ABSTRACT

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE

PARANA LIYANAGE, KRISHANI BSC ENG. UNIVERSITY OF MORATUWA, 2005

M.S. CLARK ATLANTA UNIVERSITY, 2011

OUTLIER DETECTION IN SPATIAL DATA USING THE M-SNN ALGORITHM

Committee Chair: Roy George, Ph.D.

Thesis dated July 2013

Outlier detection is an important topic in data analysis because of its applications to numerous domains. Its application to spatial data, and in particular spatial distribution in path distributions, has recently attracted much interest. This recent trend can be seen as a reflection of the massive amounts of spatial data being gathered through mobile devices, sensors and social networks. In this thesis we propose a nearest neighbor distance based method the Modified-Shared Nearest Neighbor outlier detection (m-SNN) developed for outlier detection in spatial domains. We modify the SNN technique for use in outlier detection, and compare our approach with the widely used outlier detection technique, the LOF Algorithm and a base Gaussian approach. It is seen that the m-SNN compares well with the LOF in simple spatial data distributions and outperforms it in more complex

distributions. Experimental results of using buoy data to track the path of a hurricane are also shown.

OUTLIER DETECTION IN SPATIAL DATA USING THE M-SNN ALGORITHM

A THESIS

SUBMITTED TO THE FACULTY OF CLARK ATLANTA UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE MASTER OF SCIENCE

BY

KRISHANI PARANA LIYANAGE

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE

ATLANTA, GEORGIA

July 2013

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to my Research Advisor and the Committee Chair, Dr. Roy George, for his guidance, direction and supervision. I would also like to thank committee members, Dr. Khalil Shujaee and Dr. Peter Molnar for their advice and suggestions. I acknowledge the support given by the faculty and staff of the Department of Computer and Information Science, and especially Research Scientist, Mr. Ali Sazegarnejad. This research is partially supported by research grants from the Army Research Laboratory Grant No: W911NF-12-2-0067 and the Army Research Office Contract Number W911NF-11-1-0168.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

LOF	Local Outlier Factor
SNN	Shared Nearest Neighbor
m- SNN	Modified Shared Nearest Neighbor
SVM	Support Vector Machines
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

CHAPTER 1

1. INTRODUCTION

An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism [2]. Substantial research has been done in outlier detection and these are classified into different types with respect to the detection approach being used. Exemplar techniques include Classification based methods, Nearest Neighbor based methods, Cluster based methods and Statistical based methods [16]. In the Classification-based approach [28, 29] a model is learnt from a set of labeled data points and then a test point is classified into one of the classes using appropriate testing. Support Vector Machine (SVM) based methods [27], methods based on Neural Networks [30] and Bayesian Networks based methods [22, 23, 31] belong to Classification based technique. The testing phase of this method performs fast as each test data is compared against the pre-built model. The accuracy of classification based methods relies on the availability of accurate pre classified examples for different normal classes, which is very rarely found. Nearest Neighbor based methods [24, 26, 32] involve a distance or similarity measure which is defined between data points. The main advantage of the Nearest Neighbor based method is that it does not make any assumptions about the distribution of data. Therefore having an appropriate distance measure helps to apply this method for any type of data sets. The LOF (Local Outlier Factor) method [13] is

a Nearest Neighbor based approach. LOF gives each data point a degree of being an outlier via a relative Density Nearest Neighbor technique. Clustering based methods [19, 20, 21] use the approach of grouping similar data points into clusters. The performance of clustering based techniques depends on the success of clustering algorithm; how accurately it gathers typical data into clusters. According to the basis of Statistical based methods [17, 18, 25] a point is an anomaly, because it is not generated by the stochastic model assumed. Here the normal data points are taken place in high probability regions of the stochastic model, whereas outliers are in the low probability regions of the model [16]. Both parametric [17] and non parametric [25] methods are applied under statistical techniques. Therefore outlier detection using statistical methods is more accurate if the assumptions regarding the underlying data distribution are true. But the assumption that the data is generated from a particular distribution is often not correct.

The technique proposed in this thesis, the m-SNN (modified-Shared Nearest Neighbor) method is based on the non-parametric clustering algorithm Shared Nearest Neighbor (SNN) Approach developed by Ertöz et al. [6]. In contrast to parametric methods this technique does not assume an underlying probability distribution model for the data. m-SNN can also be regarded as a variant of nearest neighbor method. In this method, we consider the ratio between the summation of Euclidean distances to shared nearest neighbors and total number of shared neighbors. To differentiate between outliers and normal points hypothesis testing is used, which is the similar technique used by Babara et al [15] and Rogers [1]. m-SNN does not require any assumption about the data and does not require a threshold for declaring outliers. The number of nearest neighbors

and the confidence level used are the only inputs required by m-SNN. We compare m-SNN approach with LOF method and Gaussian as a baseline parametric method and show that the algorithm presented can be used to detect outliers in distributions with different shapes and different densities. It is seen that the m-SNN compares well with the LOF in standard spatial data distributions and outperforms LOF in complex spatial data distributions.

The outline of the thesis is as follows. Chapter 1 of this work introduces the topic and provides background and related work on outlier detection. Chapter 2 describes related terms and definitions which are used throughout the thesis. Chapter 3 outlines the approach that explains the algorithm behind m-SNN approach. To get a better understanding and to demonstrate the accuracy of m-SNN, several experiments were conducted with different kinds of synthetic data sets those are described in more detail in chapter 4. We apply m-SNN technique to find outliers in real data sets and the initial results are described in chapter 5. Chapter 6 concludes the research with a discussion of the performance, accuracy and the importance of the proposed technique. From the results of experiments, it is clear that this technique gives better results in comparison to LOF and the Gaussian by giving higher true positive and true negative values and very low false positive and false negative values.

CHAPTER 2

2. TERMS AND DEFINITIONS

We define Neighbor Similarity, Density, Neighbor Similarity Distance and Sparseness; then terms local and global outliers. The definitions of Similarity and Density are based on the notions given in [6] and we define the terms p-value, null hypothesis (H_0) and alternative hypothesis (H_a) relating to the proposed technique.

Definition 1: Neighbor Similarity – For a given data point u the neighbor similarity is defined as the number of nearest neighbors being shared between u and its corresponding nearest neighbor. For example, as shown in Figure1 considering only three nearest neighbors, u's nearest neighbors are A, B and C while A's nearest neighbors are B, D and E. Hence, B is shared by both u and A. Therefore the neighbor similarity between u and A is 1.

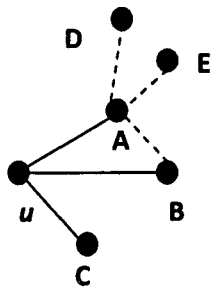

Figure 1: Shared Nearest Neighbors

Definition 2: Density – For a given data point 'u' the density is defined as total number of neighbor similarities between u and its nearest neighbors (vi).

$$density(u) = \sum_{i=1}^{K} NeighborSimilarity(u, v_i)$$

Definition 3: Neighbor Similarity Distance – For a given data point 'u' the neighbor similarity distance is defined as sum of Euclidean distances between u and its all shared neighbors.

