Outlier Detection in Spatio-Temporal Data using Data Mining Technique

Submitted By

Sandeep Joshi

13MCEC16

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Outlier Detection in Spatio-Temporal Data using Data Mining Technique

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Sandeep Joshi

(13MCEC16)

Guided By

Prof. K. P. Agrawal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Certificate

This is to certify that the major project entitled **"Outlier Detection in Spatio Temporal Data using Data Mining Technique"** submitted by **Sandeep Joshi (Roll No: 13MCEC16)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. K. P. Agrawal
Guide & Associate Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Vijay Ukani
Associate Professor,
Coordinator M.Tech - CSE,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. K. Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad.

Statement of Originality

I, **Sandeep Joshi**, Roll. No. **13MCEC16**, give undertaking that the Major Project entitled **"Outlier Detection in Spatio Temporal Data using Data Mining Technique"** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. K. P. Agrawal

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. K. P. Agrawal**, Associate Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. K. Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Sandeep Joshi

13MCEC16

Abstract

Outlier detection is very important data mining method for various applications. The definition of outlier is that "an outlier is an observation that significantly deviates from other observations". Many outlier detection techniques have been developed for certain application domains, while others are developed as more generic. Outlier detection in spatio-temporal data are more challenging compared to identifying outliers in classical data. Spatio-temporal data that relate to both space and time. Spatio-temporal data mining refers to the process of discovering patterns and knowledge from spatio-temporal data. Spatio-temporal data are dense and highly correlated in nature. Here we are concentrating on outlier detection in spatio-temporal earth observation data. After exhaustive literature survey we have identified various techniques and their advantages and disadvantages to detect outlier. Most of the techniques are focuses only on spatial aspect of the data but not the temporal aspect. These techniques were compared and based on the parameters like ability to detect arbitrary shaped cluster, time complexity, high dimensionality etc., we selected DBSCAN, ST-DBSCAN, ST-OUTLIER and SNN approaches. Various input parameters were studied which is taken by these algorithms. Implementation of DBSCAN, ST-DBSCAN and our proposed approach ST-SNN is done using R Package (Open Source) and results are displayed using QGIS tool.

Abbreviations

DBSCAN	Density-based spatial clustering of applications with noise.
OPTICS	Ordering Points To Identify the Clustering Structure.
SNN	Shared Nearest Neighbour.
CURE	Clustering using Representative.
ROCK	RObust Clustering using linKs.
EM	Expectation Maximization.
DENCLUE	DENsity-based CLUstEring.
STING	Statistical Information Grid approach.
MODIS	Moderate Resolution Imaging Spectroradiometer.
NDVI	Normalized Difference Vegetation Index.
ST-DBSCAN	Spatio-Temporal DBSCAN.
ST-SNN	Spatio-Temporal Shared Nearest Neighbour.
ST-OUTLIER	Spatio-Temporal OUTLIER.
QGIS	Quantum geographic information systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Spatial as well as spatio-temporal databases are growing very quickly, increasing the need for effective and efficient analysis methods to mining the information contained in the data. Most of the knowledge discovery in databases (KDD) process is focus on the finding common patterns. However, for various applications like credit card fraud, discovery of criminal activities in e-commerce, weather prediction etc. require finding outliers (rare events), which are deviated from remaining other observations. Identifying outliers is an important area of research in the field of data mining for variety of applications. Outlier detection has been studied in various data domains. The different data domains in outlier analysis require dedicated techniques of different types.

Recently, some studies have been proposed on outlier detection [1][2][3] on spatial datasets. Most of these studies have been not considered temporal dimension. This research presents a new clustering based spatio-temporal outlier detection technique, which is based on the shared nearest neighbour concept to cluster the data objects. Clustering algorithms like ROCK [4], DBSCAN [5], and CURE [6] can also handle outliers, but their main concern is to find clusters, and noise points are represented as outliers. Number of noise formed by any clustering algorithm is dependent on a particular algorithm and also on its input parameters.

1.1 Research Objective

The objective of this research work is to identify outliers in spatio-temporal earth observation data, where data are highly correlated, having different size, densities and dimensionality. In the first step of outlier detection, clustering is performed on the spatio-

temporal dataset with our proposed ST-SNN clustering approach, which is capable to identify arbitrary shaped cluster. In the second step of outlier detection, clustering results from the first step is used to identify spatial outliers and finally in the third step, to find presence of outliers in our dataset, identified spatial outliers in second step are compared with temporal neighbour.

1.2 Applications

Applications of outlier detection [7]:

- **Intrusion Detection Systems:**

  In networked computer systems, data are collected about system calls, network traffic etc., this data may show unusual result because of malicious activities. Identification of such activity is referred to as intrusion detection.

- **Credit Card Fraud:**

  Unauthorized use of credit card may show different patterns, such as a buying extravaganza from geographically obscure locations. Such patterns can be outliers in credit card data.

- **Medical Diagnosis:**

  In medical applications data is gathered from a devices such as MRI scans, PET scans or ECG. Unusual patterns in such data may reflect disease conditions.

- **Earth Science Applications:**

  In weather, climate, or vegetation cover applications, we can use outlier detection where anomalous regions are detected in spatial data. Therefore, many of these applications are spatial or spatio-temporal in nature. For example, sea surface temperatures are tracked to determine anomalous weather patterns.

- **Web Log Analytics:**

  Web logs often contain significant information about security holes and other anomalous activity.

Chapter 2

Literature Survey

2.1 Knowledge Discovery Process

Data mining is the process of discovering interesting patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, theWeb, other information repositories, or data that are streamed into the system dynamically.

Knowledge Discovery Process Steps [8]:

- Cleaning the data.
- Integrate data.
- Select data.
- Transformation of data.
- Data mining.
- Pattern finding.

Spatio-temporal data mining is the discovery of interesting spatial patterns from data over time using data mining techniques on spatially and temporally distributed data. Spatial location such as longitude, latitude are define by **spatial attributes**, where as non-spatial features of objects such as wave height, length are defined by **non-spatial attributes**.

