Detecting Outliers in Topological Datasets: Algorithms and Applications

Presented by: Chang-tien Lu
Department of Computer Science
University of Minnesota

Overview

- Background
- Outliers Detection: An Introduction
- Application Domain: Twin-Cities Traffic Data
- Related Work
  - *Geometry Approach: Global Approach*
  - *Geometry Approach: Local Approach*
- Topological Approach
- Experimental Observations and Results
- Discussion & Future Direction

Background

- Huge amounts of Spatial data
  - *NASA EOS: generate 50GB of date per hour*
- Data mining
  - *Discover interesting, implicit, and previous unknown knowledge from large databases*
  - *Discover knowledge rules, constraints, regularities*
- Spatial Data Mining
  - *Discover interesting spatial patterns and features*
  - *Capture intrinsic relationships between spatial and non-spatial data*
  - *Present data regularity concisely and at higher conceptual levels*
- Applications of spatial data mining
  - *GIS systems*
  - *Remote Sensing*
  - *Image Database Exploration*
  - *Medical Images*

Outliers Detection

- Four Categories of Knowledge Discovery
  - *Dependency Detection, e.g., Association Rules*
  - *Class Identification, e.g., Classification, Clustering*
  - *Class Description, e.g., Concept Generalization*
  - *Exception/Outlier Detection*
- Informal Definitions of Outliers
  - *An observation which appears to be inconsistent with the remainder of that set of data*
  - *An observation which deviates so much from other observations*
    - To arouse suspicious that is was generated by a different mechanism
- Application of Outlier Detection
  - *Discovery of truly unexpected knowledge*
    - Electronic commence exceptions, credit card fraud
  - *Detect abnormal events in the past*
  - *Predict potential trends in the future*
  - *New direction for future invest, marketing*

Application Domain: Twin-Cities Traffic Data

- Map and Tables

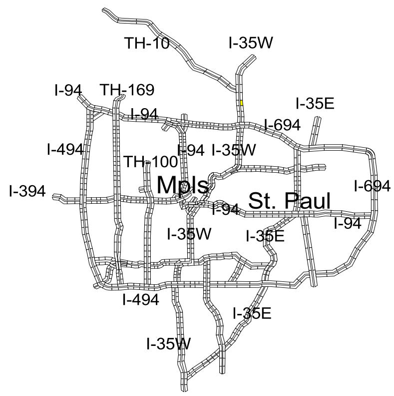
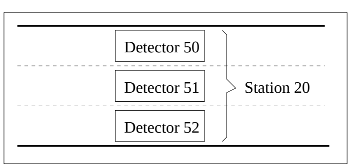


Figure 1: Detector map in station level



(a) Relationship between detectors and stations

Station Table

Station	Polygon_id	Polygon Boundary	Location	Freeway	Direction	Zone	.....
1	P1	(3,5),(4,10),..	26th St.	I-35W	N	Q4	
2	P3	(5,7),(6,4),....	28th St.	I-35W	N	Q4	
....							

Detector Table

Detector	Station
1	1
2	1
....	

Value Table

Time	Detector	Volume	Occupancy
1997 1 12 12:30	1	50	3
1997 1 12 12:50	2	60	12
....			

(b) Three basic tables

Figure 2: Detector-station Relationship and Basic Tables

- Traffic Outlier Related Questions
  - *What forms the abnormalities manifest themselves?*
  - *When those abnormalities arise, and how long does it last?*
  - *Where those abnormalities happen?*
  - *How would it affect its neighborhood traffic stations?*

Our Topological Approach

- Comparison

	Geometric (Multi-dimension)		Topological	
	Global	Local	Global	Local
Distance	Ng	Ramaswamy		
Density	Yu($FindOut$)	Breunig($OPTICS-OF$)		
Statistical (Distribution threshold)				
Statistical (Depth threshold)	Ruts			

Table 1: Summary of related work

Geometric: Global: Distance Based

- Definition
  - *For a k dimensional dataset T with N objects*
  - *An object O in T is a $DB(p, D)$-outlier*
    - At least fraction p of the objects in T lies greater than D from O
  - *Execute a range search with radius D for each object O*
  - *Complexity $O(kN^2)$*
- Cell-based algorithm
  - *Divide each dimension i into m_i partitions*
  - *Assign objects to the cells*
  - *Complexity, linear to N, exponential to K*
  - *Efficient when $K \leq 4$*
- Key weakness
  - *Require the existence of an appropriate distance function*
    - Hard to find distance function for categorical attributes
  - *In a high-dimensional space, almost all pairs of points are about as far away as average*
  - *Require user to specify a distance d which could be difficult to determine*
  - *Does not provide a ranking for the outliers*

Geometric: Global: Wavelet Transform, Cell Based(Find-Out)

- Remove the clusters from the original data and identify outliers
- Algorithm
  - Quantize feature space
  - Apply wavelet transform
  - Find the connected components(clusters)
  - Map data to the clusters
  - Outliers = feature space - clusters - boundaries
- Time Complexity
  - Max(O(N), O(K)), N: no of objects, K $=m^d$, m : no of cells, d : n of dimensions
- Key weakness
  - Unfit for high dimension, unless dimensionality reduction(e.g., PCA) can be done
  - The definition of significant cell is a global parameter
- Example

Geometric: Statistical (Distribution threshold)

- main idea
  - *Fit the dataset to a known standard distribution, determine outliers using this distribution*
  - *Discordancy tests: distribution-based definitions of outliers*
- Different Discordancy Tests
  - *The data distribution*
  - *Whether the distribution parameters are known*
  - *No. of expected outliers*
  - *Types of expected outliers*
- Key weakness
  - *For many applications, distribution of attribute values is unknown*
  - *Apply distribution fitting*
    - Given distribution may not fit any standard distributions
    - Take long time to conduct such fitting
  - *Almost all of these tests are univariate, unsuitable for multidimensional datasets*

Geometric: Statistical (Depth threshold)

- Main idea
  - *Organize the data objects in layers in the data space*
  - *Outliers are more likely to occur in shallow layers*
- Simple definition of depth
  - *Convex hull peeling depth*
  - *Repeatly find the convex hull of the set of points*
  - *Assign depth*
  - *Remove the points on the convex hull*
- Key weakness
  - *Not applicable for high dimensional datasets*
    - Best case convex hull computation: $\Omega(N^{\lceil k/2 \rceil})$
    - N: no. of objects, k: dimensionality
  - *Restricted to be extreme values*
    - Not capture distance and distribution
    - Possibility: High depth, but far away from most of the points
    - Example: Bi-modal distribution

