Creative Commons Attribution 4.0 International (CC BY 4.0)
https://creativecommons.org/licenses/by/4.0/

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

International Journal of Data Warehousing and Mining
Volume 18 • Issue 1

Density-Based Spatial Anomalous Window Discovery

Prerna Mohod, University of Maryland, Baltimore County, USA
Vandana P. Janeja, University of Maryland, Baltimore County, USA*
https://orcid.org/0000-0003-0130-6135

ABSTRACT

The focus of this paper is to identify anomalous spatial windows using clustering-based methods. Spatial anomalous windows are the contiguous groupings of spatial nodes which are unusual with respect to the rest of the data. Many scan statistics-based approaches have been proposed for the identification of spatial anomalous windows. To identify similarly behaving groups of points, clustering techniques have been proposed. There are parallels between both types of approaches, but these approaches have not been used interchangeably. Thus, the focus of the work is to bridge this gap and identify anomalous spatial windows using clustering-based methods. Specifically, the authors use the circular scan statistic-based approach and DBSCAN-density-based spatial clustering of applications with noise to bridge the gap between clustering and scan statistics-based approach. They present experimental results in US crime data. The results show that the approach is effective in identifying spatial anomalous windows and performs equal to or better than existing techniques and does better than pure clustering.

KEYWORDS

Anomaly Detection, Density-Based Clustering, Scan Statistics

1. INTRODUCTION

The focus of this approach is to identify anomalous spatial windows using clustering-based methods. Anomalous spatial windows (Shi and Janeja 2009, Janeja and Atluri 2008, Das and Schneider 2007) are formed by a set of contiguous spatial nodes which are unusual with respect to the rest of the data. Many scan statistics based approaches (Kulldorff 1997, Kulldorff et al. 2007) have been proposed to identify such windows. Clustering on the other hand is used to identify similarly behaving groups of nodes. There are many parallels in these two classes of techniques. For example, the spatial circular scan statistic method (Kulldorff 1997) scans the region with a circular shape while intrinsically is looking for spatial proximity, which some clustering methods also look for. However, scan statistics based methods do not look at the similarity of other attributes as clustering methods would do. In this paper, we bridge this gap between scan statistic methods and clustering methods. There is a huge amount of work done in scan statistics as well as in clustering. But scan statistics type framework has not been applied to clustering methods yet, to discover spatial anomalous windows.

In this paper, we examine if the framework of anomaly detection could be applied to the spatial clustering methods. Particularly, we focus on adopting a density based clustering method (Ester et al. 1998) to identify spatial anomalous windows. Density is an intuitive way to represent spatial nodes.

DOI: 10.4018/IJDWM.299015 *Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

We use the density-based clustering method because it is very similar to the existing spatial circular scan statistics. Density based methods use the concept of dense neighborhoods to identify clusters which is similar to circular scan statistic (Kulldorff 1997) based method. Circular scan statistics uses a circle of a particular radius, as a starting point and then expands the circle to scan the region. Similarly, density-based clustering method uses a circular neighborhood which is preset and remains fixed. We apply the variable circular scan window to density-based neighborhood.

Motivating Domain: Location has a very significant impact on crime in the United States. While some places are nearly free of serious crime, others are plagued by some of the highest crime rates. It is therefore clear that the risk of being victimized by crime in the United States varies greatly from one location to another. Hence, we aim to find a group of locations that is unusual as compared to the rest of the data. Usually, the crime rankings that are available are based on cities of various populations, metropolitan areas or states. The 'crime rate' is considered to rank the spatial units (cities, metropolitan areas, or states). However, these rankings/approaches do not identify contiguous neighborhoods which are high in a particular type of crime say, murders. What we aim to find is a group of contiguous nodes or neighborhoods, instead of just individual spatial unit, which are unusual with respect to the rest of the data.

Scan statistics based methods can be used to address this problem. Similarly, there are several clustering methods that can be used to identify unusually shaped clusters. However, they do not identify the unusual nature of the clusters as compared to the rest of the data. Thus, we modeled clustering as a scan statistics problem to bridge the gap between clustering and scan statistics framework.