An outlier is a data point which is significantly different from the remaining data. ST-Outliers are spatio-temporal objects whose behavioural (non-spatial and non-temporal) attributes are significantly different from those of the other objects in its spatial and temporal neighbourhoods [9].

2.2 Different Aspects of Outlier Detection problem

- **Input Data Type[9]:**

  Collection of data objects is given as input. Data objects are described using a set of features. The features can be of different types such as continuous, binary, or categorical. Each data object may be of only one features or multiple features.

- **Types of Outliers:**

  - Point Outliers.
  - Collective Outliers.

- **Data Labels[9]:**

  The labels associated with a data instance shows whether that data instance is normal or outlier. Based on the availability of the labels, outlier detection techniques can operate in one of the following modes:

  - Supervised outlier detection.
  - Semi-Supervised outlier detection.
  - Unsupervised outlier detection.

- **Outlier Detection output format[10]:**

  outlier detection techniques outputs are either:

  - **Scores:** Assign an outlier score to each data object, OR
  - **Labels:** Assign a label or mark (outlier or normal) to each data object.

2.3 Classification Based Outlier Detection Techniques

Classification is used to learn a model from a set of labelled data instances and then, classify a test instance into one of the classes using the learned model.

Outlier identification approaches that use different algorithms for classification[11]:

- Neural Network Based.
- Bayesian Network Based.
- Support Vector Machine Based [12].
- Rule Based.

2.4 Clustering Based Outlier Detection Techniques

Clustering is used to group similar kind of data instances into clusters. Even though clustering and outlier detection appear to be fundamentally different from each other but several clustering-based outlier detection techniques have been developed with the assumption that outlier points are not belong to any cluster [13].
Disadvantage of such type of techniques are, they are not optimized to identify outliers, because the main aim of the clustering is to find clusters not the outliers.

2.5 Statistical Outlier Detection Techniques

Statistical outlier detection techniques assuming that normal data objects occur in high probability regions of a model, while outlier objects occur in the low probability regions of the model.

Statistical techniques fits a statistical model to the given dataset and then determine that is there any unusual object belongs to this model or not by applying statistical inference test. Objects that have a low probability in that learned model are declared as outlier objects.

2.6 Proximity Based Outlier Detection

The proximity of a data point may be defined in a variety of ways, which are different from one another, but are similar enough to merit a unified treatment within a single chapter.

The most common ways of defining proximity for outlier analysis are as follows [13]:

- Cluster based.
- Distance based.
- Density based.

2.7 Existing Techniques

Existing techniques shown in table 2.1 :[10][7][14][11][15][12]

Method	Advantages & Disadvantages	Assumption	Source
Depth Based	(+)efficient in 2D/3D Spaces. (-)Inefficient for large Dataset.	Normal Objects are in center. outliers are located at border.	IEEE'01
Deviation Based	(-)Naive Solution is in O(2^n) for n data objects. (+)Heuristics (like Best-First Search) are Applied.	Outliers are outermost points of Data Set.	Springer'13
Distance Based	(-)if different parameters are used for different region of data set.	Normal Data Objects have dense neighbor. Outliers have less dense neighborhood.	IEEE'09
Density Based [6]	(-)Exponential run-time w.r.t. data Dimentionality.	Density around normal data objects similar to the density around it's neighbors.	ACM'13
Cluster Based	(+)Efficient to Discover Clusters. (-)Not Developed to optimize outliers.	No assumptions related to outliers.	ACM'09
Angle Based [8]	(-) Used for High Dimensional Data. (+) naive algo. is in O(n^3).	No Assumption	ACM'14
Grid Based	(-)Quality depends on grid size. (+) Used for High Dimensional Data.	No Assumption	CRC Press'14
Graph Based	(+)Able to detect Point Outliers & Regional Outliers. (-)Focus on single non-spatial attribute outlier detection.	No Assumption	IEEE'07
Classification Based	(-)Selection of kernel Parameter	No Assumption	IEEE'04

Table 2.1: Comparison table for existing techniques.

2.8 Clustering Techniques

Existing Techniques shown in table 2.2 : [16] [17] [18] [5] [13] [10]

Clustering Method	**Algorithms**	**Advantages**	**Disadvantages**
Partitioning Based	K-means (O(n*k*t)). CLARANS O(k*nˆ2). K-medoids O(k*(n-k)ˆ2).	(1)Gradually improves clustering quality. (2)Better than Hierarchical clustering in terms of quality of final clustering solution.	(1) Need initial seeds (clusters). (2) Different initial partitions or values of K affect outcome.
Hierarchical Based	BIRCH (O(n)).ROCK. CHAMELEON (O(nˆ2)). CURE (O(nˆ2(log(n)))).	(1)No apriori information about no.of clusters required. (2)can start off with the individual data points in single clusters.	(1)Use of different distance metrics for measuring distances between clusters may generate different results. (2) Time complexity of at least O(n2 log n).
Density Based	DBSCAN O(nˆ2). DBCLASD . OPTICS O(nˆ2). Denclue O(D*(log(D))).	(1) Can find Arbitrary Shape clusters. (2) Scalability to large Databases. (3) no assumptions about the number of clusters.	(1)Sparse areas are treated as noise and are not assigned to any cluster.
Grid Based	STING O(n*g). WaveCluster O((n)). CLIQUE (O(kˆh + n*h)).	(1)efficient in mining large data sets.	(1)Non-uniformity. (2)Locality. (3)Dimensionality.
Model Based	EM O(nˆ2). Cobweb,Self Organized Map.	(1)good for univariate or less dimensional data.	(1) Data Distributions are fixed. (2) Curse of Dimensionality.

Table 2.2: Comparison table for existing cluster based techniques.

- **In the above algorithm comparison table:**
  1. n → no. of objects.
  2. g → total grid cells.
  3. k → no. of clusters.
  4. t → no. of iteration.
  5. D → active dataset.
  6. h → maximum dimensionality.
- Comparison table shows that density based clustering algorithm are suites to our requirement to obtaining arbitrary shaped clusters.