Geometric: Local: Distance

- Main idea
  - *Based on the distance of a point form its k-th nearest neighbor*
  - *Rank each point on its distance to its k-th nearest neighbor*
  - *Top n points in the rank* $\Rightarrow$ *outliers*
- Key weakness
  - *User specify: parameter k*
- Algorithm
  - *Generate partitions*
    - Use a clustering algorithm, e.g. CF-Tree in BIRCH
  - *Compute bounds on* D^k *for points in each partition*
    - Lower bound and upper bound
  - *Identify candidate partitions containing outliers*
    - Prune entire partitions that cannot contain outliers
  - *Compute outliers from points in candidate partitions*

Geometric: Local: Density-based

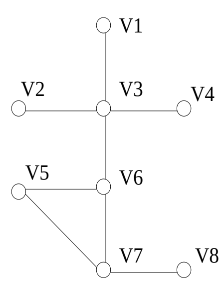
- Main idea
  - *Outlier degree: determined by clustering structure in a bounded neighborhood of the object*
  - *Objects are outliers relative to their local, surrounding object distribution*
- Key idea
  - Eps-neighborhood: within a radius Eps of a given object
  - Core object: Eps-neighborhood of an object contains at least a minimum number, MinPts
  - Density-reachable: p is within the Eps-neighborhood of q, and q is a core object
- Define outlier factor
  - Based on the same theoretical foundation as density based cluster analysis
  - Capture this relative degree of isolation or outlierness
- Key weakness
  - $O(n^2)$, *n: no of objects, depends on k-nearest-neighbor query, with spatial index: O(n log(n))*
  - *User specify: Eps, and MinPts*
- Examples

Local Topological Approach

- Given
  - V *is the attribute data set in* S
  - *Topological Graph* $G = (V, E)$

- Output
  - *Outlier_Set*

- Steps
  - *For each data object*
    - Find its topological neighbors
    - Calculate the average distance to its neighbors in S
  - *Construct the distribution model*
  - *Detect outliers via the distribution model*



(a) Topological Space

	Attr1	Attr2	.....	AttrN
V1				
V2				
V3				
V4				
V5				
V6				
V7				
V8				

(b) Attribute Space

Figure 3: Topological Space and Attribute Space

Topological Algorithm for Outlier Detection

Input: S is the multidimensional attribute space;
D is the attribute data set in S;
F is the distance function in S;
ND is the depth of neighbor;
$G = (D, E)$ is the topological graph;
CI is the confidential interval;

Output: Outlier_Set

```
for(i=1;i ≤ |D| ;i++){
    O_i=Get_One_Object(i,D); /* Select each object from D */
    NNS=Find_Neighbor_Nodes_Set(O_i,ND,G);
   /* Find neighbor nodes of O_i from G */
    Accum_Dist=0;
    for(j=1;j≤ |NSS|;j++){
      O_k=Get_One_Object(j,NNS); /* Select each object from NNS */
      Accum_Dist += F(O_i, O_k, S)
    }
    Avg_Dist = Accum_Dist/|NNS|;
    Add_Element(AvgDist_Set,i); /* Add the element to AvgDist_Set */
}
μ = Get_Mean(AvgDist_Set); /* Compute Mean */
δ = Get_Standard_Dev(AvgDist_Set); /* Compute Standard Deviation */
Normailze(AvgDist_Set,μ,δ); /* Normalize AvgDist_Set*/
Outlier_Set = Check_Table(AvgDist_Set,CI); /* Derive the Outlier_Set*/
return Outlier_Set;
```

Experiment Design

- Experiment Data Set
  - *Twin-Cities Traffic Data*
  - *Point: Each station*
  - *Edge: each station and its neighbor stations*
  - *Each data point*
    - Attribute Values
    - Successor List
    - Predecessor List
    - Size: 256 bytes