Definition 4: Sparseness – For a given data point 'u' the neighbor sparseness is the ratio between Neighbor Similarity Distance and Density.

$$sparseness(u) = \frac{Neighbor\ Similairty\ Distance}{Density}$$

In complex real world situations both global and local outliers may be found [1]. A global metric would be unsuccessful of detecting local outliers. Locally an outlier could be discovered relative to a dense area of points. Conversely, a point with higher sparseness might not be considered an outlier, if it is in a neighborhood of a sparse set of data.

Definition 5: Null and Alternate Hypothesis

The null hypothesis and alternative hypothesis statements for m-SNN method are expressed below.

Null Hypothesis = H_0: Point u is not an outlier (p-value >= τ)

Alternative Hypothesis = H_a: Point u is an outlier (p-value < τ)

p-value is the maximum probability of observing a test statistic as the null hypothesis is true. p-value is also known as observed level of significance while τ is the actual significance level. In [14], the p-value is obtained as the fraction of points in the class that have strangeness greater than or equal to that of the point. According to m-SNN algorithm, p-value of a point is calculated as the fraction of points in the class that have sparseness less than to that of the corresponding point. Therefore larger p-value indicates the high probability of accepting null hypothesis where as smaller p-value implies the high probability of rejecting null hypothesis and accepting the alternative hypothesis.

CHAPTER 3

3. APPROACH

The m-SNN method is based on shared nearest neighbor approach and p-value technique of hypothesis testing for finding outliers. For each data point we calculate its k nearest neighbors by using the Euclidean distance measure. Next, we calculate the Neighbor Similarity of the corresponding data instance i.e., for each data point we calculate the number of neighbors being shared between current node and its nearest neighbors. Subsequently, we calculate the Euclidean distance between current node and shared neighbors. Then the Density of the point is calculated by summing up its all Neighbor similarities. Finally, Sparseness is calculated by taking the ratio between sum of Euclidean distances to shared neighbors and density.

The pseudo-code in Table 1 outlines the algorithm. Here *knn* is to store **k** nearest neighbors for a given data point, and *findkNN* finds the **k** nearest neighbors using Euclidean distance. *knnI* and *knnJ* are the **k** nearest neighbors for $\mathbf{i}^{th}$ data point and its $\mathbf{j}^{th}$ neighbor respectively. For a given data point Euclidean distance to shared neighbors and number of such nodes are stored in temporary variables distance and density respectively. The calculated sparseness is stored in sparseness. **n** is the number of data points in the sample.

As our method needs to find the **k** nearest neighbors for each data point, it is required to calculate the Euclidean distance between each other data points. Hence, since we have '**n**' data points the complexity of calculating Euclidean distance is equal to $O(n^2)$. Finding **k** nearest neighbors for a given data point based on Euclidean distance can be done in a constant time by finding the **k** shortest distanced points to the original point. This does not require sorting all the data points. Finally to find outliers, we need to compare each data point with each other remaining data points, thus resulting $O(n^2)$ complexity.

Table 1: The Modified Shared Nearest Neighbor Algorithm

Procedure: m-SNN Based Outlier Detection
Inputs: data[], a set of data points; k, the number of nearest neighbors; τ, the confidence level
Output: Print Outliers
Assume knn[] stores the k nearest neighbors for the data point, density [],To store shared neighbor density;

```
// Finding k-nearest neighbors for all the data points
for i = 1 to n
   knn[i] = findkNN(data[i])  //Find k-nearest neighbors for data point i and store in the array
end for

//Finding the shared neighbor nodes and distances to them
for i = 1 to n
    distance = 0
    density = 0
    knnI [] = knn[data[i]] // Get neighbors for data point i

    // Find the shared neighbors of data point i
    for j = 1 to k
        knnJ [] = knn[knnI[j]]   // Get neighbors for jth neighbor of data point i
        for x = 1 to k
            for  y = 1 to k
                if knnJ[y] == knnI[x]   // checking for overlapping
                    // Calculate the distance to the overlapping data points
                    distance = distance + euclidean_distance(data[i],   knnJ[y])
                      density = density + 1
                end if
            end for
        end for
    end for
    sparseness[i] = distance/density   // Calculate sparseness for data point i
end for

// Printing outliers
for i = 1 to n
    count = 0
    for j = 1 to n
        if  sparseness [i] >=  sparseness [j] then
              count = count + 1
         end if
        p-val = 1 - (count -1)/n
         if p-val < τ then   // If p-value less than τ, then point i is an outlier
            data[i] is an outlier
        else
            data[i]  is not an outlier
         end if
    end for
end for
```

CHAPTER 4

4. EXPERIMENTING WITH SYNTHETIC DATASETS

This section describes experiments and results with synthetic data sets followed by how the data was generated. We ran the experiments where τ was taken as 0.05. i.e., these experimental results are with 95% confidence.

4.1 Synthetic Data generation

To cover the broad range of applications we generated two main categories of spatial data sets; 1.clusters, 2. Complex spatial paths. In each case we use probabilistic distribution based data generation which takes user inputs to decide parameters of the data pattern. i.e., identify variables and then use a probabilistic model to generate the required number of data points and outliers.

Clusters are the baseline choice of experimentation, and have been the focus of outlier detection algorithms. Path data has recently become very interesting in numerous applications, particularly since location sensing devices have become ubiquitous (mobile devices, etc.). We apply a rigorous set of tests to path data to understand the strength (or weakness) of the method.

After generating data, each set of data points with feature scaling was tested both with proposed outlier detection method and with the LOF technique. Since LOF technique gives a degree of being an outlier of a point, there is no clear cutoff value

differentiating normal points from outliers. For comparison and calculation purposes, we considered a data point with LOF value greater than 2.0 as an outlier. Then the results are also tabulated here.

4.2 Cluster Data Analysis

In our analysis we generated data set with two clusters with 1015 total data points where 15 of them were generated as global outliers. After applying m-SNN technique with tau 0.05, all the expected global outliers were detected and 35 additional points were detected where some of those can be considered as local outliers as shown in Figure 2. LOF approach also was able to detect all above labeled outliers correctly producing all LOF values corresponding to outliers greater than 2.0.

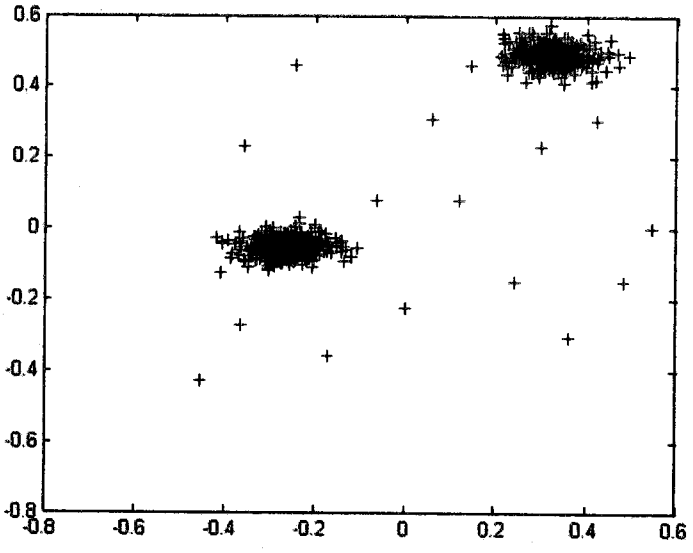

Figure 2: Outlier detection: m-SNN Technique - Two clusters with 1015 data points

4.3 Path Data Analysis

To check how effective our proposed method, we generated data sets with different behaviors. Here we have a set of points that are located on curved paths and some deviated points as well. This set consists with 1000 normal data and 23 significantly deviated points. Figure 3 represents the output results of outlier detection using the proposed method. Generating equivalent results to m-SNN approach, LOF technique also detected 22 outliers with LOF values greater than 2.0.