Chapter 3

DBSCAN

3.1 Definitions

- **Eps-neighbors of a object:** The Eps-neighbors of an object p is defined by N_{Eps}(p)=$\{q \varepsilon D | dist(p,q) \leq Eps\}$.
- **Core Object:** Point p is called as core object if it contains atleast minimum number of points ($MinPts$) in radius (Eps).
- **Directly-density-reachable:** Point p is $Directly - density - reachable$ from a point q wrt Eps, MinPts if p is in Eps-neighbourhood of q, and q is core point.
- **Density-reachable:** wrt Eps, MinPts, a point p is density-reachable from point q if there is a chain of points $p_1, ...p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i.
- **Density-connected:** Point p and q are density-connected wrt Eps, MinPts if both the points are density-reachable from a point o wrt Eps, MinPts.
- **Cluster:** Cluster C wrt Eps, MinPts is non-empty subset of dataset D should satisfy the condition of maximality and connectivity :-
  - $\forall p, q$: if $p \varepsilon C$ and q is density reachable from p wrt Eps, MinPts, then $q \varepsilon C$.
  - $\forall p, q \varepsilon C$:p is density-connected to q wrt MinPts and Eps.

3.2 DBSCAN-Algorithm

3.2.1 Pseudo code of DBSCAN

Algorithm 1 DBSCAN: a density based clustering algorithm.

```
1: mark all objects as unvisited;
2: for each unvisited point p in dataset do
3:    randomly select an unvisited object p;
4:    mark p as visited;
5:    if ∈-neighbourhood of p has atleast MinPts objects then
6:       create a new cluster C, add p to C;
7:       let N be the set of objects in the ∈-neighbourhood of p;
8:       for each point p' in N do
9:          if p' is unvisited then
10:            mark p' as visited;
11:            if ∈-neighbourhood of p' has atleast MinPts points. then
12:               add those points to N.
13:            end if
14:         end if
15:         if p' is not yet a member of any cluster; then
16:            add p' to C.
17:         end if
18:      end for
19:      output C;
20:   else
21:      mark p as noise.
22:   end if
23: end for
```

In algorithm 1 [5] pseudo code of DBSCAN algorithm is given. Initially this algorithm marks all objects as unvisited. Then randomly select an unvisited object p, mark p as visited and find Eps-neighborhood of p. If Eps-neighborhood of p has atleast $MinPts$ objects then create a new cluster C and add p to C. Assume that Eps-neighborhood of point p is N number of objects. For each object p' of N it checks, if p' is unvisited then mark p' as visited and check if Eps-neighborhood of p' has atleast $MinPts$ points then add those points to N. At this point of time if point p' is not yet a member of any cluster, add p' to C. Repeat this until no object is unvisited. Output assigned clusters, if any point p is having Eps-neighborhood less than $MinPts$ then that object is marked as noise.

3.2.2 Problems in DBSCAN

Not reasonable when different density clusters exist [16]:

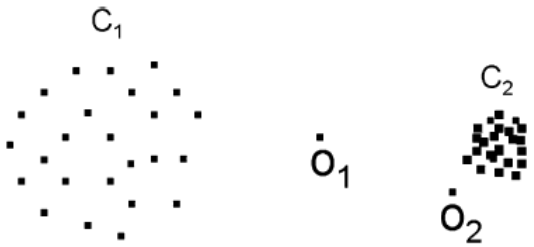

Figure 3.1: Example dataset which having clusters with different-different densities.

Problem will occurs with DBSCAN algorithm when different densities clusters exists. In the given example fig. 3.1. Cluster C1 contains 25 objects and cluster C2 also contains 25 objects, and two noise objects o1 and o2. In the example, cluster C2 is more dense cluster than C1. Densities of both the clusters are different. We can't identify the value of Eps parameter because if Eps value is small then objects o1 and o2 as well as all object of cluster C1 are detected as noise objects.
If Eps value is good enough to identify C1 and C2 as clusters then noise object o2 are may also be in cluster C2.

Problem of identifying adjacent clusters [16]:

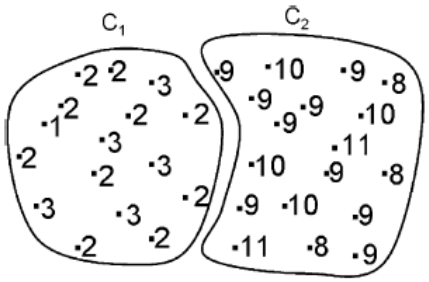

Figure 3.2: Example of adjacent clusters.

DBSCAN clustering algorithm is good if clusters having good enough distance from each other, but not reasonable when clusters are adjacent to each other. The border object value may be very different than the most far border object value. Because changes on small value of neighbors can cause big value changes between starting points and ending

points of a cluster. However cluster objects should be within a certain distance from the cluster means.

Chapter 4

ST-DBSCAN

4.1 Parameters for ST-DBSCAN

ST-DBSCAN algorithm will take four parameter as input.

Parameters: Eps1, Eps2, MinPts, $\triangle\varepsilon$.

where Eps1 radius is used for spatial data and Eps2 radius is used for non-spatial data. $MinPts$ are the minimum number of points within Eps1 and Eps2 radius of an object and $\triangle\varepsilon$ is Threshold.

- calculation of parameter Eps1 ,Eps2 and $\triangle\varepsilon$ is done using k - distance graph.
- MinPts = ln($Dataset\ size$)

4.1.1 Finding Eps1, Eps2 and $\triangle\varepsilon$ using k - distance graph

Heuristic is used to identify Eps1, Eps2 and $\triangle\varepsilon$ using k - distance graph.

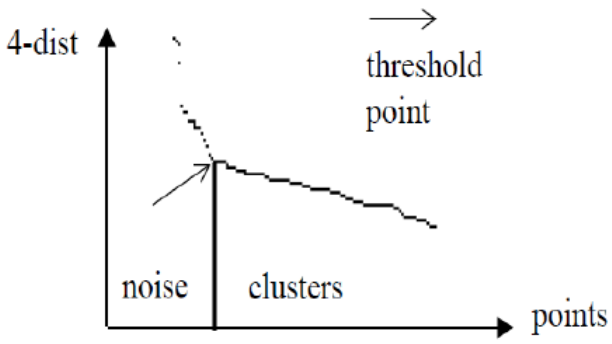

Figure 4.1: For sample dataset, sorted 4-distance graph.