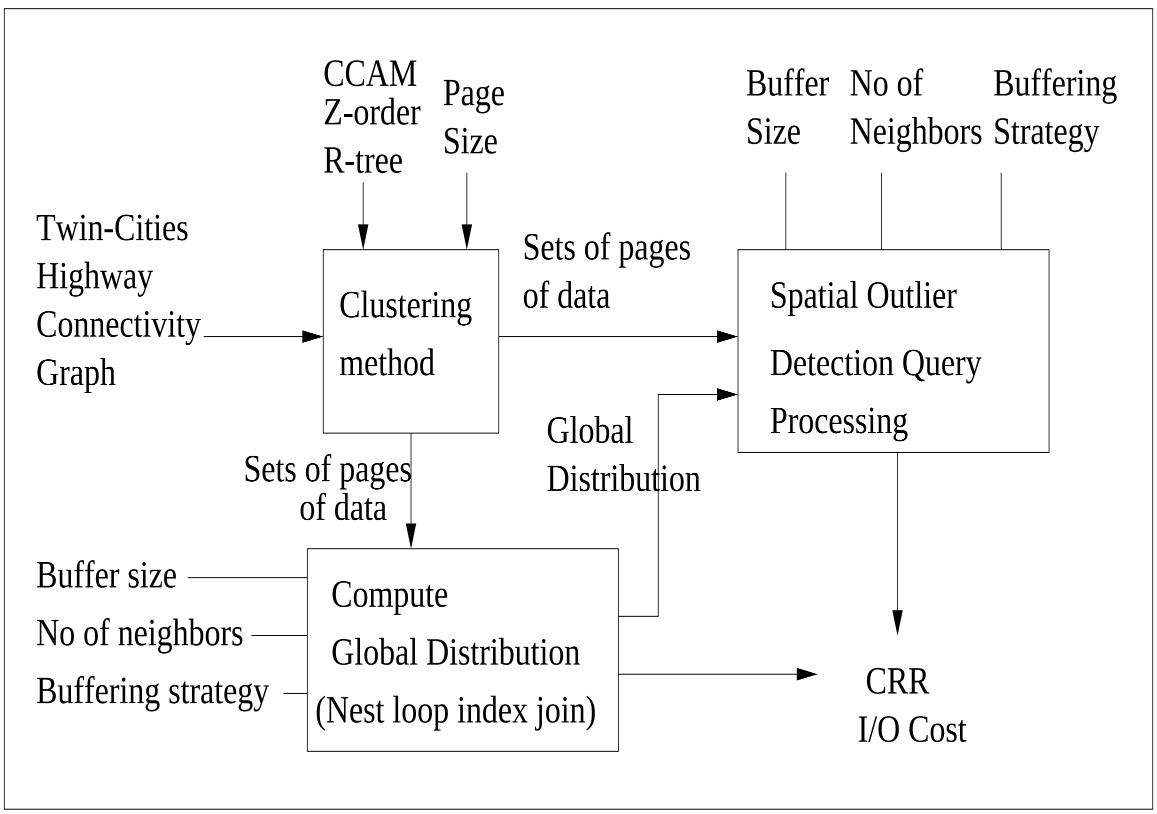


Figure 4: Experimental Layout

Parameters of Interest

- Parameters
  - *Physical Page Clustering Methods*
    - CCAM
      - Cluster the node of the network via graph partitioning
      - *Each partition corresponds to one disk page*
    - Cell Tree
      - *Each cell-tree node corresponds to a convex polyhedron*
      - *Each cell-tree node corresponds to one disk page*
      - *Viewed as a combination of Binary Space Partitioning(BSP) and* R^+*-tree*
    - Z-order
      - Interleaving the bits in binary representation of the two values
  - *Buffer Management Strategies*
    - LRU: Least Recently Used Algorithm
    - MRU: Most Recently Used Algorithm
    - FIFO: First In First Out
  - *Page Size*
  - *Buffer Size*
  - *Number of Neighbors*
  - *Neighborhood Depth*

- Measures
  - *I/O cost (number of page access)*
  - *CRR value = (Total number of unsplit edges)/(Total number of edges)*

Experimental Observation and Results

- Step 1: Model Construction
  - *Compute global distribution*
  - *Nest Loop index join*
- Step 2: Outlier Detection
  - *Random Point Test*
  - *Detect Outliers along a Route*
  - *Detect Outliers within a Region*

Model Construction

- Fixed Parameters
  - *Page size: 2K*
  - *Buffering strategy: LRU*
  - *CRR value: CCAM = 0.81 Cell = 0.69 Z-ord = 0.51*
- Variable Parameters
  - *Number of buffers: 2,4,6,8,10,12,14,16,18,20*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*

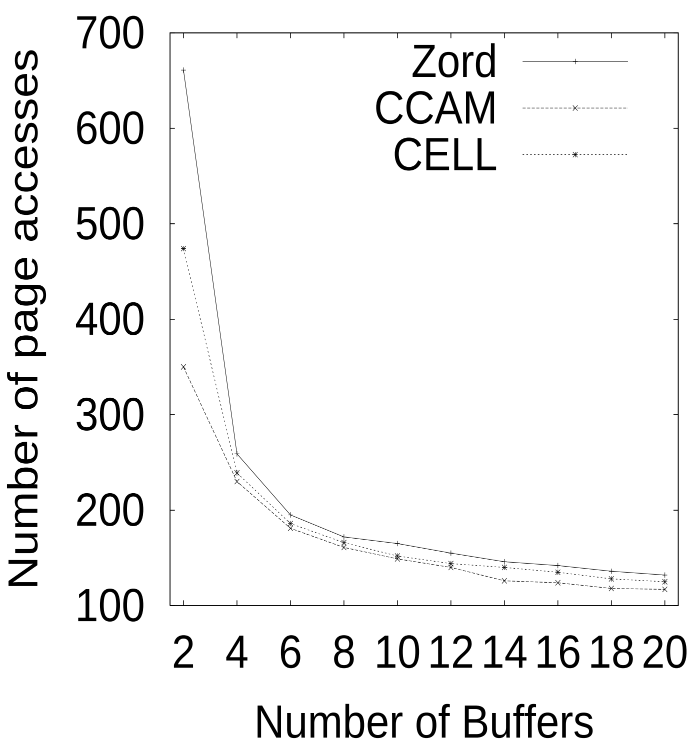


Figure 5: Effect of buffering on data page accesses (Block Size = 2K)

- *Increase Buffer size => reduce number of page accesses*
- *CCAM has the best performance*

Model Construction

- Fixed Parameters
  - *Page size: 2k*
  - *Page clustering strategy: CCAM (CRR:0.81)*
  - *Neighborhood depth: 1*
- Variable Parameters
  - *Number of buffers: 4,6,8,10,12,14,16,18,20*
  - *Buffering strategy: MRU, LRU, FIFO*

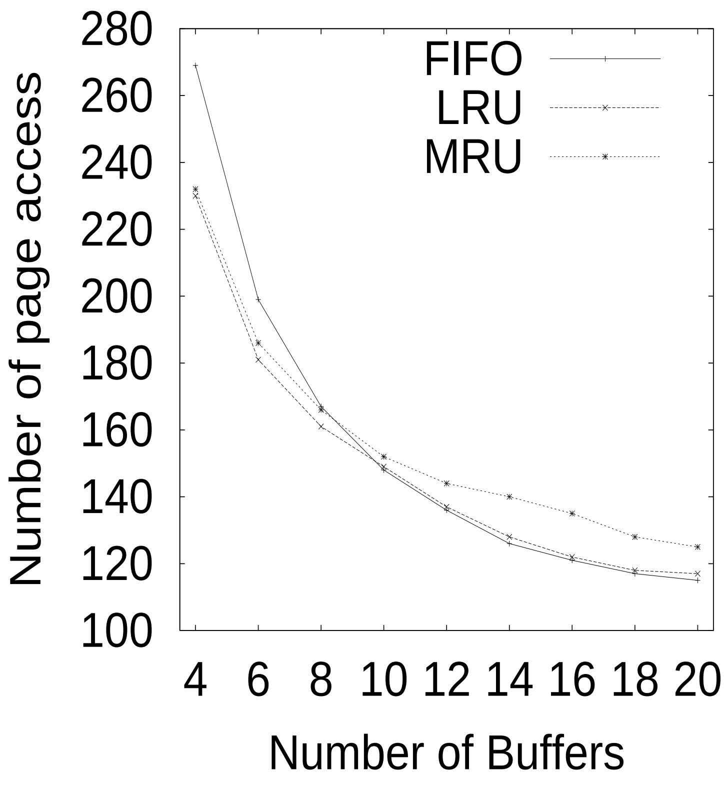


Figure 6: Effect of buffering strategy (Block Size = 2K)

- *LRU has the best performance with small buffer size*
- *FIFO and LRU perform better with large buffer size*

Model Construction

- Fixed Parameters
  - *Buffer Size: 64k*
- Variable Parameters
  - *page size: 1K,2K,4K,8K*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*
- Measure
  - *Number of page accesses*