We specifically take DBSCAN-Density based Spatial Clustering of Applications with Noise (Ester et al. 1998), as our clustering model. DBSCAN is an established spatial clustering method that has acquired good results and reviews in literature (Janeja and Palanisamy 2012, Han 2005) and has several recent uses (Mungekar et al. 2021, Shaojun et al. 2022, Sheridan et al. 2020, Xie and Shekhar 2019) showing that it has proven the test of time. It relies on a density-based notion of clusters, which is designed to discover clusters of arbitrary shape. Hence, we adapt DBSCAN to find unusual shaped spatial windows and identify if these windows are unusual with respect to the rest of the data. DBSCAN addresses several limitations of scan statistics based methods.

In circular scan statistic based method (Kulldorff 1997) some locations that are actually anomalous might not be identified as anomalous because they do not fall in the circular scan window. Conversely, some locations that are not anomalous might be recognized as anomalous because they fall in the circular scan window (Tango 2021). This lowers the accuracy of the method. We address this limitation by retaining the density reachable feature in the DBSCAN algorithm that helps identify windows with unusual shapes.

We make the following key **contributions:**

- We bridge the gap between scan statistics based approach and spatial clustering method,
- We present algorithms for univariate data that find arbitrary shaped anomalous windows,
- We present several variations on the algorithms to adapt the DBSCAN parameter settings so that we do not have to provide a single value for the input parameters,
- We present detailed experimental results in real world crime datasets for each of the variations of algorithms as well as for circular scan statistic based method (Kulldorff 1997) and DBSCAN (Ester et al. 1998), to illustrate the efficacy of our approach. In addition, we validate our ðndings with certain known crime trends in these real world datasets, indicating the efficacy of our approach. Our results indicate that we generally attain equal or high accuracy as compared to the scan statistics based approaches and higher accuracy than pure clustering approach, in real world crime data.

The rest of the paper is organized as follows: Section 2 discusses the related work in the areas of density based clustering and spatial scan statistics. We also discuss some background of the approaches

that we adapted. In section 3, we discuss our methodology for density based spatial anomalous window discovery using brute force and efficient method and outline multiple variations of algorithms. In section 4, we discuss detailed experimental results for all our algorithms in US crime data. Finally, in section 5, we summarize our work and discuss future work.

2. BACKGROUND AND RELATED WORK

Circular based Scan Statistic Method: Spatial scan statistics is used to detect the anomalies in a spatial region. This is done by imposing windows on the spatial region and identifying a group of contiguous nodes. One well accepted approach is the circular scan statistic approach (Kulldorff 1997). Let us say there are some spatial nodes dispersed in space and each node is associated with the attribute of interest such as number of robberies and the base attribute such as the total population. A circular window is centered on each node in the region. For each node, the radius of the window is varied continuously from zero to some upper limit specified by the user. The circular window varies the radius to capture the most anomalous window. Likelihood ratio for each of these windows is computed. It is important to note that nodes within each circle are a potential anomalous window (Janeja and Palanisamy 2012, Kulldorff 2010). Finally, a window with the maximum likelihood ratio is selected as the most unusual spatial window.

DBSCAN (Density Based Spatial Clustering of Applications with Noise): Clustering is the task of grouping the objects of a database into meaningful subclasses (clusters) so that the members are as similar as possible whereas the members of different clusters differ as much as possible from each other. DBSCAN relies on a density-based notion of clusters which is designed to discover clusters of arbitrary shape (Ester et al. 1998). We first discuss the input parameters of DBSCAN. This algorithm needs two input parameters, viz. ε (Epsilon) and Minimum Points (Minpts). For our work, we use spatial nodes so we will refer to this as Minimum Nodes (Min-nodes) henceforth. ε is the radius defined to identify the neighborhood in which the minimum number of nodes are to be found, for this to be a sufficiently dense circle. DBSCAN was first discussed in 1998 since then several variations and uses of this algorithm have been discussed. Recent works have used DBSCAN for several applications such as crime analysis (Mungekar et al. 2021), fault diagnosis (Shaojun et al. 2022), flight anomaly detection (Sheridan et al. 2020), robust clustering (Xie and Shekhar 2019). These recent papers show the test of time for the DBSCAN clustering method.

Other Related Work: Clustering is a key technique used to find groupings in the data. Most of the clustering techniques like DBSCAN, OPTICS and (Harel and Koren 2001) identify the major groupings in the data but fail to identify the unusual ordered groupings. Thus, clusters are formed but we cannot conclude if any of the clusters is necessarily unusual as compared to the rest. Other outlier approaches identify individual outliers with respect to the rest of the objects in the dataset. In addition, the other approaches do not identify unusual group of objects nor do they quantify them. Recent approaches (Emadi 2018, Zhang 2018, Amarbayasgalan 2018) have used density based methods for anomaly detection in various applications.