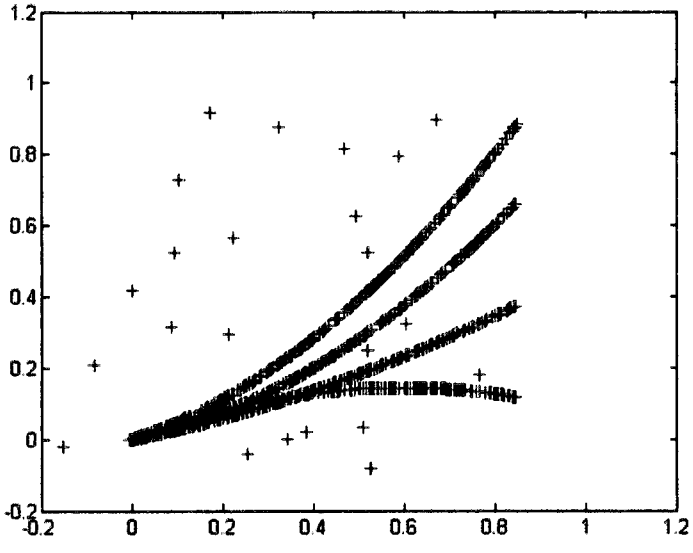

Figure 3: Outlier detection: m-SNN Technique - Four curved paths with 1023 data points

4.4 Spiral Path Data Analysis

A synthetic spiral data with 1010 total data points and including 10 possible outliers, was generated. As shown in Figure 4, m-SNN algorithm could detect 10 of those expected deviated points as outliers and it detected 50 points altogether as outliers. LOF approach detected 8 outliers correctly having Local Outlier Factor greater than 2.0.

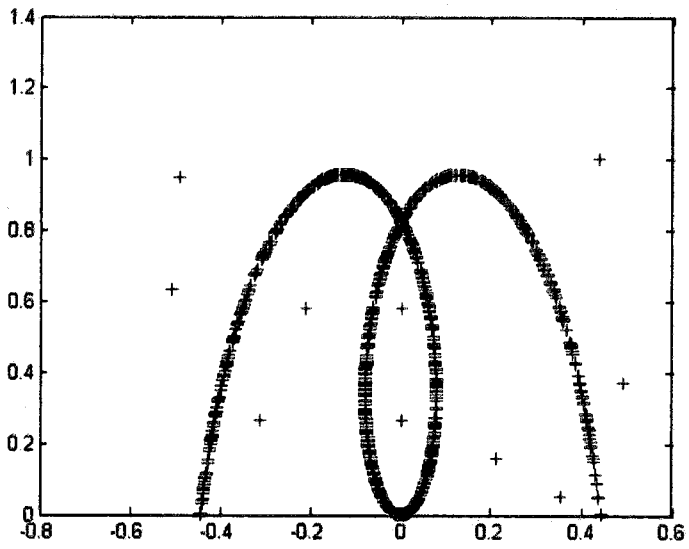

Figure 4: Outlier detection: m-SNN Technique - Spiral path with 1010 data points

4.5 Circular Paths Data Analysis

As the next step, two circled paths were generated which contains total of 1035 data points. This includes 35 points that can be regarded as outliers and 1000 typical data. The

proposed method was successful to detect 51 points as outliers including all 35 expected outlier points which is graphically illustrated in Figure 5. Only 29 points were detected correctly as anomalous data by LOF with minimum LOF of those being 2.0.

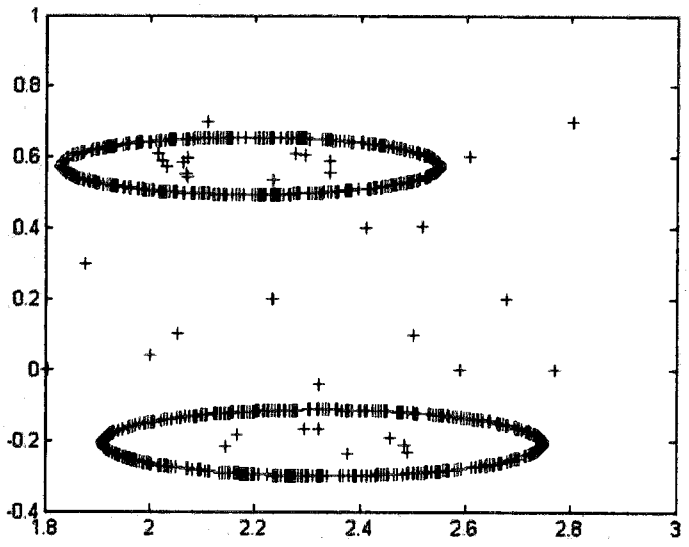

Figure 5: Outlier detection: m-SNN Technique - Two circular paths with 1035 data points

As a control method above data sets were tested with Gaussian approach too. All the results obtained are summarized in Table 2 to demonstrate True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) values as percentages. FP is also known as Type 1 error and FN is also known as Type 2 error in statistics. Moreover in some circumstances TP is entitled Sensitivity and TN is named as Specificity.

Table 2: Summarized results for the experiments

		TP(%)	FP(%)	TN(%)	FN(%)
Clusters	m-SNN	100.0	3.5	96.5	0.0
	LOF	100.0	0.08	99.2	0.0
	Gaussian	7.1	0.0	100.0	92.9
Curved Paths	m-SNN	95.7	2.9	97.1	4.3
	LOF	95.7	0	100	4.3
	Gaussian	30.4	6.1	93.9	69.6
Spiral Path	m-SNN	100.0	4.0	96.0	0.0
	LOF	80.0	0.0	100.0	20.0
	Gaussian	40.0	3.8	96.2	60.0
Circular Paths	m-SNN	100.0	1.6	98.4	0.0
	LOF	82.9	0.0	100.0	17.1
	Gaussian	2.9	0.0	100.0	97.1

According to the tabulated results of Table 2, it is clear that m-SNN has very high TP, TN percentages and very low FP, FN percentages. Therefore m-SNN is enhanced in accuracy and performance when detecting outliers comparing to Gaussian approach. Also the proposed method performs at least as equivalent to LOF approach. In particular comparing the values corresponding to non clustering data sets it is evident that m-SNN is more robust when finding outliers. This shows that Gaussian approach failed to find

anomalies correctly as it gave low TPs and high FNs. m-SNN can be successfully used to detect outliers when the data distribution model is not known exactly. In addition to that, Table 2 shows that m-SNN is stronger in anomaly detection of datasets having arbitrary shapes and densities.