4.1.2 Steps to find parameters using Heuristic

Steps [16]:-

- Find the K-NN distances for each instance, where K is same as $MinPts$.
- Sort in descending order to these k-distance values.
- Plot the sorted distance graph.
- First valley of the sorted graph is the threshold point.
- Eps to be selected as less than the distance found by the first valley.

4.2 ST-DBSCAN Algorithm

[16] DBSCAN algortihm required two inputs, ST-DBSCAN algorithm needs four parameters as input, which are Eps1, Eps2, MinPts, and $\triangle\varepsilon$. Eps1 distance parameter is used for spatial attributes (latitude and longitude). Eps2 distance parameter is used for non-spatial attributes. A distance method such as Euclidean distance, Manhattan distance or Minkowski Distance can be used for calculating Eps1 and Eps2. Minimum number of points within Eps1 and Eps2 distance of a point is represented as MinPts. Based on the heuristic MinPts $\approx \ln(n)$ where size of the database is n.

Algorithm 2 is the ST-DBSCAN algorithm. This algorithm begins with first point o_i of the database D and find the neighbors of that point within Eps1 ans Eps2 radius using $Retrieve\_Neighbors(o_i, Eps1, Eps2)$, if o_i is not in any cluster. If number of points returned by $Retrieve\_Neighbors(o_i, Eps1, Eps2)$ are less than MinPts, then that objects is marked as noise object. It means that the selected object doesn't contains enough neighborhood to be clustered. The noise points may be changed later, if they are density-reachable from some other point of the dataset but not directly-density-reachable. This may happens for border objects of a cluster.

Otherwise, a new cluster is formed and all directly-density-reachable objects of o_i are assigned to that cluster. Then stack is used in the algorithm to repeatedly finds density-reachable points from this core object. Stack is required to find density-reachable objects from directly-density-reachable objects.

If the object is not not in a cluster or it is marked as noise, and the difference between average value of the cluster and new coming value of the object is smaller than $\triangle\varepsilon$, it is placed into the current cluster.

Algorithm 2 ST_DBSCAN (D, Eps1, Eps2, minPts, $\triangle\varepsilon$)

```
1: Cluster_Label = 0
2: for i = 1 to n do
3:    if o_i is not in a cluster then
4:       X = Retrieve_Neighbors(o_i, Eps1, Eps2)
5:       if |X| < MinPts then
6:          Mark o_i as noise.
7:       end if
8:    else
9:       Cluster_Label = Cluster_Label + 1
10:      for j = 1 to |X| do
11:         Mark all objects in X with current Cluster_Label
12:      end for
13:      Push (all objects in X)
14:      while not IsEmpty() do
15:         CurrentObj = Pop()
16:         Y = Retrieve_Neighbors(CurrentObj, Eps1, Eps2)
17:         if |Y| ≥ MinPts then
18:            for All objects o in Y do
19:               if (o is not marked as noise or it is not in a cluster) and
                  |Cluster_Avg() - o.value| ≤ △ε then
20:                  Mark o with current Cluster_Label
21:                  push(o)
22:               end if
23:            end for
24:         end if
25:      end while
26:   end if
27: end for
```

ST-DBSCAN algorithm's proposed approaches to solve the problems of DBSCAN algorithm.[16]

- **Problem-I:** Not reasonable when different density clusters exist.
- **Solution:** Assign $densityfactor$ to each cluster.

  Maximum distance between the object p and its neighbor objects within the given Eps radius is represented by $density\_distance\_max$ and minimum distance between the object p and its neighbor objects within the given Eps radius is represented by $density\_distance\_min$.

  $density\_distance$ of an object p = $density\_distance\_max(p)/density\_distance\_min(p)$.

  $density\_factor$ of a cluster C = 1 / [$\sum_{p \epsilon C} density\_distance(p)/|C|$]
- **Problem-II:** Problem of identifying adjacent clusters.
- **Solution:** $cluster\_Avg()$ - obj.val $\leq threshold(\triangle\varepsilon)$

Chapter 5

ST-OUTLIER

5.1 Definitions

- Outliers are the observations which deviate with the remainder of the dataset.
- Spatial outliers are the objects whose non-spatial attribute values are notably deviated from the values of its spatial neighbourhood.
- Temporal outliers are the objects whose non-spatial attribute value is notably deviate from other objects in its temporal neighbourhood.
- Spatio-temporal outliers are the objects whose non-spatial attribute values are notably deviate from other objects in its spatial neighbourhood as well as temporal neighbourhood.

5.2 ST-Outlier Detection Algorithm Steps

Steps: [19]

- **Step-I:** Clustering.
- **Step-II:** Identify spatial outliers.
- **Step-III:** Identify temporal outliers.

Chapter 6

SNN Similarity and Our Proposed Approach

6.1 SNN Similarity

An alternative to a direct similarity measures (Euclidean Distance, Cosine Similarity etc.) between a pair of points in terms of number of points two points are share. That is similarity between two points is measured by common nearest neighbours. This approach was first proposed by R.A. JARVIS and EDWARD A. PATRICK [4].

In Jarvis-Patrick approach, a shared nearest neighbour (SNN) graph is constructed from the similarity matrix by a link is created between a pair of points, A and B, if and only if A and B have each other in their k-nearest neighbour lists. This process is named as k-nearest neighbour sparsification. The weights of the links between two points in the SNN graph is the number of nearest neighbour two points share. If A and B be two points then the strength of link between A and B, i.e., their similarity if given by :-

$$similarity(A, B) = size(NN(A) \cup NN(B))$$

where NN(A) and NN(B) are nearest neighbour lists of A and B. Clusters can be obtained by removing all edges with similarities less than threshold and remaining all connected components are clusters. This approach is names as Jarvis-Patrick clustering [20].