Traditional outlier detection approach is not suitable for discovering unusual group of objects and they do not identify and quantify the anomalous window (Ester et al. 1998, Alimohammadi et al 2022). Scan statistic is a proven technique (Naus 1965) for discovering the set of unusual continuous nodes and quantifies them by calculating local statistic (Youngser et al. 2010) for each window. Kulldorff et.al. proposed a spatial and spatio-temporal scan technique by scanning the spatial region using the circular or cylindrical (for spatio-temporal data where the height is the time frame) window (Kulldorff 1999). In the Poisson based model (Kulldorff 1997), they compare the number of cases inside the circle with the number of cases outside the circle. The proposed approach takes the location and case information as an input and finds the high or low rate incidences in the given single domain dataset. While spatial scan statistics is a well-received technique, it also has some major issues in terms of identifying free form, flexible and irregular shaped anomalous windows (Tango 2021). We

propose to use DBSCAN to address this challenge so the cluster shapes can determine the shape of the anomalous window rather than including surrounding areas in the circular shaped scan windows.

3. ANOMALY DETECTION WITH DBSCAN (ADD)

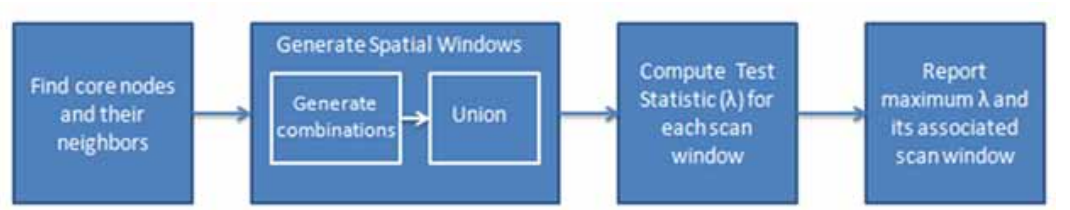
In this section, we present our approach to discover spatial anomalous windows by adapting the DBSCAN clustering technique. We propose a class of algorithms called Anomaly Detection with DBSCAN (ADD).

3.1 ADD (Anomaly Detection with DBSCAN)

We adapt the traditional DBSCAN algorithm to detect spatial anomalous window. The traditional DBSCAN algorithm (Ester et al. 1998) expands the core node cluster by finding the density reachable nodes. In our approach shown in figure 1, we first form the clusters by finding only the directly density reachable nodes. Within the directly density reachable region for each core node, we find several sub-sets of potentially anomalous windows by taking combinations. We then form unions from these combinations to find bigger, potentially anomalous sub-sets in the neighborhood.

As discussed in section 2, the DBSCAN algorithm needs two input parameters, viz. ε (Epsilon) and Minimum Nodes (Min-nodes). In the traditional DBSCAN (Ester et al. 1998) algorithm these parameters have to be provided as an input. For our purposes, we use variations to how the input parameters are provided to our (ADD) anomaly detection with DBSCAN algorithm.

Figure 1. *ADD approach*



In the basic ADD approach, we first find all core nodes using pure DBSCAN algorithm and identify each node's corresponding neighbors. We call this set as "Core Node and Neighbors Set". Next, we generate a "Combination Set" by generating combinations based on Core Node and Neighbors Set. Then, we form unions to generate "Union sets". Each of the entries in these sets forms a spatial window or what we refer to as 'scan window'. We compute the test statistic for each scan window, in all these sets. Then, we report the scan window with the highest value of the test statistic. This is our spatial anomalous window. We present the algorithm 1 for ADD in Appendix A and next outline each step in detail.