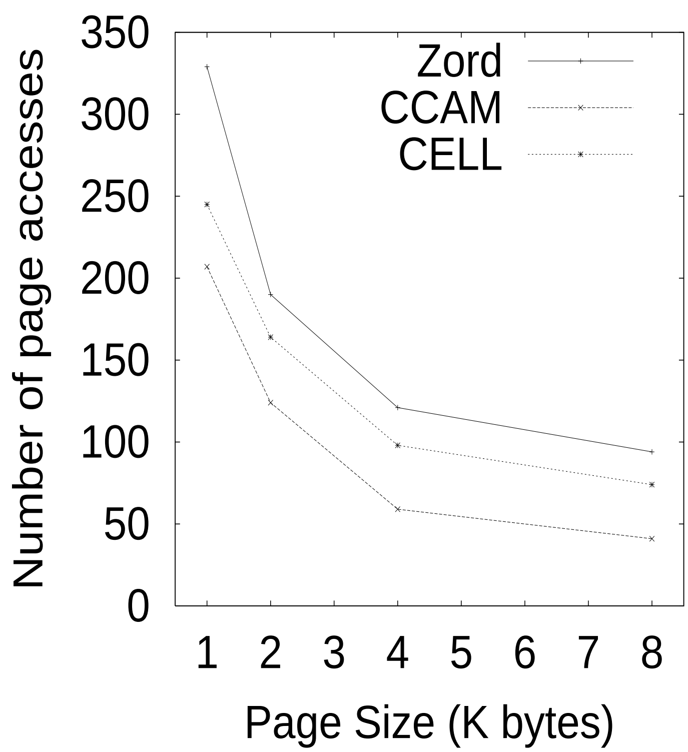


Figure 7: Effect of page size on data page accesses (Buffer Size = 64K)

- *CCAM has the best performance*
- *Increase page size => reduce number of page accesses*

Model Construction

- Fixed Parameters
  - *Buffer Size: 64k*
- Variable Parameters
  - *page size: 1K,2K,4K,8K*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*
- Measure
  - *CRR value*

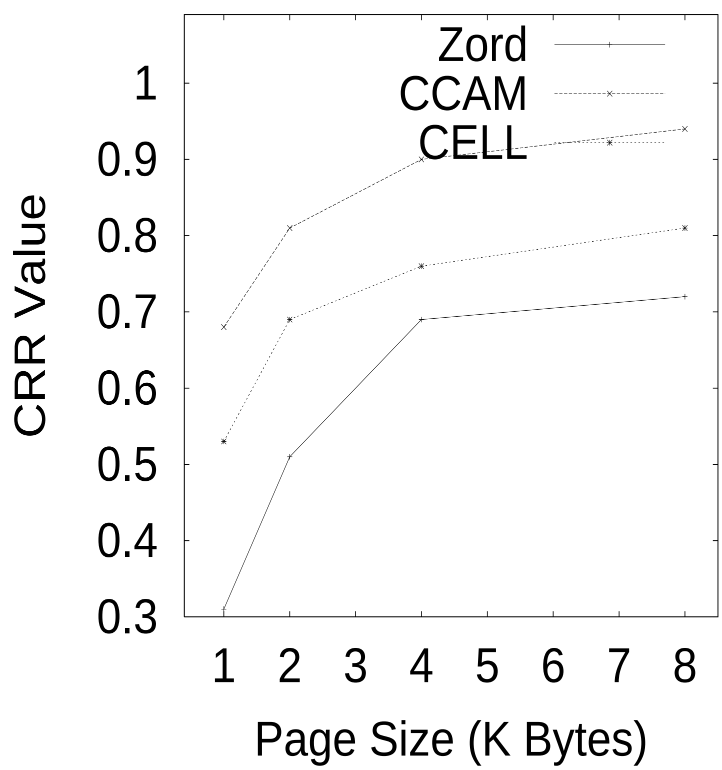


Figure 8: CRR value for different page size

- *CCAM has the highest CRR value*
- *High CRR => Low I/O cost*

Model Construction

- Fixed Parameters
  - *Page Size: 1k*
  - *Buffer size: 4K*
  - *Buffering Strategy: LRU*
- Variable Parameters
  - *No of Neighbors: 2, .., 10*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*

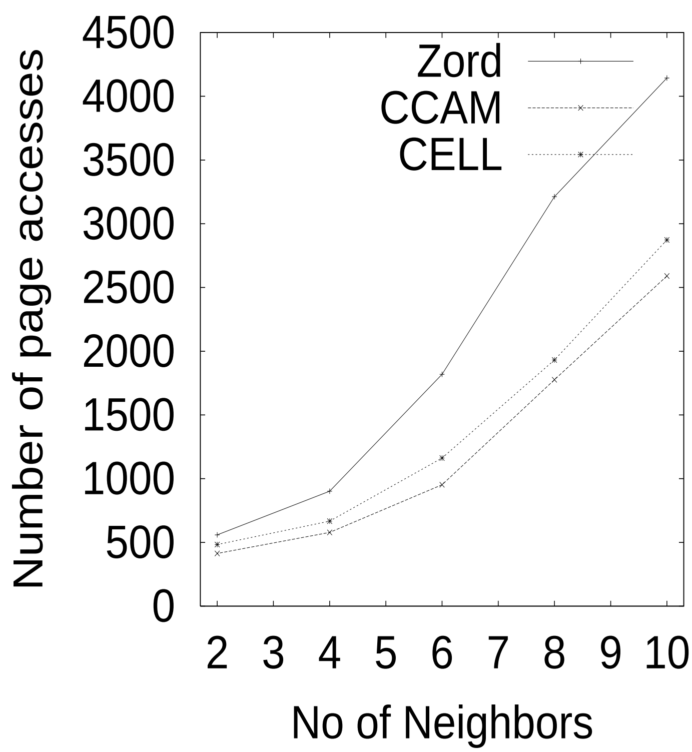


Figure 9: Effect of neighborhood number on data page accesses

- *Increase No of neighbors => High I/O cost*
- *CCAM has the best performance*

Model Construction

- Fixed Parameters
  - *Page Size: 2k*
  - *Buffer size: 16K*
  - *Buffering Strategy: LRU*

- Variable Parameters
  - *Neighborhood Depth: 1, 2, 3, 4, 5*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*