CHAPTER 5

5. EXPERIMENTING WITH REAL DATASETS

In this section we discuss experiments and results of applying m-SNN technique to real data sets. For this experimenting purpose, we chose data recorded from buoys located in Gulf of Mexico. Description of datasets, Experiments and results are presented below.

5.1 Description of Dataset

There are many buoys located in Gulf of Mexico area and data recorded from those buoys are used for several purposes including weather forecasts, marine forecasts and climate predictions. The buoys record data by making a number of routine measurements such as wind direction, wind speed, wave height, barometric pressure, air temperature, sea surface temperature and dew point temperature.

For experimenting with real datasets and detecting outliers, we chose 17 datasets, which contain hourly basis weather data from 17 buoys located at specific geographic locations in the Gulf of Mexico during year 2005. Figure 6 shows the locations of 17 buoys that we considered for empirical evaluations. From original data sets [34], we selected five features; wind direction, wind speed, barometric pressure, air temperature and water temperature at each hour.

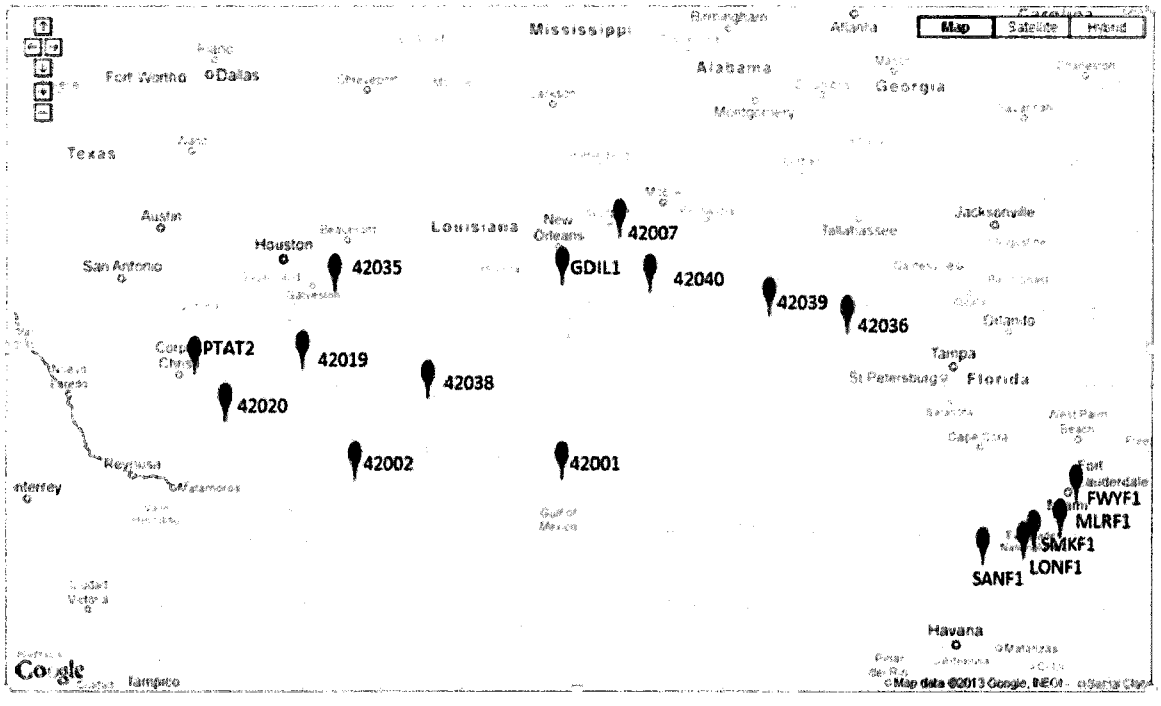

Figure 6: Buoy Locations

5.2 Experiments

We conducted experiments to detect outliers in two different ways i.e. detecting outliers at specific buoy location and detecting outliers at specific time.

5.2.1 Detecting outliers at specific locations

To find outliers which occurred at each buoy location, first each data set was tested with m-SNN algorithm to detect outliers, taking tau as 0.01. Then the detected outliers are analyzed and grouped into time durations when that have been occurred. These results clearly show that outliers are associated to time periods when major hurricanes (Katrina, Rita, Wilma) occurred. The time periods when outliers appeared

differ from one buoy location to another, according to the hurricane track. Therefore the occurrence of outliers correlated with actual track's of hurricanes. Table 3 shows basic details of buoy datasets and how many outliers were detected at each buoy location. While Table 4 shows detected major outlier time intervals resulted from hurricane Katrina accordance with corresponding buoys, Table 5 includes detected major outlier time periods resulted from hurricane Rita and those corresponding buoys.

Table 3: Summarized results for the experiments with Buoy Data

Buoy ID	Geographic Location		No. of instances	No. of detected outliers
	Latitude	Longitude		
42001	26.0 N	90.0 W	8741	87
42002	26.0 N	94.0 W	8729	87
42007	30.1 N	88.9 W	6705	67
42019	27.9 N	95.0 W	8676	86
42020	27.0 N	96.5 W	8685	86
42035	29.2 N	94.4 W	8738	87
42036	28.5 N	84.5 W	8346	83
42038	27.4 N	92.6 W	8095	80
42039	28.8 N	86.0 W	5091	50
42040	29.2 N	88.3 W	8251	82
FWYF1	25.6 N	80.1 W	8153	81
GDIL1	29.3 N	90.0 W	6264	62
LONF1	24.8 N	80.9 W	8750	87
MLRF1	25.0 N	80.4 W	8313	83
PTAT2	27.8 N	97.1 W	8746	87
SANF1	24.5 N	81.9 W	6295	62
SMKF1	24.6 N	81.1 W	6543	65

Table 4: Outlier buoys and time periods resulting from Katrina

Buoy ID	Major outlier time periods resulted from hurricane Katrina
42001	08/28 10:00 to 08/29 12:00
42035	08/28 20:00 to 08/29/06:00
42040	08/29/04:00 to 08/30/03:00
FWYF1	08/25 16:00 to 08/26 11:00
GDIL1	08/28/23:00 to 08/29/12:00
LONF1	08/26 02:00 to 08/26 15:00
MLRF1	08/26 01:00 to 08/26 15:00
SANF1	08/26/05:00 to 08/26/23:00

Table 5: Outlier buoys and time periods resulting from Rita

Buoy ID	Major outlier time periods resulted from hurricane Rita
42001	09/22 07:00 to 09/23 15:00
42002	09/23/ 13:00 to 09/24 00:00
42019	09/23 17:00 to 09/24 14:00
42038	09/27 19:00 to 09/28 17:00
LONF1	09/20 08:00 to 09/20 17:00
MLRF1	08/26 01:00 to 08/26 15:00
PTAT2	09/24/15:00 to 09/24 19:00
SANF1	09/20/13:00 to 09/20/22:00 (No data available after 09/20/22:00)
SMKF1	09/20 12:00 to 09/20 17:00

5.2.2 Detecting outliers at specific time

To capture the outliers among 17 buoys at specific time, first we selected data corresponding to that specific time from each and every buoy. Then a new dataset was created by adding selected data into one set. Next new dataset was tested with m-SNN technique to detect outliers. This procedure was conducted to each new dataset corresponding to specific times. During this experiment tau was taken as 0.1, which means all the results obtained here is with 90% confidence level. We used m-SNN algorithm to detect outliers from August 26, 2005 at 2:00 am until August 29, 2005 at 2:00 pm on each 12 hours basis. Figure 7 shows the actual path of the Katrina through the Gulf of Mexico between (8/26/2005 and 8/29/2005), and Figures 8-15 show the hurricanes position juxtaposed against outlier data from the buoys during the same time period.