3.1.1 Find Core Nodes and Their Neighbors

We first define a spatial node:

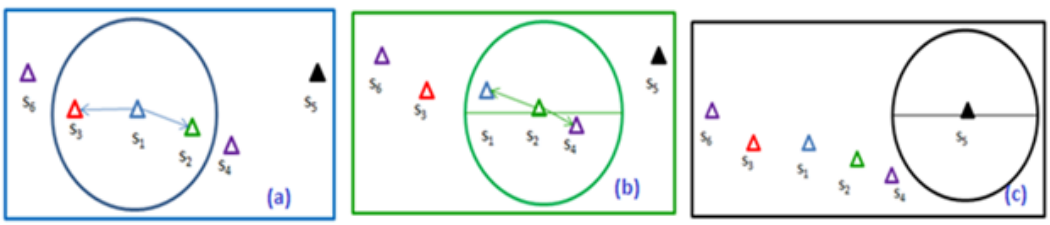
Definition 1: *[Spatial Node]* (Janeja and Palanisamy 2012) S is a set of spatial locations $\{s_1,\ldots,s_n\}$ which we call as Spatial Nodes where each spatial node s_i is associated with coordinates $\{s_{ix}, s_{iy}\}$ i.e. longitude, latitude and non-spatial attributes$\{a_{i1},\ldots,a_{ik}\}$.

One or all these non-spatial attributes are called "*attribute of interest*" referred to as in which we want to measure the unusualness.

We first find all the core nodes and their corresponding neighbors i.e., all the nodes that fall in the ε neighborhood of the core nodes using the basic DBSCAN algorithm. Example: Suppose that there are six spatial nodes in space viz. s_1, s_2, s_3, s_4, s_5, s_6 and the input parameters are: ε = 1 km and Min-nodes = 2.

We start traversing the region with a spatial node. For example, let us say we start with node s_1. Spatial nodes s_1, s_2, s_3 lie in node s_1's ε neighborhood i.e., within 1 km distance from s_1 as shown in figure 2 (a). Thus, s_1 is a core node. Similarly we visit other points, as shown in figure 2 (Youngser et al.) and (c).

Figure 2. *Example of finding Core Node*



We call the set of core nodes and their neighbors as 'Core Node and Neighbors Set'. Each core node represents a scan window as well. In addition, each of the scan windows comprising of contiguous nodes in the 'Core Node and Neighbors set' is a potential anomalous window. The Core Node and Neighbors Set for our example is shown in table 1 below. Note that node s_5 is not listed in the table as it is not a core node, but we do consider every node as its own window, so it will form the window as well. Now, we use these core nodes and their neighbors to generate other potential anomalous scan windows which may be subsets or supersets of these windows provided they are contiguous.

We define an anomalous scan window as:

Definition 2: *[Anomalous Window] (Janeja and Palanisamy 2012*Given a set of spatial nodes *S*, an anomalous window a*w* = $\{s_1, \ldots, s_m\}$,where a*w S* and each s_i, s_j *sw* are *spatial nodes* and there exists spatial relationship $sr(s_i, s_j)$, such that *aw* deviates from *S-aw* in terms *of* non-spatial attribute of interest .

We refer to the horizontal entries in the table as 'scan windows' throughout this paper. The core and Neighbor set is discovered using the basic DBSCAN algorithm.