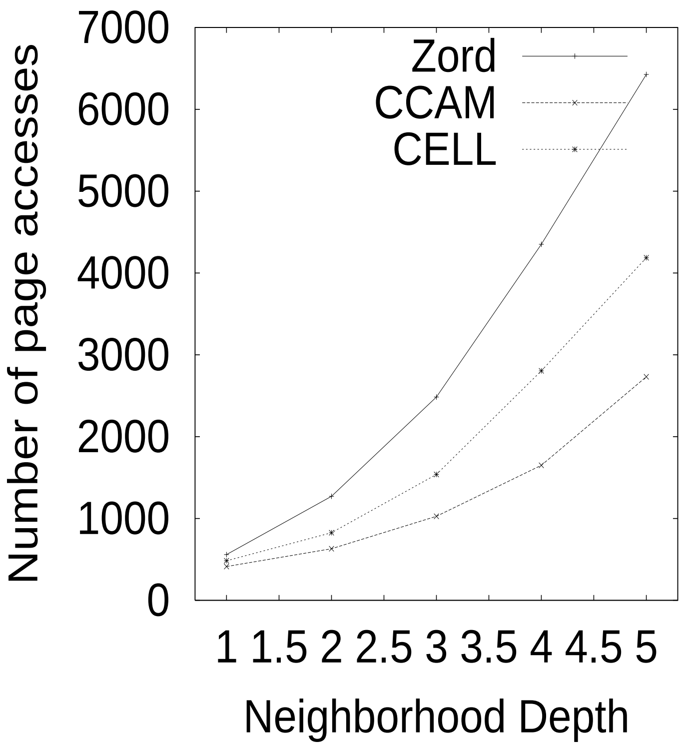


Figure 10: Effect of neighborhood depth on data page accesses

- *Increase neighborhood depth => High I/O cost*
- *CCAM has the best performance*

Random Point Test

- Fixed Parameters
  - *Random Point Size: 150 points*
  - *Data point size: 256 bytes*
  - *Page size : 1 K bytes*

- Variable Parameters
  - *Buffer Size: 2, 4, 6, 8 Kbytes*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*

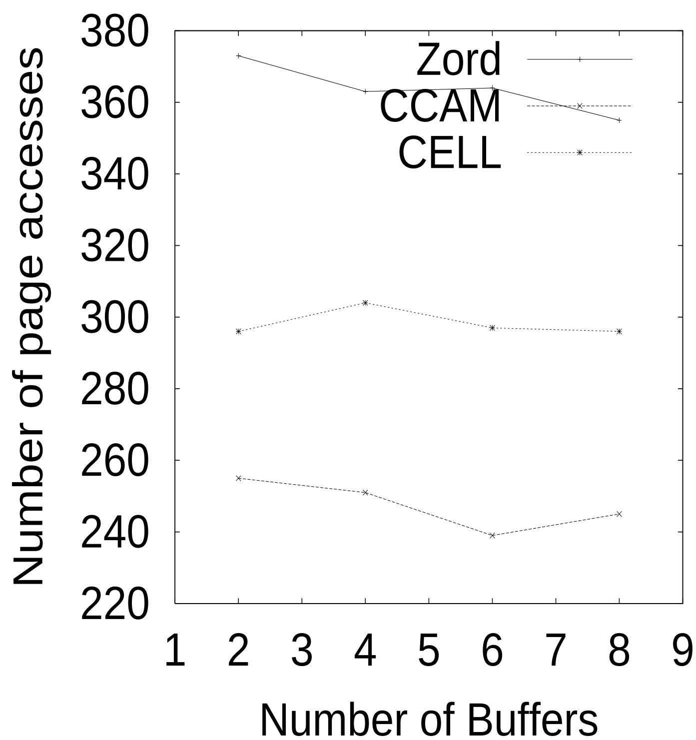


Figure 11: Effect of buffer size on the number of page accesses

- *Increase buffer size => No effect*
- *CCAM has the best performance*

Random Point Test

- Fixed Parameters
  - *Sample Point Size: 150 points*
  - *Data point size: 256 bytes*
  - *Page size : 1 K bytes*
  - *Buffer size: 4K*
- Variable Parameters
  - *Page size: 0.5, 1, 2 Kbytes*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*

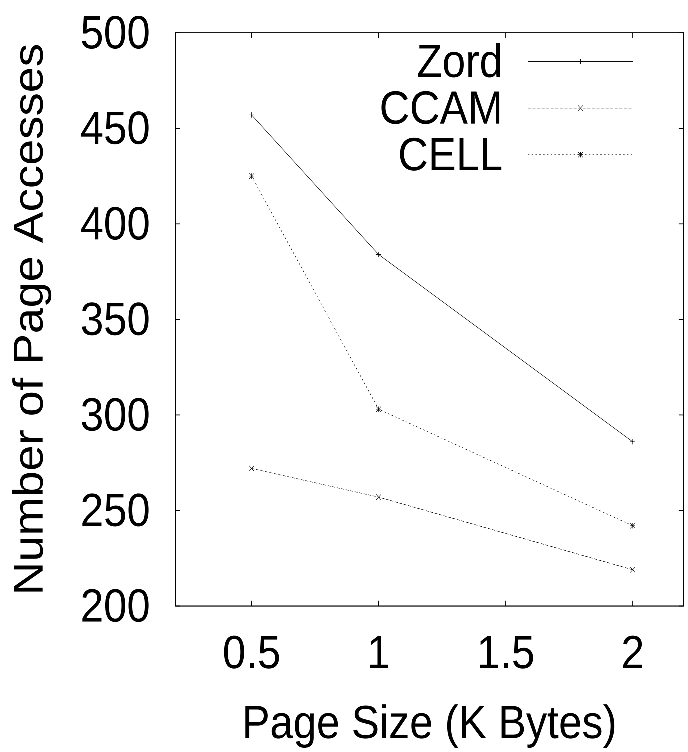


Figure 12: Effect of page size on the number of page accesses

- *Increase page size => reduce no of page accesses*
- *CCAM has the best performance*

Detect Outlier (Route)

- Fixed Parameters
  - *Highway - 35W N*
  - *Data point size: 256 bytes*
  - *Page Size: 1k*
  - *Clustering strategy: CCAM*
  - *CRR value : CCAM = 0.68 Cell = 0.53 Z = 0.31*
- Variable Parameters
  - *Buffer number: 2,4,6,8,10*
  - *Buffering Strategy: LRU, MRU, FIFO*