Figure 7: Path of Hurricane Katrina (8/26/2005 – 8/29/2005)

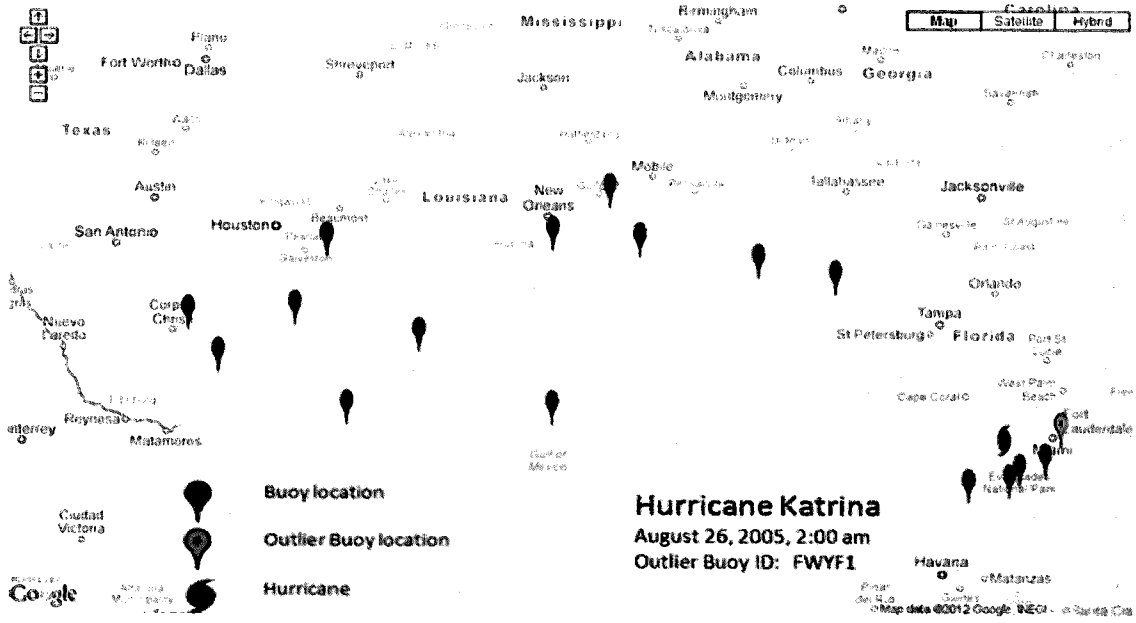

Figure 8: Outliers and normal Buoy locations on August 26, 2005 at 2:00 am

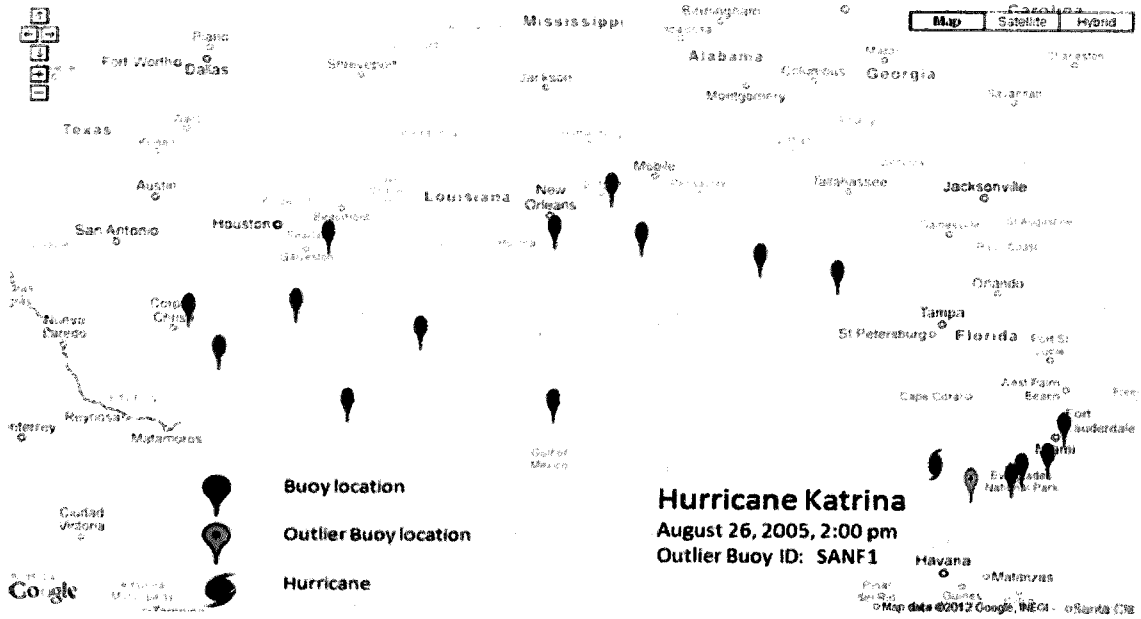

Figure 9: Outliers and normal Buoy locations on August 26, 2005 at 2:00 pm

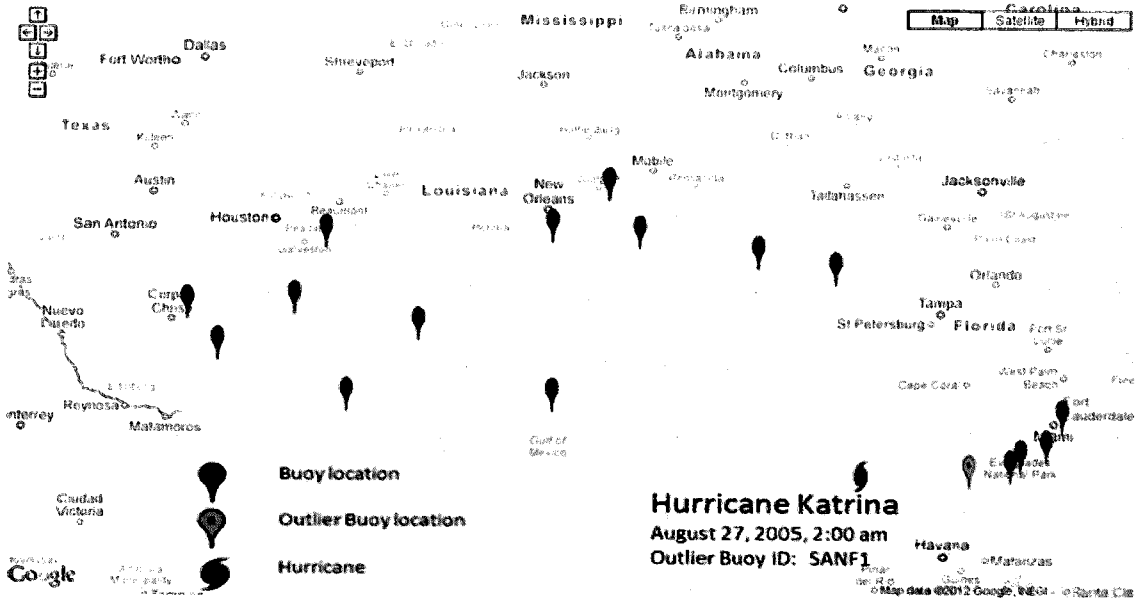

Figure 10: Outliers and normal Buoy locations on August 27, 2005 at 2:00 am

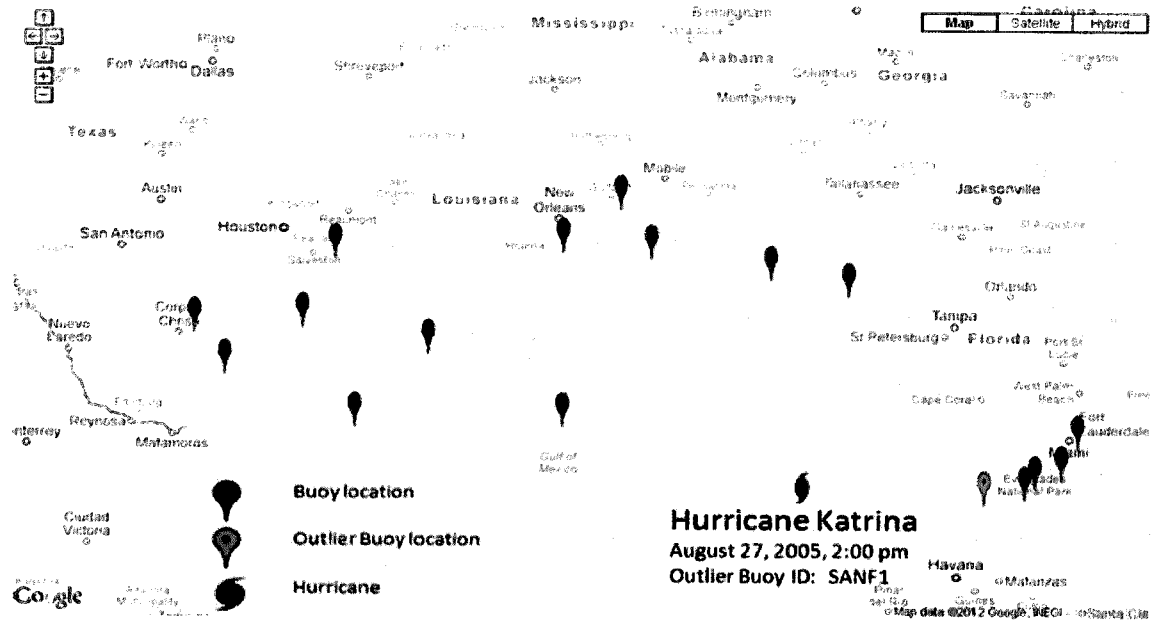

Figure 11: Outliers and normal Buoy locations on August 27, 2005 at 2:00 pm

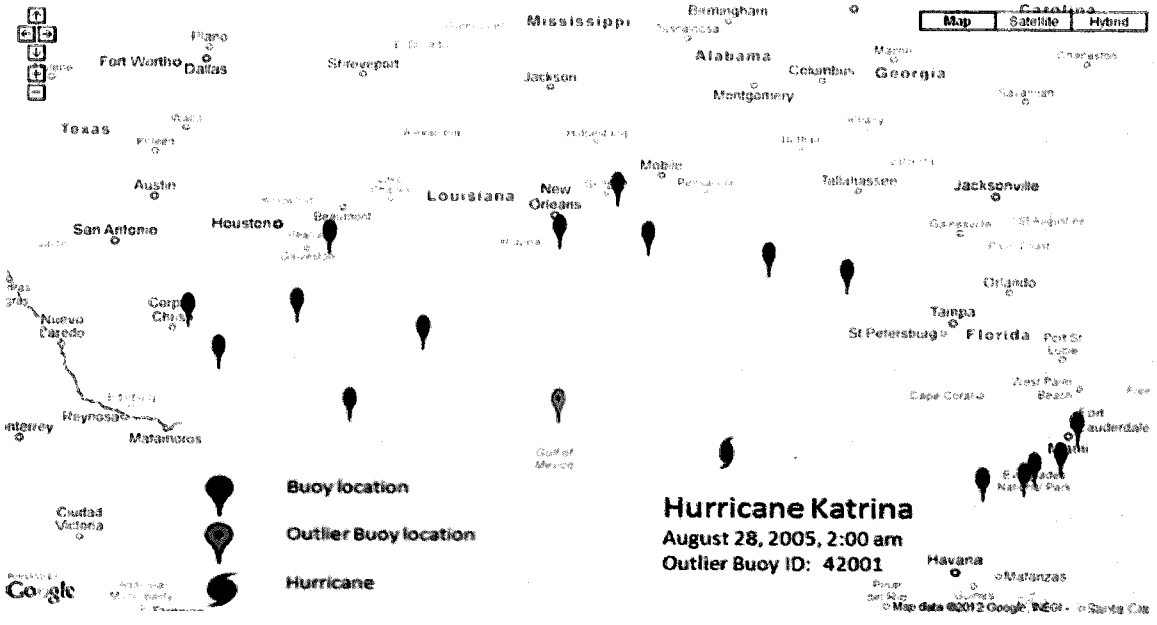

Figure 12: Outliers and normal Buoy locations on August 28, 2005 at 2:00 am

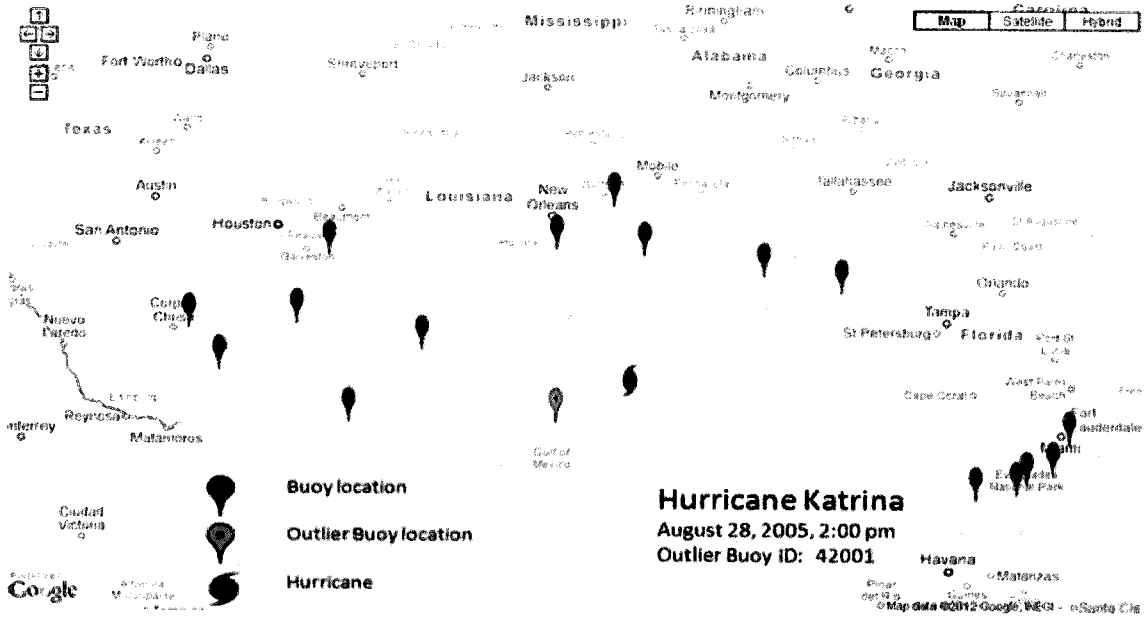

Figure 13: Outliers and normal Buoy locations on August 28, 2005 at 2:00 pm

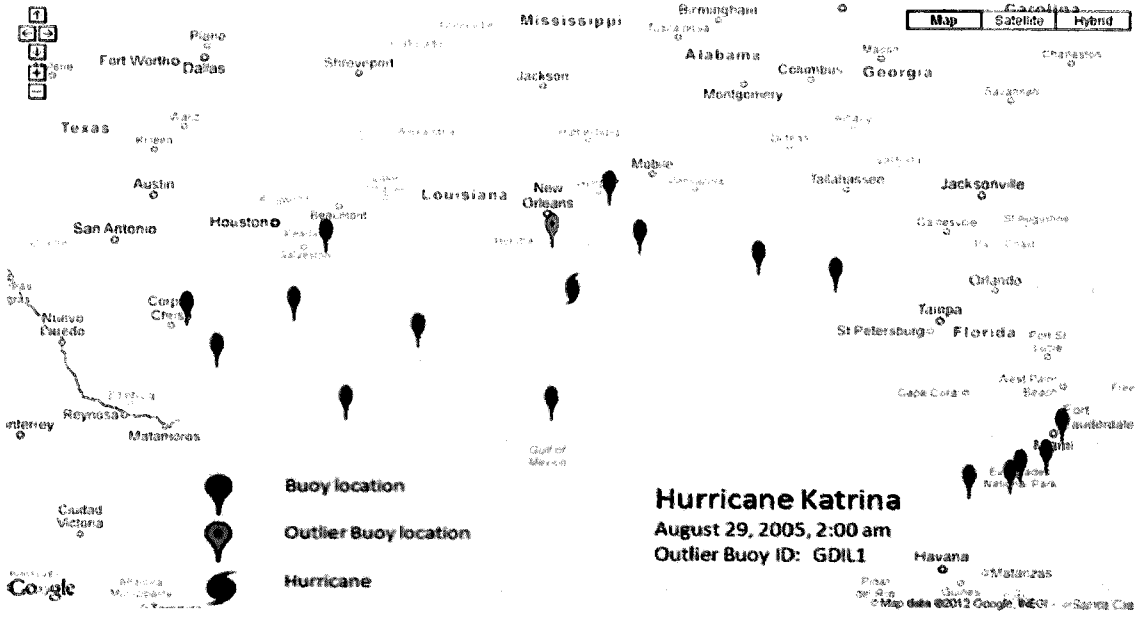

Figure 14: Outliers and normal Buoy locations on August 29, 2005 at 2:00 am