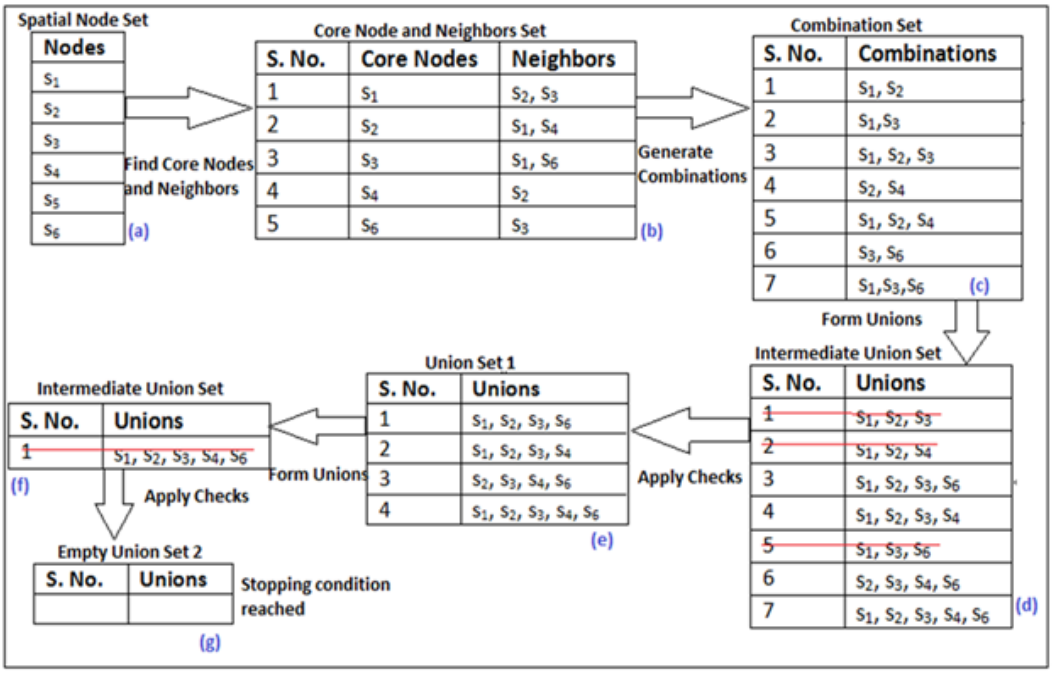
Table 1. Core nodes and neighbors sets

S. No	Core nodes	Neighbors
1	s_1	s_2,s_3
2	s_2	s_1,s_4
3	s_3	s_1,s_6
4	s_4	s_2
5	s_6	s_3

3.1.2 Generate Spatial Windows

In this step, we generate the Combination Set and Union Sets that comprise of potentially anomalous spatial windows using the methods explained below. We outline the entire process, in the figure 3:

Figure 3. *Summary of Generation of All Sets*



3.1.2.1 Generate Combinations

In this sub-step, various scan windows are obtained by generating combinations of nodes from Core Node and Neighbors Set. The newly formed Set is "Combination Set".-There are two ways to do this (a) brute force: by checking for combinations and then checking if they are contiguous nodes, and (Youngser et al efficient. We discuss here an efficient way to do this

In theory, what we want is the core node to be a part of every combination we make. By doing this, we ensure that all the nodes in a combination are neighbors i.e., they are contiguous. Because we account for the contiguity of nodes while making combinations, there is no pruning required here as in the brute force approach.

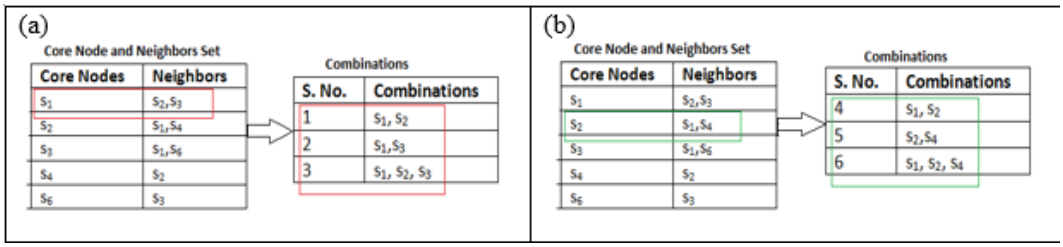
As shown in figure 4 (a), we first combine core node (s_1 with s_2); then core node (s_1 with s_3); and then core node (s_1 with both s_2 and s_3). Effectively, we take all possible combinations of only neighbors and add the core node to the combination to ensure contiguous spatial nodes.

Similarly, figure 4 (Youngser et al shows how all the combinations are obtained for the second scan window from the Core Node and Neighbors Set. Similarly, we visit all the scan windows in the Core Node and Neighbors Set, to generate combinations. We eliminate duplicates before adding the combinations to the 'Combination Set'.

Figure 3 (c) shows all the combinations after eliminating duplicates. We call this the 'Combination Set'. It is important to note that there will be only one 'Combination Set'. We present the algorithm

2 to generate the 'Combination Set' by taking combinations of scan windows from Core Node and Neighbors Set in Appendix A.