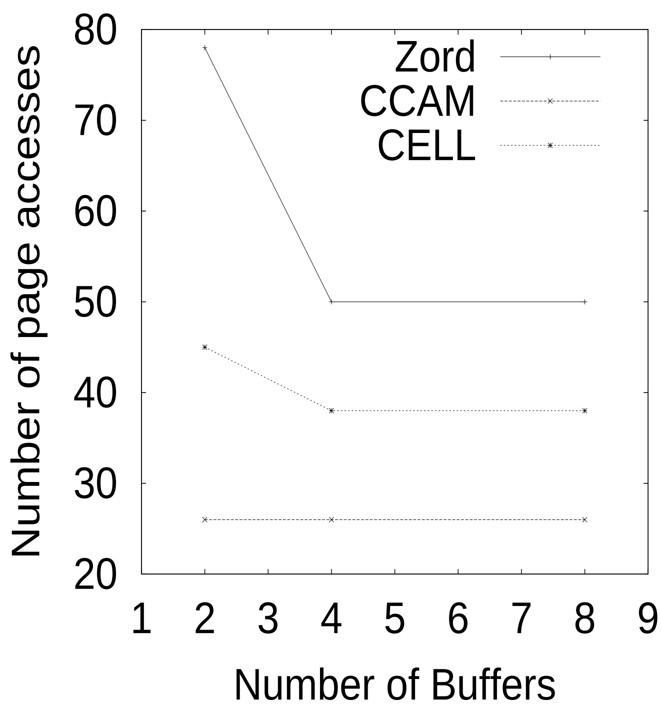


Figure 13: Effect of buffering on data page accesses (Block Size = 2K)

- *Increase buffer size => reduce no of page accesses*
- *CCAM has the best performance*

Detect Outlier (Route)

- Fixed Parameters
  - *Highway - 35W N(62 stations)*
  - *Page Size: 1k*
  - *Buffer size: 3K*
  - *Buffering Strategy: LRU*
- Variable Parameters
  - *Neighborhood Depth: 1, 2, 3, 4, 5*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*

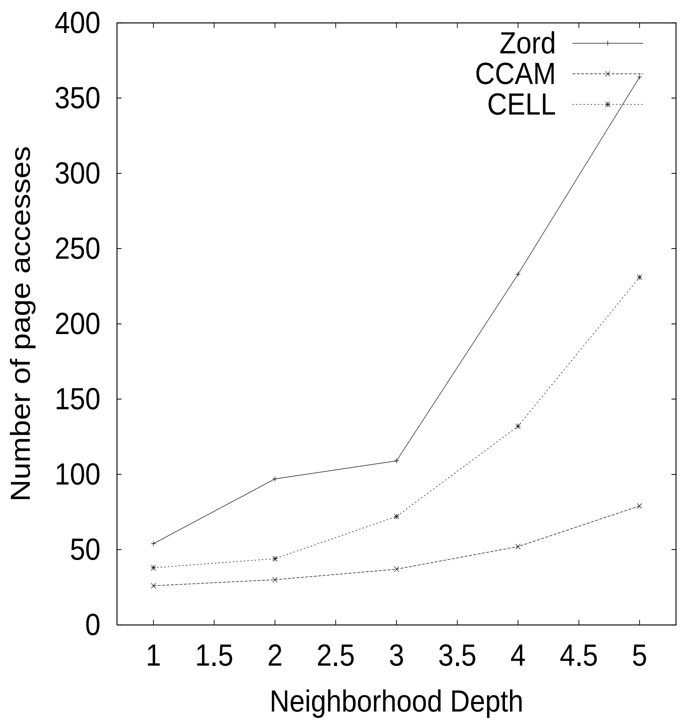


Figure 14: Effect of neighborhood depth on data page accesses

- *Increase neighborhood depth => increase no of page accesses*
- *CCAM has the best performance*

Detect Outlier (Route)

- Fixed Parameters
  - *Data point size: 256 bytes*
  - *Buffering Strategy: LRU*
  - *Buffer Size: 4 Kbytes*
  - *Highway - 35W S (64 stations)*
- Variable Parameters
  - *Page size : 0.5K, 1K, 2K*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*

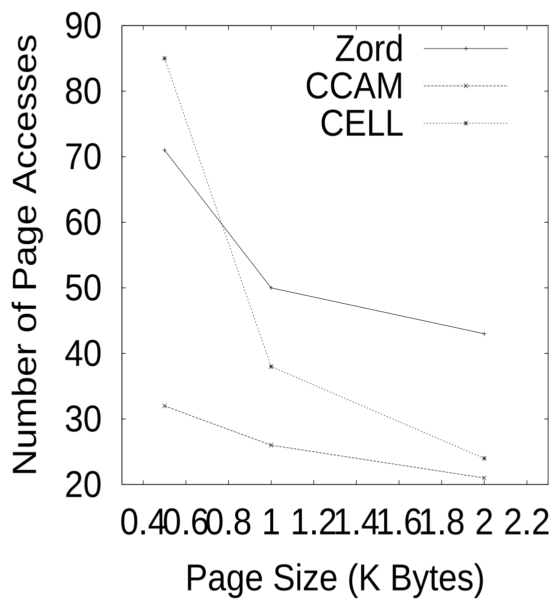


Figure 15: Effect of block size on data page accesses

- *Increase page size => reduce no of page accesses*
- *CCAM has the best performance*

Detect Outlier (Route)

- Fixed Parameters
  - *Data point size: 256 bytes*
  - *Buffering Strategy: LRU*
  - *Buffer Size: 4 Kbytes*
  - *Highway - 35W S (64 stations)*
- Variable Parameters
  - *Page size : 0.5K, 1K, 2K*
  - *Clustering strategy: CCAM, Cell Tree, Z-order*
- Measure
  - *CRR value*

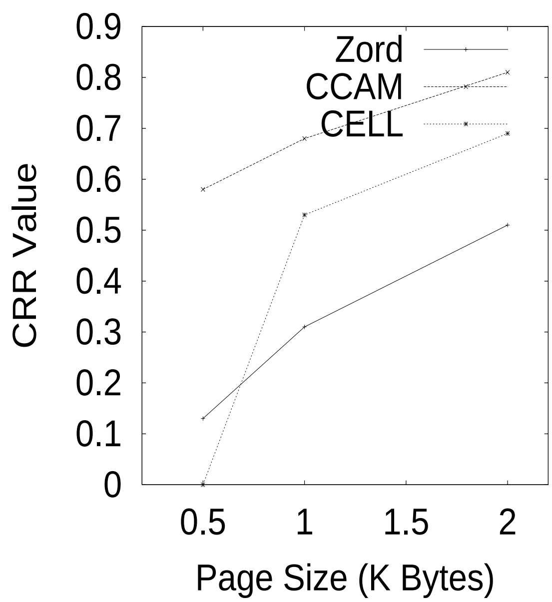


Figure 16: CRR value for differt page size

- *Cell Tree has zero CRR value when Bfr=2*
- *CCAM has the highest CRR value*

Detect Outlier (Area)