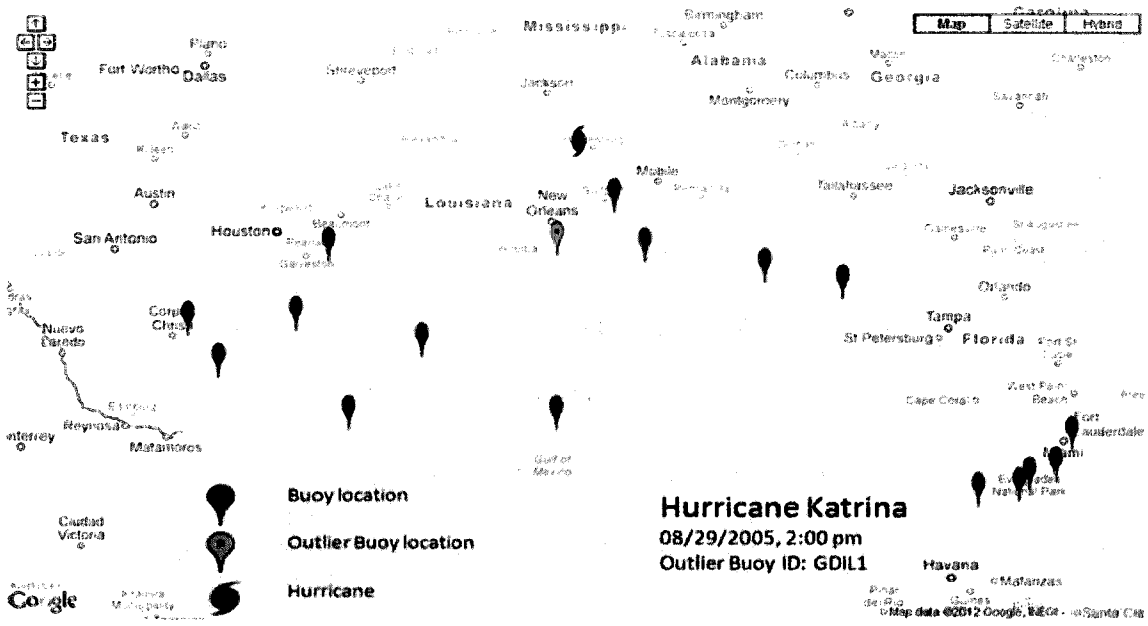

Figure 15: Outliers and normal Buoy locations on August 29, 2005 at 2:00 pm

By considering the results obtained for above two kinds of experiments on real data sets, it is clear that m-SNN has successfully captured outliers resulting from major hurricanes both at each buoy locations and at specific times. It is seen that the technique can accurately capture outliers from key events, such as Hurricane Katrina that occurred during this time period. Further, the outliers may track the path of the hurricanes. Therefore m-SNN is strongly capable of detecting outliers in actual data sets also.

CHAPTER 6

6. CONCLUSIONS AND FUTURE WORK

In this thesis, we have described an algorithm m-SNN which is capable of detecting outliers in different types of spatial data sets. This proposed approach is a combination of Shared Nearest Neighbor and distance based methods that avoids assumptions about data distributions and uses hypothesis testing to detect outliers.

First we compared the proposed technique with LOF and also with a baseline Gaussian approach on several synthetic data sets that containing different patterns of data distributions in two dimension environment. Through experimentations, we have shown the method achieve good results with k as 15 and 95% confidence level. The proposed m-SNN technique results very high true positive and true negative values as well as very low false positive and false negative values. According to the experimentation results this technique provides good results on a variety of synthetic datasets when detecting both global and local outliers. In addition, the m-SNN approach produces outlier detection results equivalent or better than other two comparative methods.