Figure 4. *Example of Efficient Method*



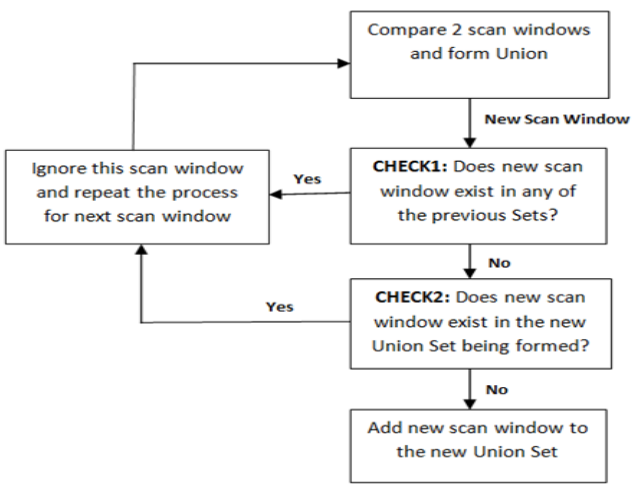
3.1.2.2 Form Unions

The second sub-step in generating spatial windows is forming unions as shown in figure 5. Unions are formed using two scan windows in a set at a time. This sub-step is an iterative process until the stopping condition is reached and hence multiple Union Sets could be generated in this sub-step.

In the process of forming unions, we compare every single scan window with all the subsequent scan windows in the given set and take a union (unique nodes) from each. These newly formed unions are the candidate scan windows for the final Union Set and potentially anomalous. We check if each newly formed scan window already exists in any of the previous sets or in the new Union Set (being formed). If it exists, we ignore it and move on to check the next scan window. If it does not, we add it to the new Union set. We continue to do so until the stopping condition is reached. This process is illustrated in the block diagram below:

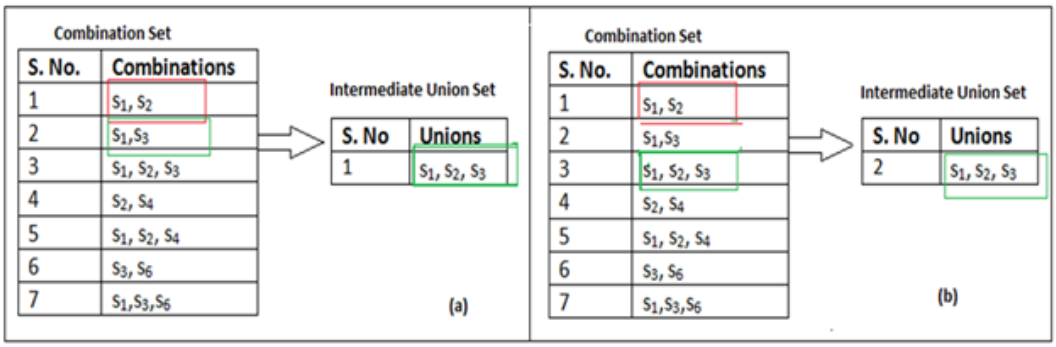
We explain the process of forming unions as well as the stopping condition with the help of example below.

Figure 5. *Checks applied after a new scan window is formed by Union*



Example: We first consider scan windows 1 and 2 from the Combination Set as shown in figure 6 (a) below. They consist of nodes (s_1, s_2) and (s_1, s_3). We take union of both i.e., nodes (s_1, s_2, s_3). We check if this new scan window (s_1, s_2, s_3) already exists in the Core Node and Neighbors Set or the Combination Set. It does in the Combination Set and hence we move on to compare scan windows 1 and 3. As shown in 6 (Youngser et al.), we then take union of scan window 1 with 3 and all the subsequent scan windows; form intermediate scan windows and add to the final Union Set only if the new scan windows do not exist in any of the previously made sets.

Figure 6. *Example of Forming Union*



Next, we compare scan window 2 in the Combination Set with all the subsequent entries, to form unions. We compare each scan window only with the subsequent scan windows because comparing scan window 1 with scan window 2 or comparing scan window 2 with scan window 1 would both result in the same union. Hence, to avoid computation and clear-cut duplicates, we compare a given scan window only with subsequent scan windows. Thus, we compare all the scan windows with subsequent scan windows in the 'Combination Set' to obtain "Union Set 1". Subsequent Union Sets if any are formed by taking Unions of scan windows from preceding Unions Sets. The figure 3 (d) shows intermediate unions (with duplicates removed) obtained from the Combination Set and figure 3 (e) shows the final Union Set 1 obtained after applying checks.

We repeat the process of forming unions until there are no more scan windows to be added to the subsequent Union Set. Thus, we reach the stopping condition. This is similar to the apriori method for generating frequent item sets (Agrawal et al. 1993).

Figure 3 (f) above shows the intermediate unions formed from 'Union Set 1' and after applying checks we see in figure 3 (g) that there are no scan windows to be added to 'Unions Set 2'.

