Health Weather: Real Time Analysis of Flu Outbreaks Using Social Media

Sai Kishore Bandaru, Jagadish Rao, Ebenezer Anand Arapally, Priyanka

Abstract-- "Health is Wealth" is the popular slogan we hear all the time. Now a days certain flus are spreading rapidly and also with less awareness. However we can see many people tweeting them or posting such related message sin social network. But do we notice them? Can we view other people tweets? Cane we validate such tweets? Keeping these points in view, we have developed an application "Health Weather" which can analyze the social media messages in order to identify the severity of flus and recommend certain remedies or suggest nearby hospitals for checkups at a particular location.

1.Introduction

In general we have many applications which provide latest news. But most of them include more details and not precise to disease related information. There is lots of data available in many websites regarding health. But these data can be only used for batch processing. The whole point of developing this application is to make a real time processing of social media data to provide information about the flu outbreaks at a particular location. And also make some recommendations to the user regarding the precautions to be taken for the flus, and also recommend doctors or hospitals to be visited. Traditional approach of health based applications include analysis of health related data and suggest the necessary remedies. However these applications may not deal with the real time data and also the flu outbreaks at a particular location. However this issues can be addressed in this application. In this paper we shall discuss about the detailed view of the application and how does it address the issues. In our application we deal with twitter tweets for analysis and it is the only social media data source we are using.

The application has two parts. Real Time analysis of data and a recommendation system. The real time analysis runs at the back end and the recommendation algorithm runs on the GUI. Real time analysis of data requires high availability and low latency. This can be achieved only using the new technology STORM and KAFKA. Storm is a stream based real time tool which can be used for machine learning. Kafka is a message queue system which can be used for integration social media content with the storm topology. In storm we will be performing the classification process that will be used to classify the data.

2. Related Work

There are many systems which are health related or weather related analysis. Some of the applications like "sick weather" also make recommendations based on the weather and also the symptoms of diseases. However these applications does not use the real time data for recommendation mechanisms. And most of the systems does not include the real time tools like Storm and Kafka which are highly scalable and have low latency. There are wide range of NoSQL databases which provide fast read and fast write access but the systems accessing this systems are not that feasible. The traditional RDBMS mechanisms cannot used for this kind real time applications as these systems are highly sensitive to ACID properties. So the latency and the data analysis becomes highly impossible with low latency. Here we also need to consider the scalability of the application. Since the application is a native based mobile application. The availability of data and easy access to this data is highly recommendable. These issues can be addressed only with the real time tools like STORM and message brokers like KAFKA.

3. SYSTEM ARCHITECTURE

The below figure depicts the system architecture:

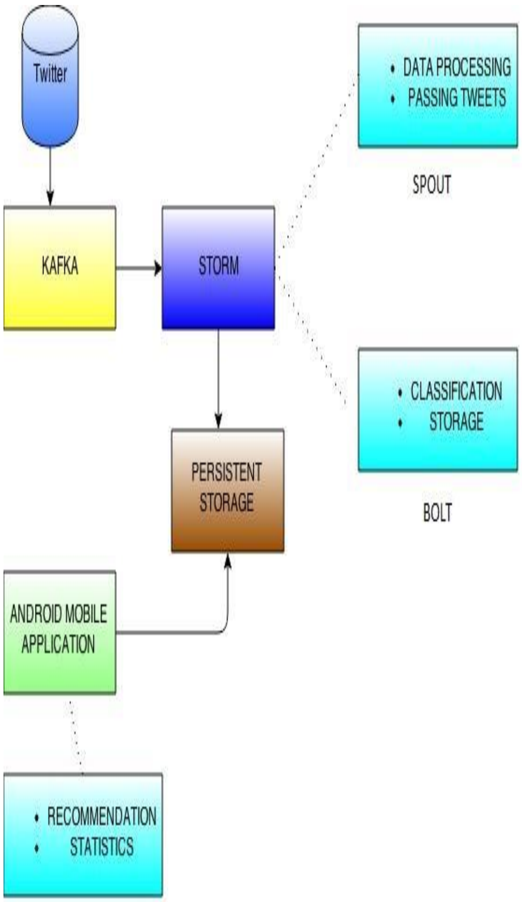

Figure 1 System Architecture

The architecture typically consists of two parts. In the first part we will collect the data from twitter, classify the data and push it to a persistent storage. Now using a mobile application we shall collect the data from persistent storage and analyze the data give a state wise flu outbreaks and their severities. We also make certain recommendations regarding the hospitals to be visited. Kafka acts a message queue and Storm acts as real time analyzing tool for machine learning. So here we collected the data from Kafka from twitter and use this real time data with trained data to analyze the tweets. Here we used Naïve Bayes, D- Tree, SVM methodologies to find the best suitable method based on its analysis and also with respect to the data available.