SNN similarity provides us with a more robust measure of similarity that works well for data with low, medium and high dimensionality [21].

To calulate SNN similarity between spatio-temporal data points :-

- Calculate k-nearest neighbour for each points of spatial as well as for non-spatial.
- Identify common nearest neighbours for a point in spatial and non-spatial domain. Those points are nearest neighbours for a point.
- Based on the identified nearest neighbours in spatial and non-spatial domain, find SNN similarity between each points and create a SNN similarity matrix.

6.2 Parameters

- parameters for ST-DBSCAN :- Eps1, Eps2, MinPts, $\triangle\varepsilon$.
- parameters for ST-SNN :- K, Eps, MinPts, $\triangle\varepsilon 1$, $\triangle\varepsilon 2$.
- K, Eps and MinPts are our design parameters which are selected by trial and error method.
- $\triangle\varepsilon 1$ and $\triangle\varepsilon 2$ are selected using K-distance graph.

6.3 Proposed Approach

Our proposed approach is based on shared nearest neighbour similarity measure. Our approach first find SNN-similarities of each point as explined above and create a similarity matrix. After creating similarity matrix our algorithm process all the points from the dataset. In our algorithm Eps and MinPts are taken as a fraction of K. Our algorithm start with taking a point and find shared nearest neighbours(SNN) within Eps range. If number of shared nearest neighbours of that point are less than MinPts then that point is marked as noise it means that the selected object doesn't contains enough neighbourhood to be clustered. The noise points may be changed later, if they are density-reachable from some other point of the dataset but not directly-density-reachable. This may happens for border objects of a cluster. Otherwise create a new cluster and place all the neighbours of that point into this new cluster and then we find density reachable points from all directly-density-reachable points.

If any object is not marked as noise or it is not in a cluster and the difference between the spatial average value of the cluster and the new coming spatial value is smaller than $\triangle\varepsilon 1$ and the difference between the non-spatial average value of the cluster and the new coming non-spatial value is smaller than $\triangle\varepsilon 2$ then it is placed into current cluster.
After processing selected object, the algorithm selects the next object in dataset and algorithm continues to process all the objects. To make our results comparable, agglomerative clustering is used to merge the clusters. Agglomerative clustering required number of clusters as input parameter, so for this purpose we have taken crop list from Gujarat agriculture department, where number of crops are 20, so 20 as the number of clusters are given input to the agglomerative clustering.

Using the above approach we properly clustered our dataset. After clustering the dataset we moved to step-II (identifying spatial outliers) and found spatial outliers and then found ST-Outliers which is our step-III, by comparing the spatial outliers regions with its temporal neighbours. If spatial outliers regions does not have significant differences with its temporal neighbours, then this is not a ST-Outlier, otherwise this region is confirmed as a ST-Outlier.

Chapter 7

Experiments

7.1 Specification of Data

Satellite	-	Moderate Resolution Imaging Sepctro-radiometer (MODIS)
Measure index	-	Normalized Difference Vegetation Index, (NDVI)(16 days composite)
Region	-	Whole Country (India)
No.of Grids	-	1,30,307
Size of the Grid	-	5*5 km

Table 7.1: Specification of Data

7.2 ST-DBSCAN Results

7.2.1 K - Distance graph

We performed our experimentation on "Vegetation Data of Gujarat Region for the Year-2001 to 2004". Year-2001 data have 7028 grid, Year-2002 data have 7534 grid year-2003 data have 7534 grid and Year-2004 data have 7534 grid. Using k - distance graph heuristic as shown in figure 7.1 an 7.2, we identified input parameters Eps1, Eps2 and $\triangle\varepsilon$.

7.2.2 K - Dist Graph

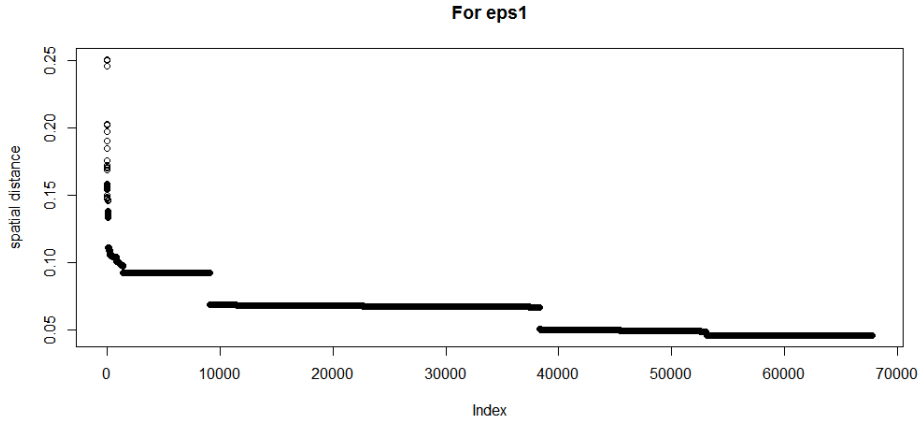

Figure 7.1: Graph for Eps1 value.

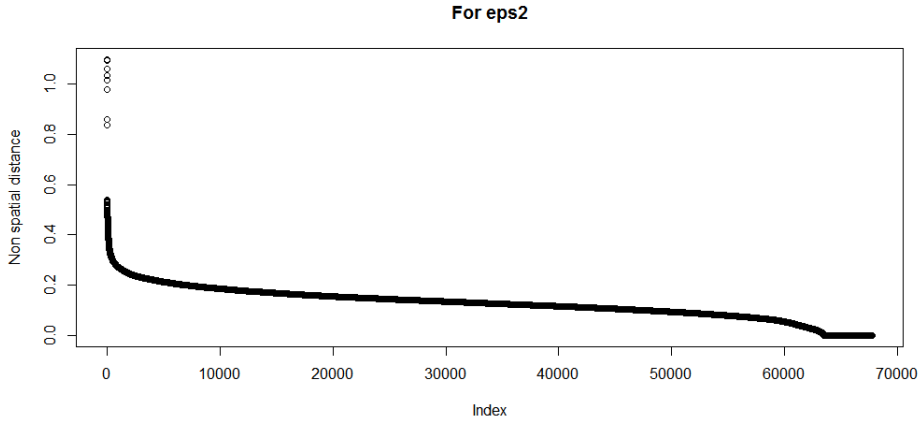

Figure 7.2: Graph for Eps2 value.