Thus, the stopping condition for the process of generating Union Sets is when there remains no window to be added to the subsequent Union Set. At this point, all possible windows of contiguous nodes have been accounted for. The entire process of finding the ' Core Node and Neighbors Set', 'Combination Set' and 'Union Sets until the stopping condition is reached is represented in figure 3 above.

We present the algorithm to form unions in algorithm 3 in Appendix A

We next check if each of the windows is anomalous by computing the unusualness test statistic.

3.1.3 Compute Unusualness Test Statistic (λ) for Each Scan Window

We then compute a test statistic λ for each of the scan windows, in the Core Node and Neighbors Set, Combination Set and all the Union Sets. For the Scan window *sw* the test statistic $\lambda_{sw} = (/)/(/)$ where is the value of our behavioral attribute of interest for the spatial nodes within the scan window and is

the total population of the spatial nodes within the scan window. and are their corresponding values outside the scan window which are computed by (−) and (−). is the total value of our behavioral attribute of interest and is the total population.

For example, in the crime motivating example, 'murders' or 'combined crime' are our attributes of interest and base attribute is the total population of the region. This test statistic λ indicates the degree of unusualness of the scan window as compared to the rest of the data outside of the window in terms of the attribute of interest. The higher the λ, the greater is the unusualness of the scan window. The scan window with the highest test statistic λ is ranked as the most anomalous window.

For our example above, we compute λ for all the scan windows shown in table 2.

Next we report the maximum of all the λ's computed and its corresponding scan window. As discussed above, this will be the most unusual window, or the anomalous scan window as defined below.

Table 2. Name of set and number of scan windows

Name of Set	Number of Scan Windows
Core Node and Neighbors Set	5
Combination Set	7
Union Sets	4
Individual Nodes	6
Total	**22**

Definition 3: *[Anomalous Scan Window] (Janeja and Palanisamy 2012*Given a set of scan window $SW = \{sw_1,\ldots,sw_z\}$and their associated test statistics λ, the anomalous window is the one with maximum λ (λ_{max}).

For the variations of our ADD approach, we also report the ε and/or Min-nodes at which λ_{max} is found. We next discuss variations of the ADD approach. The different algorithms differ in how we vary the input parameters and are shown in Appendix A algorithm 4,5,6 where we depict the changes to the basic algorithm in blue color.

VE-ADD (Variable Epsilon): only the ε is varied keeping the Min-nodes fixed. λ_{max} is the maximum value of test statistic across all variations of ε's.

VM-ADD (Variable Min-nodes): only the Min-nodes is varied keeping the ε fixed. The important thing to note here is that λ_{max} is the maximum value of test statistic across all variations of Min-nodes.

VEM-ADD (Variable Epsilon, Min-nodes): both ε and Min-nodes are varied. The important thing to note here is that λmax is the maximum value of test statistic across all variations of ε's as well as Min-nodes.

ED-VE-ADD (Euclidian Distance based with Variable Epsilon): This variation and its algorithm are the same as the VE-ADD, except that the Euclidian distance metric is considered while computing the ε neighborhood. In all the above variations, the distance between two nodes is computed in Kilometers. In the ED-VE-ADD approach, the Euclidian distance between all the 4 variables (viz. X, Y, attribute of interest, base population) is computed.

We next discuss the detailed experimental results.

4. EXPERIMENTAL RESULTS

We report results obtained from experimental data as described below, and then compare and evaluate our methods with existing methods in clustering (Ester et al. 1998 and circular anomalous window detection method (Kulldorff 1997.We first describe our dataset and then discuss our experiments and results in details.

4.1 Datasets

The dataset used for this paper is the 'Communities and Crime Data Set' (Redmond 2011This dataset combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. For us to be able to compute the ratio-based test statistic to find the anomalous window, we got the unnormalized data from Dr. Redmond.

This dataset has 2215 instances of communities data across the US. Many of the counties and state codes have missing values, rendering only 991 instances to be useful. We aggregated these instances at the county level. Thus, we have 301 instances of counties for about 26 states. United States is divided into 4 regions and sub-divided into 9 smaller divisions (US_Census_Bureau We base our experiments mainly on the Middle Atlantic division comprising of the states of New Jersey, New York and Pennsylvania, and the South Atlantic division comprising mainly of District of Columbia, Florida, Georgia, Maryland, Virginia(US_Census_Bureau

Out of the 284 attributes containing socio-economic data, we use only the violent and property crime attributes, apart from the identifiers and base population.