We have designed the ROC curves and taken the accuracy results to find the optimum classification algorithm.

4. DATA COLLECTION

As mentioned earlier we have two parts of analysis i.e. real time analysis and batch processing for recommendation algorithms. The primary source of data for real time analysis is collected from twitter social media. Tweets are collected from the JSON format and are pruned for classification algorithm. We have collected certain static data from www.data.gov/health for recommendation algorithms. Based on the flu outbreaks we shall analyze the tweets and analyze the static data we have to recommend the users the hospitals or doctors to be visited. The static data consists of the disease, symptoms and the specialized doctors for such diseases. The sample tweet data that is collected is shown below:

{"createdAt":"Sep 20, 2014 12:36:14 PM",

"id":513381131077820416,

"text":"China Seeks Influence in Africa, Pledges Extra

$32 Million to Ebola Fight http://t.co/mlhV2vbIwZ",

"lang":"en",

"statusesCount":218805,

"isGeoEnabled":true,

"isVerified":false,"translator":false,

"listedCount":589,"isFollowRequestSent":false}}

We have collected almost 2GB of static data and has been stored in the persistent storage system.
Coming to the implementation and design aspects, let us look at the machine learning process. Now while talking about Kafka, since it is a message queue we have a producer and a consumer. Here the producer will collect the data from twitter and the consumer event will send the data to storm. So to re-iterate briefly we have twitter apis as producer and storm as consumer in Kafka. Here we neednot store the collected data in any file, as Kafka can store this data for two days without any data loss.

5. DATA PROCESSING AND METHODS

Data processing and the entire application methods has two parts real time stream processing and batch processing. Let us discuss the first part now.

Storm has two parts spout and bolts. The spout will be receiving the data stream from kafka which contains the tweets. Now the tweets which we receive will be in crude format so we need to clean the data and send it for classification.

Cleaning Data: The tweets contains many unnecessary syntactic features. The cleaning is done such that the data confirms WEKA's data format.

The following is removed from the tweet message

- -quotes
- -@ symbol
- -# symbol
- -URL
- -smiles

Additionally duplicates are also removed to reduce unnecessary computational overload.

Once the data cleaning is completed the tweets will be sent to the bolts in the form of tuples. Now in bolts we use certain Weka Java APIs to apply the classification algorithms on the data. We typically used two bolts. In the first bolt we shall receive the tuples and store them in a temporary file. Now these testing file will be given to any classification algorithm as testing data. As mentioned above we have collected certain static data and used it as training data. Here we used three algorithms to measure the accuracy. The three algorithms are:

- SVM
- Decision Tree
- Naïve Bayes

Once the classification is completed we shall store the results in a persistent storage.

Coming to the batch processing which is done in the front end for recommendation algorithms initially we shall extract the data from the persistent storage and store it in the temporary Disk storage of the Android device. The recommendation algorithm will analyze the tweet and identify the disease in the tweet and shall map this data with already existing static data and recommend the hospitals or the doctors to be visited in the nearby location of the user. We used Google Charts to show the statistics of the flu outbreaks on the basis of intensity state wise and overall analysis too. Now we designed a basic recommendation logic inside the android application to recommend the hospitals nearby the location of the user.

6. RESULTS

We have observed significant results for decision tree. But to evaluate the results the trained data could have been much better. As the trained data consisted of many data anomalies we couldn't achieve high accuracy. The other key factor for such accuracy is that the tweet can be positive or negative such discrepancies were not evaluated due to application constraints. The following are the snapshots of the confusion matrices of three different classification algorithms:

Naive Bayes

Results

Correctly Classified Instances	8594	78.0918 %
Incorrectly Classified Instances	2411	21.9082 %
Kappa statistic	0.4335	
Mean absolute error	0.26	
Root mean squared error	0.4446	
Relative absolute error	59.4453 %	
Root relative squared error	95.0654 %	
Total Number of Instances	11005	