7.2.3 Parameter Table for ST-DBSCAN

Parameter/Year		2001-Data	2002-Data	2003-Data	2004-Data
Eps1	:	0.2022	0.2022	0.2022	0.2022
Eps2	:	0.6651	0.4511	0.5851	0.5811
MinPts	:	9	9	9	9
Threshold	:	0.6661	0.4521	0.5861	0.5821

Table 7.2: Parameter table for ST-DBSCAN

7.2.4 Clusters given by ST-DBSCAN algorithm

figure 7.3 and 7.4 shows Gujarat map, taken from forest survey of India web portal [http://fsi.nic.in/]. This portal is managed by forest survey of India (Under Ministry of Environmental & Forest). We applied ST-DBSCAN algorithm on our vegetation data and found results as shown in figure 7.5 and 7.6. Comparison of figure 7.5 and 7.6 with forest survey of India map shown in figure 7.3 and 7.4, we found that ST-DBSCAN algorithm is not giving satisfactory results.

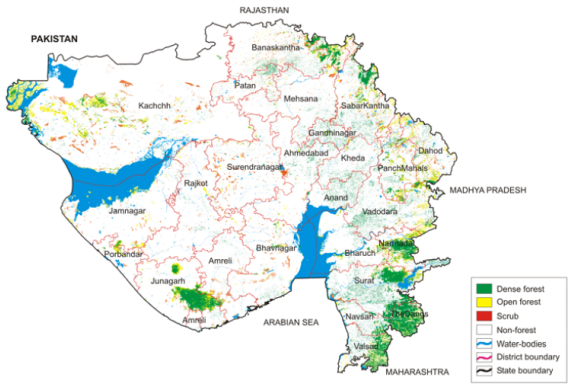

Figure 7.3: Gujarat map for year 2001 by forest survey of India.

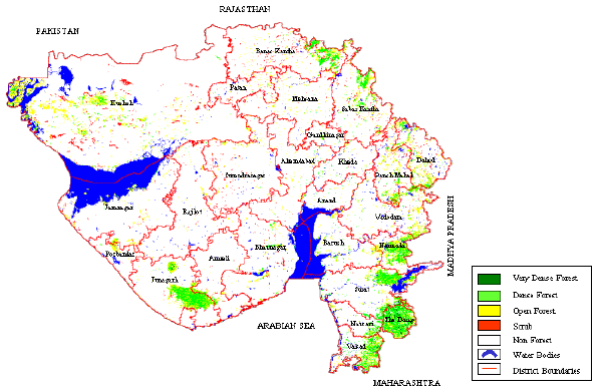

Figure 7.4: Gujarat map for year 2003 by forest survey of India.

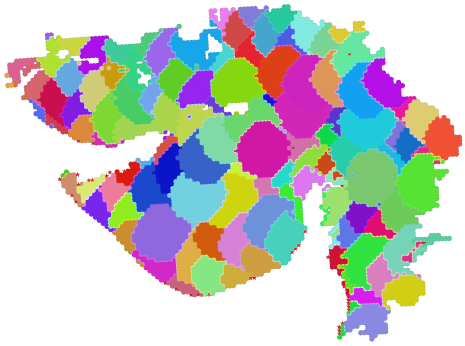

Figure 7.5: Cluster Map for 2001 Dataset

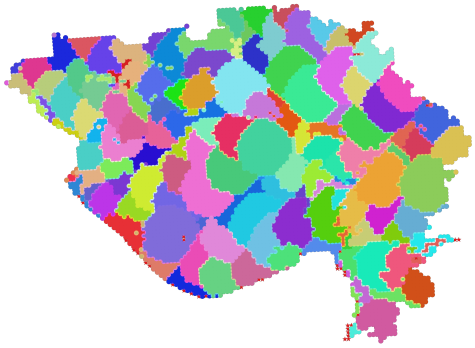

Figure 7.6: Cluster Map for 2003 Dataset

7.2.5 Agglomerative algorithm to make results comparable

To compare the clustering results with forest survey map, we are using Agglomerative algorithm. Our ST-DBSCAN algorithm result are given as an input to Agglomerative algorithm. Agglomerative algorithm takes "number of clusters required" as input parameter, so for this purpose we taken crop list from Gujarat agriculture department

[www.agri.gujarat.gov.in].

Figure 7.9 , 7.10 shows clusters after performing Agglomerative algorithm.

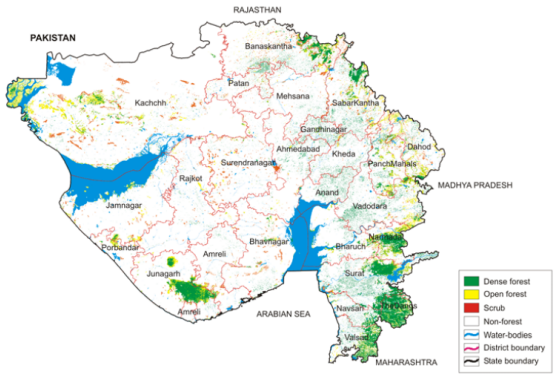

Figure 7.7: Gujarat map for year 2001 by forest survey of India.

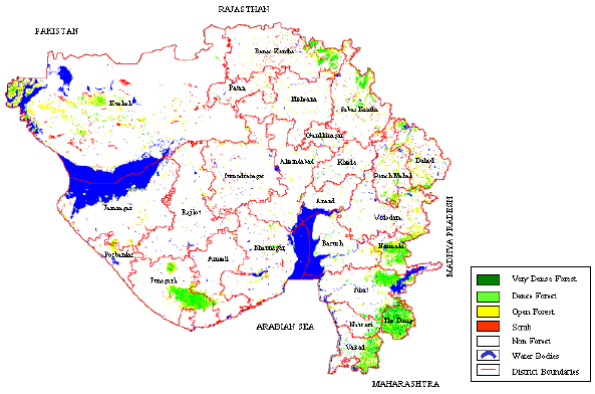

Figure 7.8: Gujarat map for year 2003 by forest survey of India.