Then the empirical evaluations were done using a high dimensional oceanographic real dataset. According to the results we obtained from running m-SNN algorithm on these buoy data sets, it is evident that m-SNN is a robust method for detecting outliers in high dimensional spatial-temporal real data.

Currently we are reformulating the algorithm to improve the run time efficiencies and also to parallelize the code to make it amenable to massively large data sets. Also to reduce the time complexity of the m- SNN algorithm, our next step is to use of k-D tree structure for m-SNN. Further we are planning to investigate on the detection of outliers in streaming spatial data. Moreover we have a future plan on continuing our experimentations with more real datasets.

REFERENCES

1. Rogers, J.P. Detection of Outliers in Spatial-temporal Data, PhD Thesis.
2. Hawkins, D. Identification of Outliers, Chapman and Hall, London, 1980.
3. Johnson, T.; Kwok, I.; Ng, R. Fast Computation of 2- Dimensional Depth Contours, Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining, New York, NY, AAAI Press, 1998, pp. 224-228.
4. Knorr, E.M.; Ng, R.T. Algorithms for Mining Distance- Based Outliers in Large Datasets, Proc. 24th Int. Conf. on Very Large Data Bases, New York, NY, 1998, pp. 293-298.
5. Knorr, E. M.; Ng R. T. Finding Intensional Knowledge of Distance-based Outliers, Proc. 25th Int. Conf. on Very Large Data Bases, Edinburgh, Scotland, 1999, pp. 211-222.
6. Ertöz, L.; Steinbach, M.; Kumar,V. A New Shared Nearest Neighbor Clustering Algorithm and its Applications.
7. Wang, W.; Yang, J.; Muntz, R. STING: A Statistical Information Grid Approach to Spatial Data Mining, Proc. 23th Int. Conf. on Very Large Data Bases, Athens, Greece, Morgan Kaufmann Publishers, San Francisco, CA, 1997, pp. 186-195.

8. Zhang, T.; Ramakrishnan, R.; Linvy, M. "BIRCH: An Efficient Data Clustering Method for Very Large Databases", Proc. ACM SIGMOD Int. Conf. on Management of Data, ACM Press, New York, 1996, pp.103-114.

9. Angiulli, F.; Pizzuti, C. Outlier mining in large high-dimensional data sets. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(2): 203-215.

10. Gammerman, A.; and Vovk, V. Prediction algorithms and confidence measures based on algorithmic randomness theory. Theoretical Computer Science. 2002, 287: 209-217.

11. Liyanage, K.P.; George, R.; Shujaee, K. Outlier Detection in Spatial Data using the m-SNN Algorithm, IEEE SouthEast Conference 2013, Jacksonville, FL, April 2013

12. Vapnik. V. Statistical Learning Theory, New York: Wiley, 1998.

13. Breunig, M.; Kriegel, H.; Ng, R.; Sander, J. LOF: Identifying Density-Based Local Outliers. Proc. of the ACM SIGMOD Conference on Management of Data, 2000, 427- 438.

14. Proedru, K.; Nouretdinov, I.; Vovk, V.; Gammerman, A. Transductive confidence machine for pattern recognition. Proc. 13th European conference on Machine Learning. 2002, 2430:381-390.

15. Barbara, D.; Domeniconi, C.; Rogers, J.P. Detecting Outliers using Transduction and Statistical Testing, KDD'06, Philadelphia, Pennsylvania, 2006

16. Velegrakis, D. Outlier Detection over Data Streams using Statistical Modeling and Density Neighborhoods, Masters Thesis

17. Eskin, E.; Anomaly detection over noisy data using learned probability distributions. Pages 255-262. Morgan Kaufmann, 2000.

18. Eskin, E.; Lee, W.; Stolfo, S. J. Modeling system calls for intrusion detection with dynamic window sizes. In In proceedings of DARPA information Survivability Conference and Exposition 2, DISCEX, 2001.

19. Ester, M.; Kriegel, H. P.; Sander, J.; Xu, X. A density based algorithm for discovering clusters in large spatial databases with noise. In Proc. Of 2nd International Conference on Knowledge Discovery and Data Mining, KDD-96, pages 226-231, 1996.

20. Ertoz, L.; Steinbach, M.; Kumar, V. Finding topics in collections of documents: A shared nearest neighbor approach. In Workshop on Text Mining, held in conjunction with the First SIAM International Conference on Data Mining, SDM 2001. Society for Industrial and Applied Mathematics, 2003.

21. Guha, S.; Rastogi, R.; shim, K. Rock: A robust clustering algorithm for categorical attributes. Inf. Syst., 25(5): 345-366, 2000.

22. Barbara, D.; Wu, N.; Jajodia, S. Detecting novel network intrusions using bayes estimators. In Proceedings of the First SIAM Conference on Data Mining, April 2001.

23. Bronstein, R.; Das, J.; Duro, M.; Friedrich, R.; Kleyner, G.; Muller, M.; Singhal, S.; Cohen, I. Self-aware services: Using Bayesian networks for detecting anomalies in internet-based services. In Northwestern University and Stanford University Gary Igor, Pages 623-638, 2001.

24. Knorr, E. M.; Raymond, T.; Ng, Tucakov, V. Distance-based outliers: algorithms and applications. The VLDB Journal, 8(3-4): 237-253, 2000.

25. Markou, M.; Singh, S. Novelty detection: A review – part 1: Statistical approaches. Signal Processing, 83: 2003, 2003.

26. Otey, M. E.; Ghoting, A.; Parthasarathy, S. Fast distributed outlier detection in mixed-attribute data sets. Data Min. Knoewl. Discov., 12(2-3): 203-228, 2006.

27. Ratsch, G.; Mika, S.; Schkopf, B.; Muller, K. R. Constructing boosting algorithms for svms: an application to one-class classification, 2002.

28. Roth, V. Kernal fisher discriminants for outlier detection. Neural Computing, 18(4): 942-960, 2006.

29. Schlkopf, B.; Platt, J.C.; Taylor, J.C.S.; Smola, A. J.; Williamson, R. C.; Estimating the support of a high-dimensional distribution. Neural Computation, 13(7): 1443-1471, 2001.

30. Williams, G.; Baxter, R.; He, H.; Hawkins, S.; Gu, L. Comparative study of rnn for outlier detection in data mining. In ICDM, page 709, 2002.

31. Sebyala, A. A.; Olukemi, T.; Sacks, L. Active platform security through intrusion detection using naïve Bayesian network for anomaly detection. In London Communications Symposium, 2002.

32. Zhang, J.; Wang, H. Detecting outlying subspaces for high-dimensional data: the new task, algorithms and performance. Knowl. Inf. Syst., 10 (3): 333-355, 2006.

33. McDiarmid, A.; Bell, S.; Irvine, J.; Banford, J.; Nodobo: "Detailed Mobile Phone Usage Dataset".

34. National Data Buoy Center, Data Availability Summary for NDBC Platforms, http://www.ndbc.noaa.gov/data_availability/data_avail.php