Figure 2 Naives Bayes Classification

We have also observed certain ROC curves to depict true positive and true negative values of the data that is being collected and used.

```
Support Vector Machines
-----------------------
Results

Correctly Classified Instances        5178               78.4189 %
Incorrectly Classified Instances      1425               21.5811 %
Kappa statistic                          0.4302
Mean absolute error                      0.2841
Root mean squared error                  0.4243
Relative absolute error                 64.9383 %
Root relative squared error             90.7412 %
Total Number of Instances             6603

=== Confusion Matrix ===

    a    b   <-- classified as
 1218 1626 |    a = yes
  292 5668 |    b = no
```

Figure 3 SVM Classification

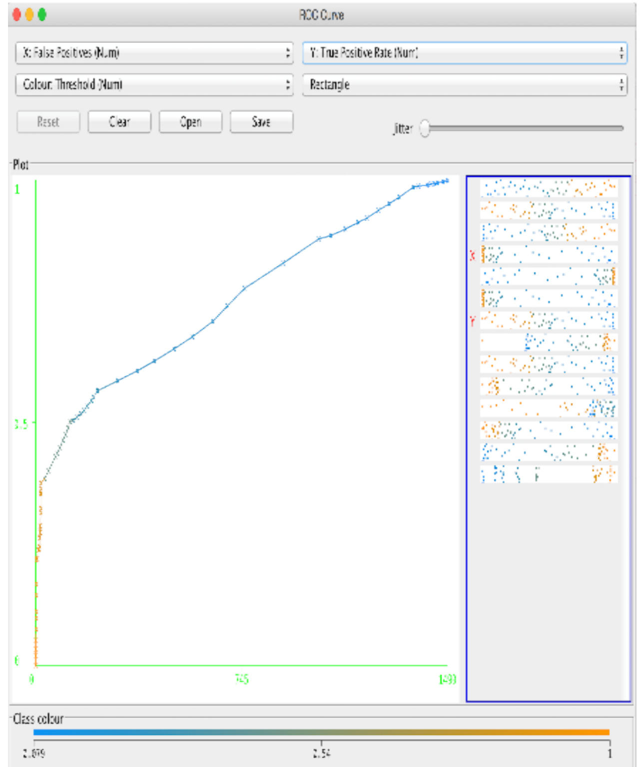

Figure 5 ROC for Naive Bayes classification

```
Decision Tree
------------
Results

Correctly Classified Instances        1733               78.7369 %
Incorrectly Classified Instances       468               21.2631 %
Kappa statistic                          0.4231
Mean absolute error                      0.3162
Root mean squared error                  0.3994
Relative absolute error                 72.2917 %
Root relative squared error             85.3984 %
Total Number of Instances             2201

=== Confusion Matrix ===

    a    b   <-- classified as
  530  892 |    a = yes
   40 2940 |    b = no
```