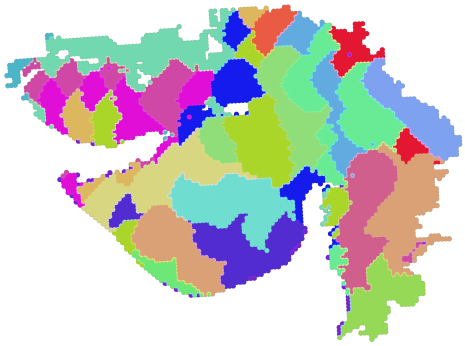

Figure 7.9: Cluster Map for 2001 Dataset

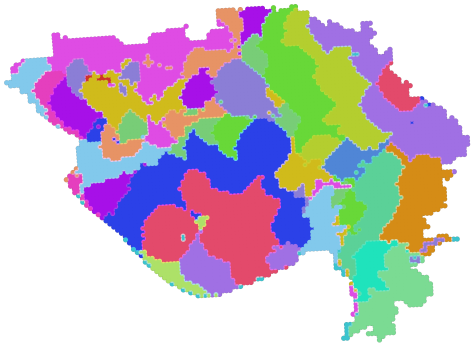

Figure 7.10: Cluster Map for 2003 Dataset

7.3 Result of Our ST-SNN and ST-Outlier Approach

7.3.1 Parameter Table for ST-SNN

We performed our experimentation on "Vegetation Data for Gujarat Region from the Year-2001 to 2004". Parameter K, Eps and MinPts are found by trial and error method where as $\triangle\varepsilon 1$ and $\triangle\varepsilon 2$ are found using k-distance graph.

Fig. 7.13, 7.14 shows clusters after performing our ST-SNN algorithm and Agglomerative algorithm.

Parameter/Year	2001-Data	2002-Data	2003-Data	2004-Data
K	200	200	200	200
Eps	4	4	5	4
MinPts	6	6	7	7
Threshold1	0.2022	0.2022	0.2022	0.2022
Threshold2	0.6661	0.4500	0.5861	0.5800

Table 7.3: Parameter Table for ST-SNN

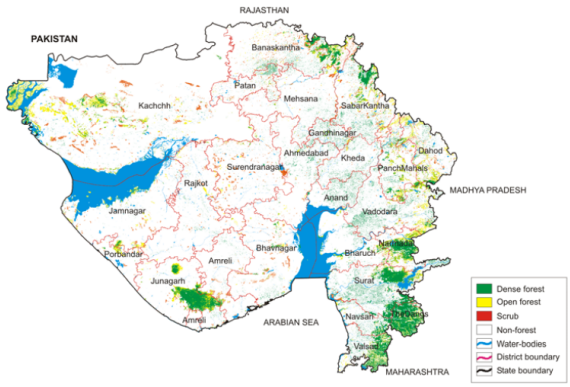

Figure 7.11: Gujarat map for year 2001 by forest survey of India.

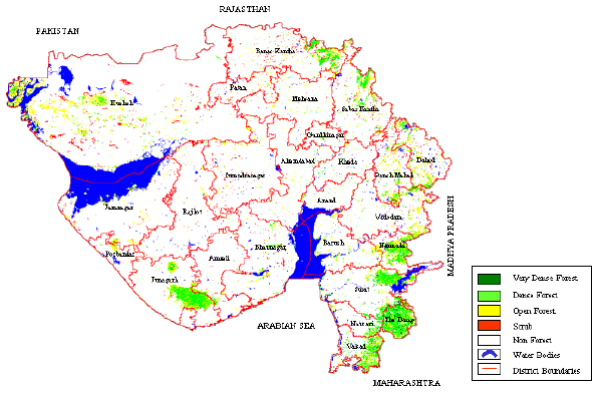

Figure 7.12: Gujarat map for year 2003 by forest survey of India.

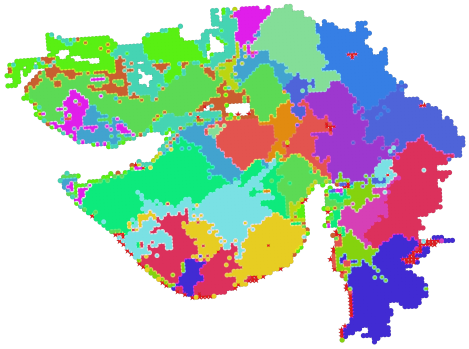

Figure 7.13: Cluster Map for 2001 Dataset

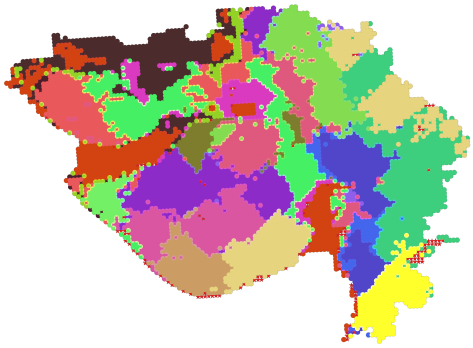

Figure 7.14: Cluster Map for 2003 Dataset

We selected Gujarat 2003 NDVI dataset to detect outlier in this year. So after successful completion of first step which is clustering, we move to second step to identify spatial outlier region in selected dataset. Figure 7.15 shows possible spatial outlier regions which are marked as red coloured astrik marks.

After identifying the spatial outlier regions in the previous step, we compare S-Outliers to other objects of the same location area but in different times to identify ST-Outliers.

