.html
[11] Twitter Development Documentation. Available: https://dev.twitter.com/rest
[12] Amcharts. Available: https://www.amcharts.com/visited_countries /#
[13] Qu Guosheng, Li Yigang, Ning Baokun et al. (2006), A Preliminary Study on Quick Estimating Model of Earthquake Catastrophe, Journal of Basic Science and Engineering, Supplement, pp 611.
[14] Richard D. Knabb, Jamie R. Rhome, and Daniel P. Brown, the Tropical Cyclone Report-Hurricane Katrina(2005 Augusts 2330), NOAA'the National Hurricane Center, December 20 2005.
[15] Liu Qiyuan, Wang Jun, Chen Jiuhui, et al. (2007), Seismogenic tectonic environment of 1976 great Tangshan earthquake: results given by dense seismic array observations. Earth Science Frontiers. 14 (6), 205213.
[16] Zhang, J.Q. and Li, N., (2007), Main quantitative methods for assessment and management of meteorological disaster risks and their applications.
[17] Mitigation. Available: https://www.fema.gov/what-mitigation

APPENDIX

WADL file for the RESTful API is placed below for further references.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application xmlns="http://10.100.16.120:8888">
   <grammars/>
   <resources base="http://10.100.16.120:8888/">
      <resource path="{continentid}">
         <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:int" style="template" name="countryId"/>
         <method name="GET" id="asia">
            <response>
                        <representation element="responseMessage" mediaType="application/json"/>
            </response>
         </method>
         <method name="GET" id="africa">
            <response>
                        <representation element="responseMessage" mediaType="application/json"/>
            </response>
         </method>
         <method name="GET" id="australia">
            <response>
                        <representation element="responseMessage" mediaType="application/json"/>
            </response>
         </method>
         <method name="GET" id="north_america">
            <response>
                        <representation element="responseMessage" mediaType="application/json"/>
            </response>
         </method>
         <method name="GET" id="south_america">
            <response>
                        <representation element="responseMessage" mediaType="application/json"/>
            </response>
         </method>
         <method name="GET" id="europe">
            <response>
                        <representation element="responseMessage" mediaType="application/json"/>
            </response>
         </method>
      </resource>
   </resources>
</application>
```