Figure 4 Decision Tree Classification

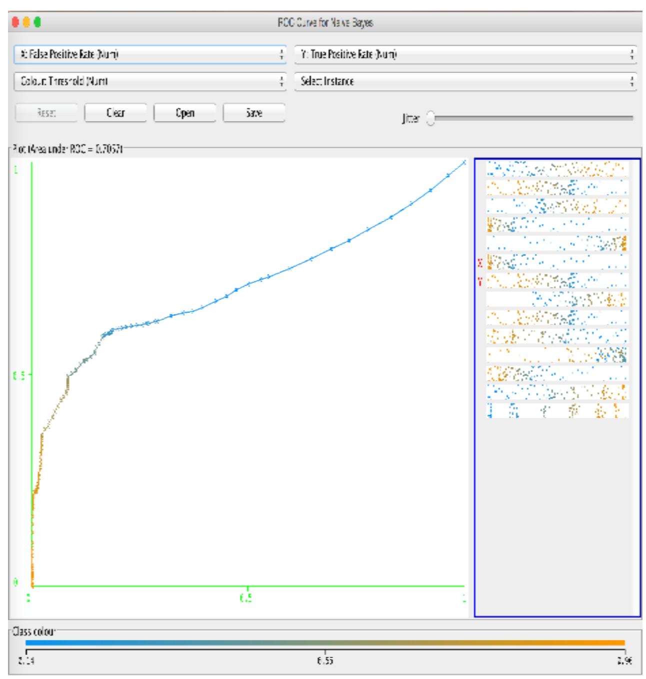

Figure 6ROC curve for Decision Tree classification

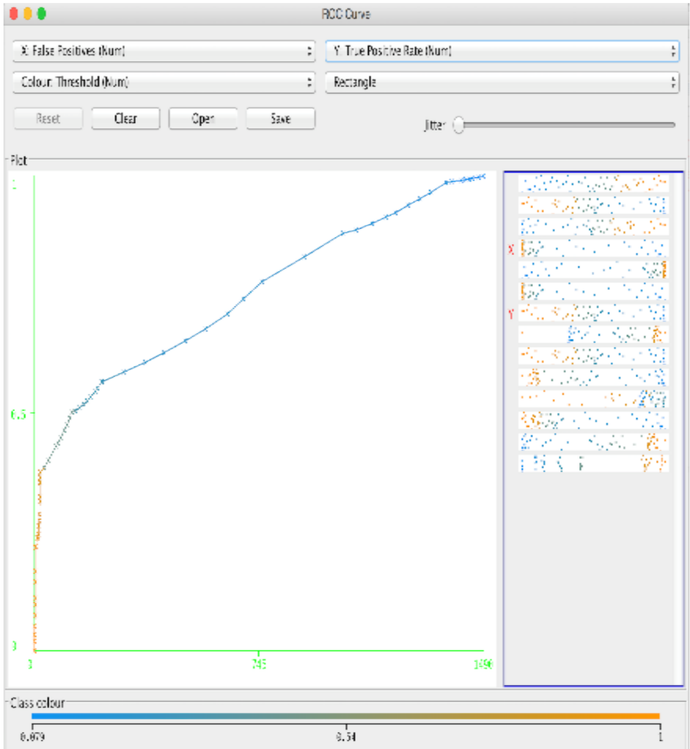

Figure 7ROC curve for SVM classification

7. DATA VISUALIZATION

The initial screen as shown in Figure 8 will consists of a button “Flu Outbreaks” which on clicking will collect data from persistent storage and displays basic information about the flu outbreaks as shown in Figure 9.

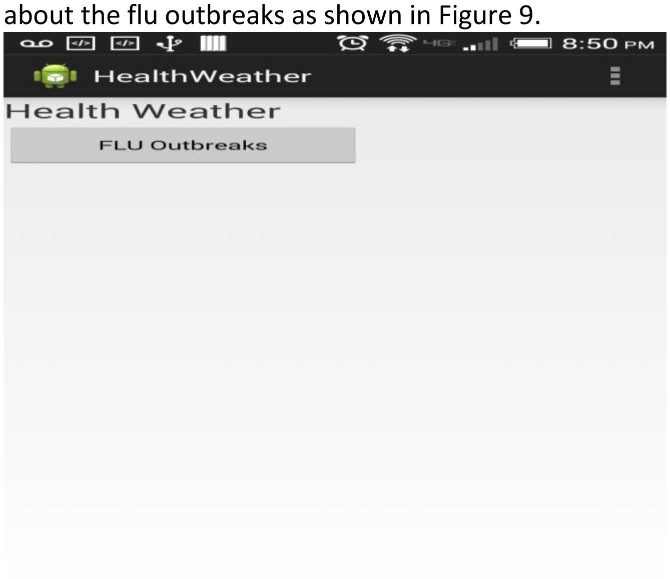

Figure 8 Welcome Page

Figure 9 Tweets and flu outbreaks

Now on choosing a location we get the tweets that were tweeted in that location and also their intensity. Below you can also observe the statistics that were made using the resultant data.