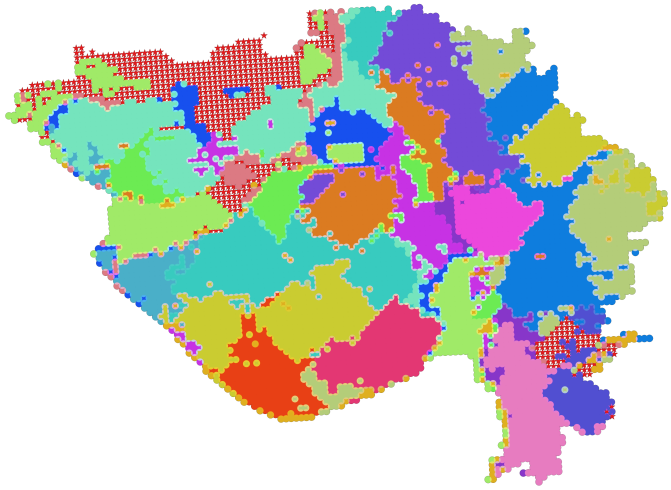

Figure 7.15: Spatial outliers detected in Gujarat 2003 dataset.

If the characteristic value of a S-Outlier does not have significant differences with its temporal neighbours, then this is not a ST-Outlier. We compare Gujarat 2003 NDVI Dataset with Gujarat 2002 NDVI and Gujarat 2004 NDVI dataset to confirm that S-Outliers are ST-Outliers.

Figure 7.16 shows spatio-temporal outlier regions which are marked as red coloured astrik marks.

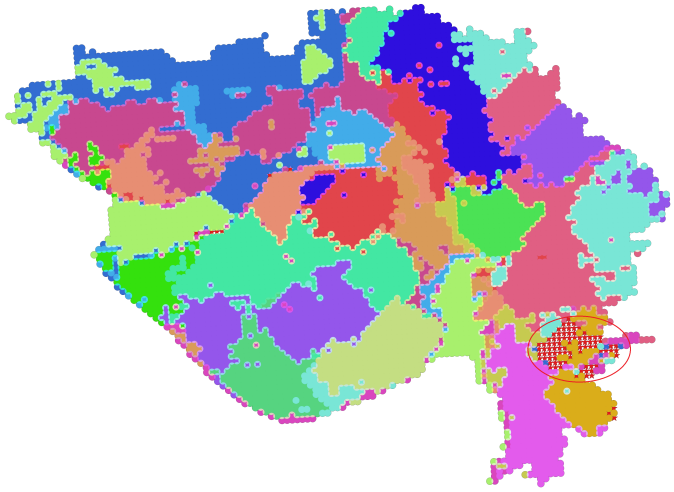

Figure 7.16: Outliers detected in Gujarat 2003 dataset.

Chapter 8

Conclusion and Future Work

In this thesis, we have described an approach to detect outliers in high dimensional spatio-temporal data. This proposed approach is uses a new clustering approach ST-SNN which is based on shared nearest neighbour and density based clustering technique. This clustering algorithm can find clusters of varying shapes, sizes, densities and automatically identifies the number of clusters. To validate through comparison, agglomerative clustering approach is used to merge the clusters and compare with forest survey of India map. Then we identified spatial outliers and then finally spatio-temporal outliers are detected in NDVI dataset which we are shown in our experimental results. Experimental results demonstrate that our outlier detection approach is very promising for spatio-temporal data.

In future studies, a heuristics may be suggested to determine the input parameter K, Eps and Minpts.

References

[1] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, (New York, NY, USA), pp. 93–104, ACM, 2000.

[2] S. Papadimitriou and C. Faloutsos, "Cross-outlier detection.," in *SSTD* (T. Hadzilacos, Y. Manolopoulos, J. F. Roddick, and Y. Theodoridis, eds.), vol. 2750 of *Lecture Notes in Computer Science*, pp. 199–213, Springer, 2003.

[3] Z. He, X. Xu, and S. Deng, "Discovering cluster based local outliers," *Pattern Recognition Letters*, vol. 2003, pp. 9–10, 2003.

[4] R. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *Computers, IEEE Transactions on*, vol. C-22, pp. 1025–1034, Nov 1973.

[5] M. Ester, H. peter Kriegel, J. S, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," pp. 226–231, AAAI Press, 1996.

[6] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," in *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD '98, (New York, NY, USA), pp. 73–84, ACM, 1998.

[7] C. Aggarwal, *Outlier Analysis*. Springer, January 2013.

[8] J. Han, M. Kamber, and J. Pei, "1 - introduction," in *Data Mining (Third Edition)* (J. H. Kamber and J. Pei, eds.), The Morgan Kaufmann Series in Data Management Systems, pp. 1 – 38, Boston: Morgan Kaufmann, third edition ed., 2012.

[9] M. Gupta, J. Gao, C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, pp. 2250–2267, Sept 2014.

[10] Kriegel/Krger/Zimek, "Outlier detection techniques," 2010.

[11] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, pp. 85–126, Oct. 2004.

[12] E. Jordaan and G. Smits, "Robust outlier detection using svm regression," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 3, pp. 2017–2022 vol.3, July 2004.

[13] S.-Y. Jiang and Q. bo An, "Clustering-based outlier detection method," in *Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on*, vol. 2, pp. 429–433, Oct 2008.

[14] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 15:1–15:58, jul 2009.

[15] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *SIGMOD Rec.*, vol. 30, pp. 37–46, May 2001.

[16] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, pp. 208–221, Jan. 2007.

[17] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *SIGMOD Rec.*, vol. 28, pp. 49–60, June 1999.

[18] J. Han, M. Kamber, and J. Pei, "11 - advanced cluster analysis," in *Data Mining (Third Edition)* (J. H. Kamber and J. Pei, eds.), The Morgan Kaufmann Series in Data Management Systems, pp. 497 – 541, Boston: Morgan Kaufmann, third edition ed., 2012.

[19] D. Birant and A. Kut, "Spatio-temporal outlier detection in large databases," in *Information Technology Interfaces, 2006. 28th International Conference on*, pp. 179–184, 2006.

[20] L. Ertz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in *in Proceedings of Second SIAM International Conference on Data Mining*, 2003.

[21] L. Ertoz, M. Steinbach, and V. Kumar, "A new shared nearest neighbor clustering algorithm and its applications," in *Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining*, 2002.