- Fixed Parameters
  - *Area - 64 Stations*
  - *Data point size: 256 bytes*
  - *Page Size: 1.5K*
- Variable Parameters
  - *Buffer number: 2,4,6,8*
  - *Clustering strategy: CCAM, Cell, Z-ord*

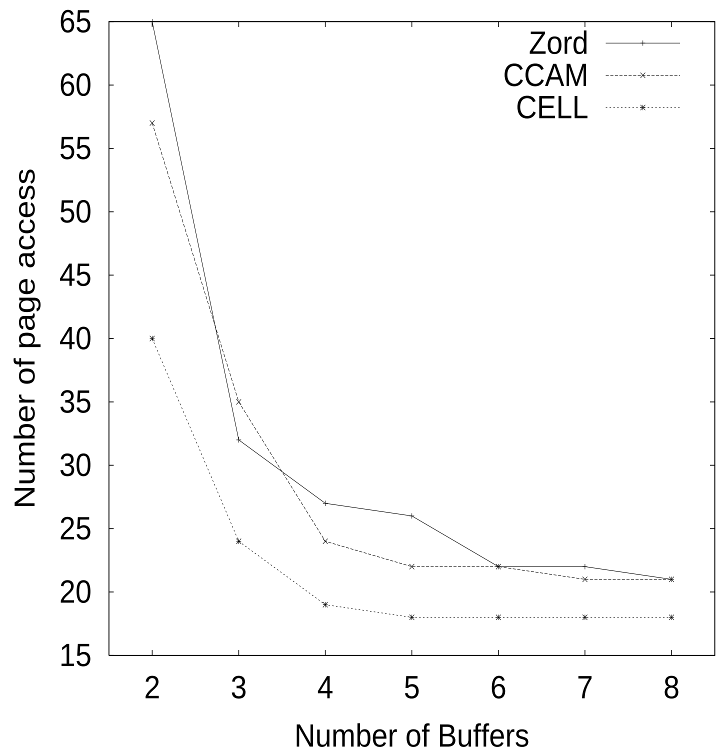


Figure 17: Effect of buffering on data page accesses (Block Size = 1.5K)

- *Increase buffer size => reduce no of page accesses*
- *Cell tree has the best performance*

Detect Outlier (Area)

- Fixed Parameters
  - *Area - 64 Stations*
  - *Data point size: 256 bytes*
  - *Page Size: 1.5K*
  - *Buffer number: 3*
- Variable Parameters
  - *No of stations*
  - *Clustering strategy: CCAM, Cell, Z-ord*

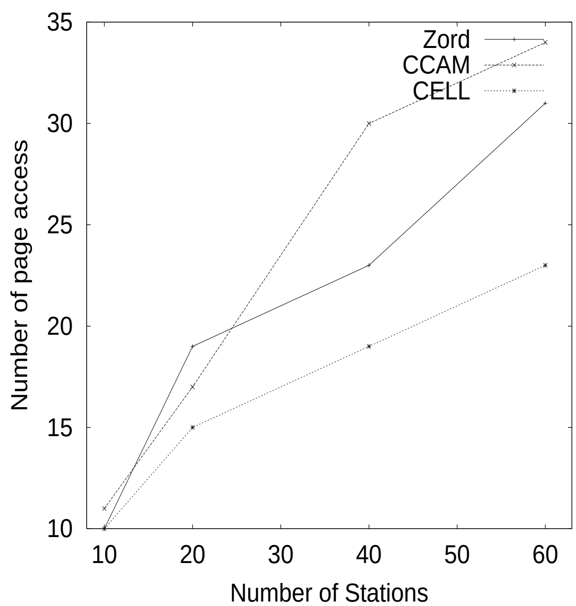
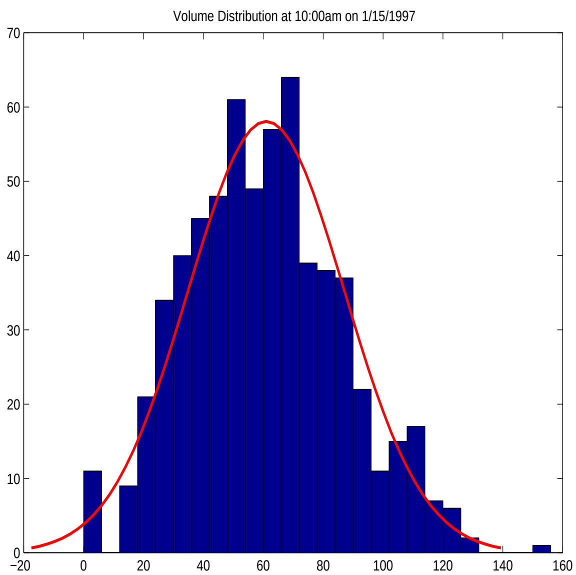


Figure 18: Effect of area size on data page accesses

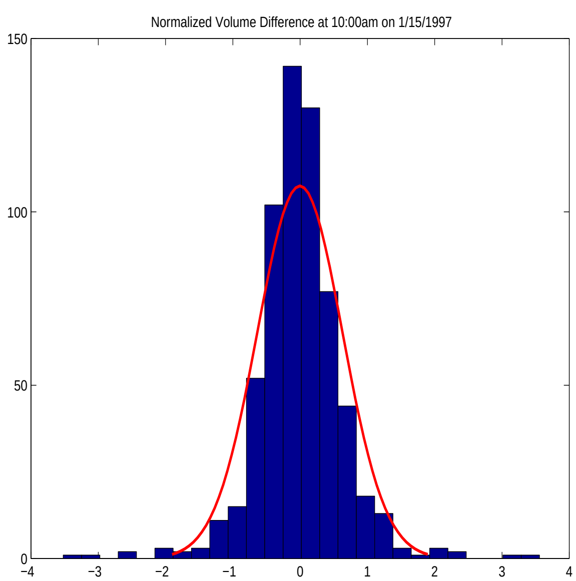
- *No of stations*
- *Clustering strategy: CCAM, Cell, Z-ord*
- *Increase query area size => increase no of page accesses*
- *Cell tree has the best performance*

Model Construction (An example)