Figure 10 Drop down list showing locations

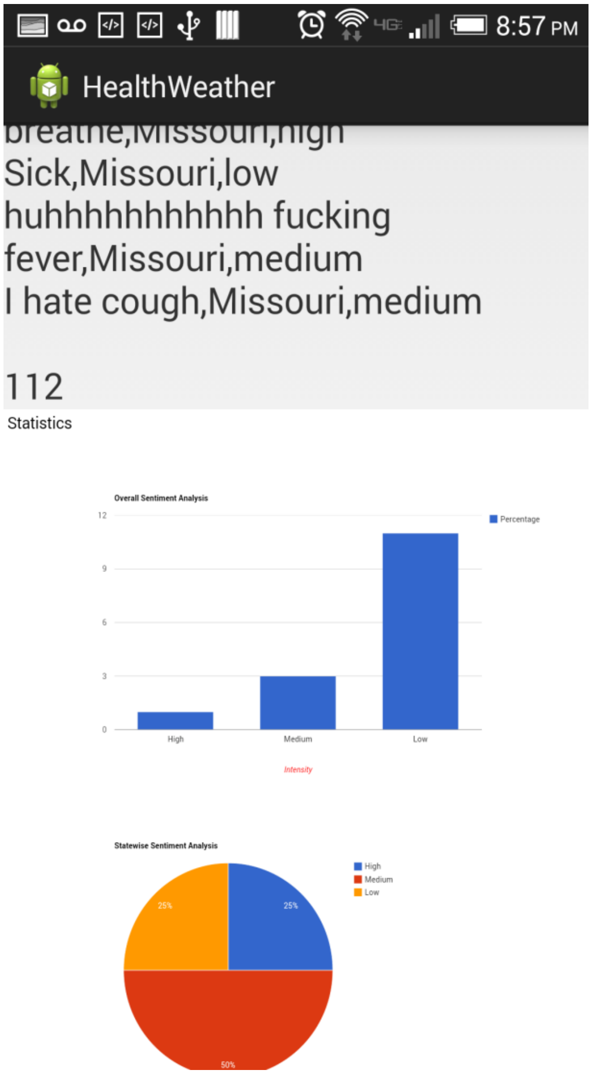

Figure 11 Results

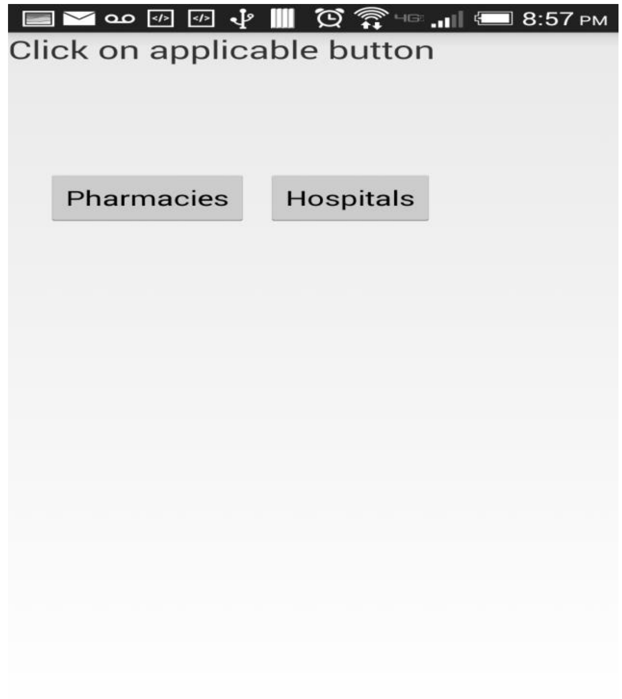

Figure 12Recommendation

In the same page you are provided with a button "View Statistics" which can be used for viewing the statistics in a separate detailed page.

Now when you click on "Immediate Cure" button will ask you to select the nearby pharmacies or hospitals which you can visit. Now on clicking the buttons you will be provided with the nearby hospitals and pharmacies based on your location. We used GPS systems of android to depict the locations. The recommendation algorithm will run when you click any one of the options either the hospitals or pharmacies.

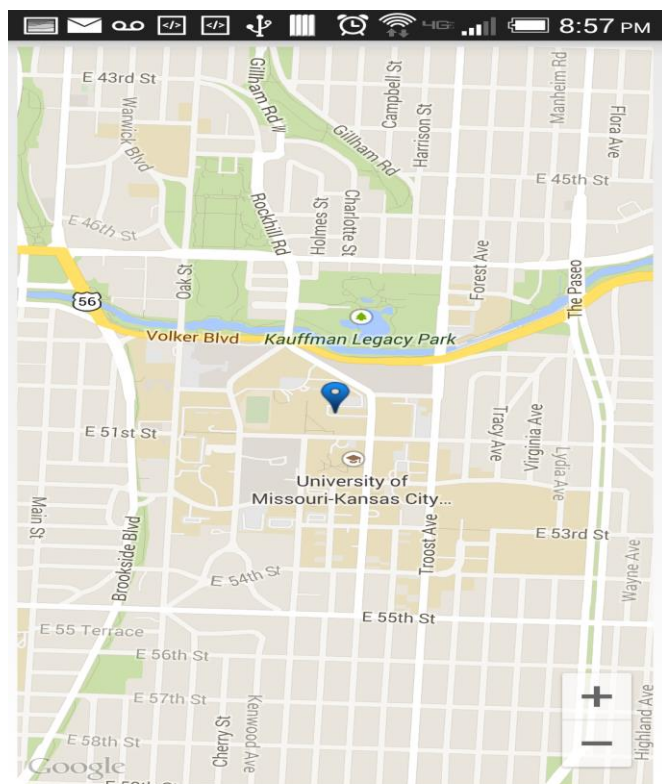

Figure 13 Maps

8. DISCUSSIONS

As we observed in the previous sections, decision tree algorithm has given more accurate results when compared to other classification algorithms. However when we tried doing the evaluation we spent lot of time for the data pruning and formatting part while making the evaluation for the decision tree classification. Since we couldn't observe significant difference in the data accuracy for Naive Bayes and decision tree we decided to use the Decision Tree as our classification algorithm. However the data should be converted into arff format while we use the data for classification. This is the most important and tough tasks in the analysis part. We used certain python scripts to do this. The user registration is one another important aspect which could be included in the system.

9. CONCLUSION

Based on our evaluation process and entire architecture we can say that more improvements can be made in the current application to make it more efficient. More efforts should be kept in data cleaning and data conversion aspects. Since we used weka the formats that are supported will be only arff. The recommendation system can be enhanced to recommend hospitals on the epidemic diseases rather providing general overview of hospitals that can be approached. We can also include certain preventions for each disease in the application based on the climatic conditions and also on the disease. Here we provided a systematic architecture, on how to link the real time analysis with our android applications without putting burden on the mobile devices, by performing entire analysis process in the backend. And also the algorithms, libraries that can be used for the machine learning process. Integration of Kafka with real time data and message queuing of Kafka and storm are the salient features of this application. We sued various web services such as Google Maps, google Charts for data visualization which reduces the code burden.

10. ACKNOWLEDGEMENTS

This project is being done as partial fulfillment of the academic course Real Time and Analytics at University of Missouri Kansas City, under the esteemed guidance of:

- Instructor: Dr.Yugyung Lee
- TA: Feichen Shen

11. REFERENCES

[1] L. Barbosa and J. Feng. Robust sentiment detection on Twitter from biased and noisy data. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010.

[2] M. Bautin, L. Vijayarenu, and S. Skiena. International sentiment analysis for news and blogs. In Proceedings of the International Conference on Weblogs and Social Media (ICWSM), 2008.

[3] V.L. Corpora. Empirical Methods in Natural Language Processing.

[4] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs. In Proceedings of the International Conference on Weblogs and Social Media (ICWSM). Cite seer, 2007.

[5] Biem, H. Feng, A. V. Riabov, D. S. Turaga, Real-time analysis and Management of big time-series data

[6] Ching-Hui Chen, Julien Favre, Gregorij Kurillo, Thomas P. Andriacchi, Ruzena Bajcsy, Rama Chellappa, Camera Networks for Healthcare, Teleimmersion, and Surveillance

[7] Gilad Mishne, Jeff Dalton, Zhenghua Li, Aneesh Sharma, Jimmy Lin, Fast Data in the Era of Big Data: Twitter's Real-Time Related Query Suggestion Architecture, Oct 27. 2012 http://arxiv.org/pdf/1210.7350.pdf

[8] Mining Big Data: Current Status, and Forecast to the Future, Wei Fan Huawei, Albert Bifet

[9] Apache Kafka for Beginners, Gwen Shapira & Jeff Holoman, Sept. 12 2014,

http://blog.cloudera.com/blog/2014/09/apache-kafka-for-beginners/
[10] Storm @Twitter, Ankit Toshniwal, Siddarth Taneja, Amit Shukla, Karthik Ramasamy, Jignesh M. Patel, Sanjeev Kulkarni, Jason Jackson, Krishna Gade, Maosong Fu, Jake Donham, Nikunj Bhagat, Sailesh Mittal, Dmitriy Ryaboy
http://db.cs.berkeley.edu/cs286/papers/storm-sigmod2014.pdf
[11] Sickweather - Sickness Forecasting & Mapping, http://www.sickweather.com/