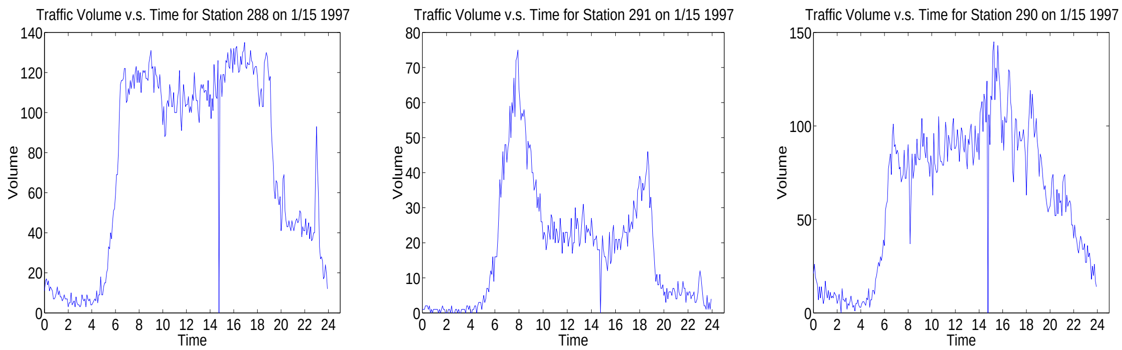
- *Distribution of highway traffic volume and normal distribution curve*



- *Distribution of difference for each station and its neighbors*

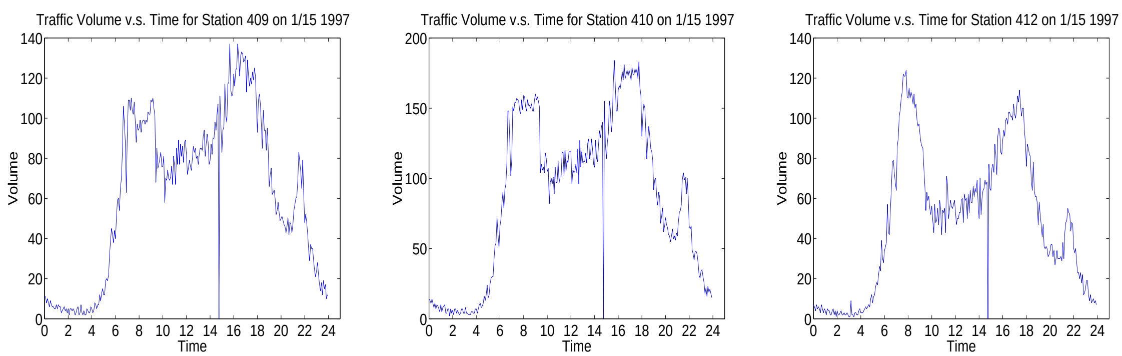


Local Topological Approach: Temporal Outlier Example



(a) Station 288 (b) Outlier Station 291 (c) Station 290

Figure 19: Outlier station 291 and its neighbor stations on 1/15 1997



(a) Station 409 (b) Outlier Station 410 (c) Station 412

Figure 20: Outlier station 410 and its neighbor stations on 1/15 1997

Local Topological Approach: Temporal Outlier Example

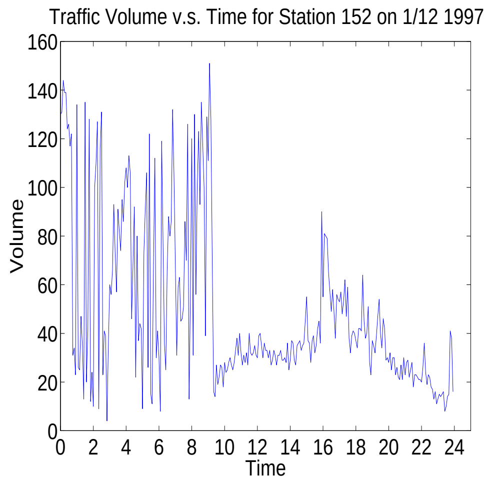


Figure 21: Temporal Outlier station 152 on 1/12 1997

Local Topological Approach: Spatial Outlier Example

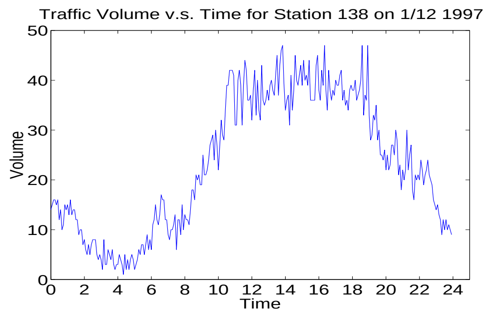


Figure 22: Station 138 on 1/12 1997

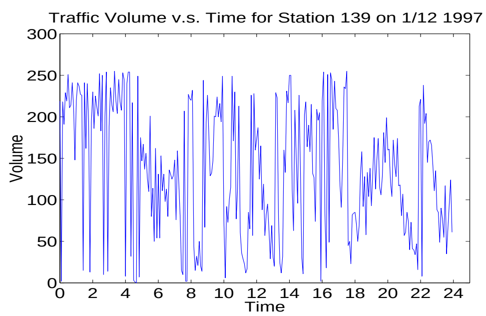


Figure 23: Station 139 on 1/12 1997

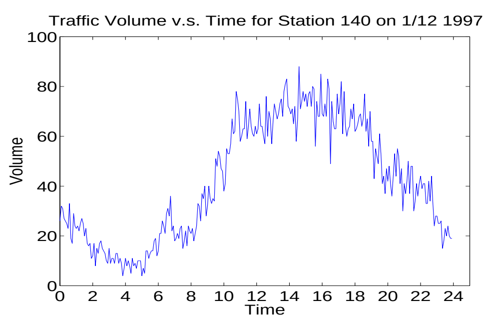


Figure 24: Station 140 on 1/12 1997

Summary

- Our approach
  - *Consider distances in both topological space and attribute space*
  - *Apply distribution to detect outliers*
- Our contribution
  - *Propose neighborhood-based outlier detection approach*
  - *Develop an efficient algorithm*
  - *Analyze performance for different queries*
    - *Model Construction*
    - *Random point test*
    - *Detect outliers along a route*
    - *Detect outliers within a region*
  - *Analyze performance for different parameters*
    - *Physical data record clustering*
    - *Buffering Strategies*
    - *Page size*
    - *Buffer size*
    - *No. of neighbors*
    - *Neighborhood depth*
- Other suggestion?