We also obtained the geographical co-ordinates of counties from (FIPS_Codes.We ran our experiments for all the 8 crime types –'combined crime', as our univariate variable. Our input file is in the format x co-ordinate, y co-ordinate, behavioral attribute of interest (combined crime), and population. We ran our experiments for the following 6 sub-parts for combined crime.

Adapting pure DBSCAN for comparison: Apart from the new approaches that we mention in section 3, we modify the pure DBSCAN algorithm by adding the test statistic computation to it and testing its efficiency as an anomaly detection algorithm. The test statistic is first computed for directly density reachable nodes (DDR). Then, as new nodes are added to the cluster in the process of expansion, the test statistic is computed for the directly density reachable nodes as well as the density reachable nodes (DDR and DR).

Table 3. Regions for which experiments are conducted

Region Name	Number of Nodes
NJ Northern	10
NY Northern	12
NY Southern	16
PA Allegheny	13
PA Rest (All except Alleghany)	16
VA Southern	17

4.2 Results

The results are discussed as follows:

- We present detailed maps for two regions here, showing the windows that each of the algorithms picked as most anomalous.
- We also present and discuss the experimental results for all regions based on accuracy, precision, recall and time. The accuracy is based on MQ Press (MQ_Home which publishes books on crime rates health care, education and other categories. We use the most dangerous city-wise crime rankings published by them (Crime_RankingsHowever, it is important to note that we use this simply as a test bed and not as a clear indication of a location being high in crime. For this, more domain input is needed.
- In each of the above we present comparative results with DBSCAN (Ester et al. 1998and circular spatial scan statistic based method (Kulldorff 1997

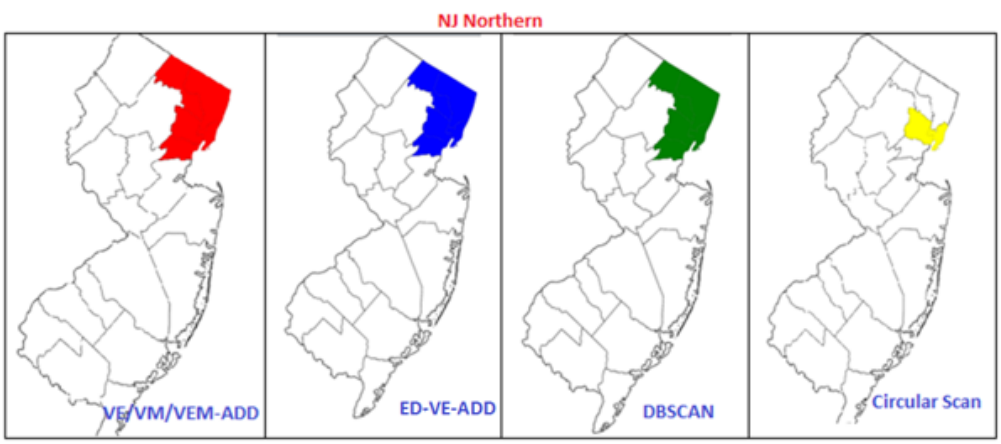
4.2.1 Anomalous Windows

Following each of the comparative maps are tables that show λ_{max}, ε at which λ_{max} was found, min and max ε for the experiments, for each of the algorithms. We also present the value of (log likelihood) test statistic for circular scan statistic along with the radius at which the circular window was found.

We want to point out that we obtain similar results with different variations for VE, VM and VEM-ADD. For each region, we obtained the ε at which we get λ_{max} using the VE- ADD approach. In the VM-ADD where ε is fixed, we used the ε that we obtained in the VE-ADD. Varying the Min-nodes does not make much difference because increasing the Min-nodes does not satisfy the core node criteria of ε neighborhood. Thus, if the density of a region is fairly uniform, VM and VEM-ADD converged at the same value of ε as VE-ADD.

We found that the variation in ε produces the most difference. Since results of VE, VM and VEM are the same, we represent all of them as a single result block. For space constraints we discuss map based results of only NJ and VA segment of the data.

Figure 7. Anomalous windows and related numbers for NJ northern region



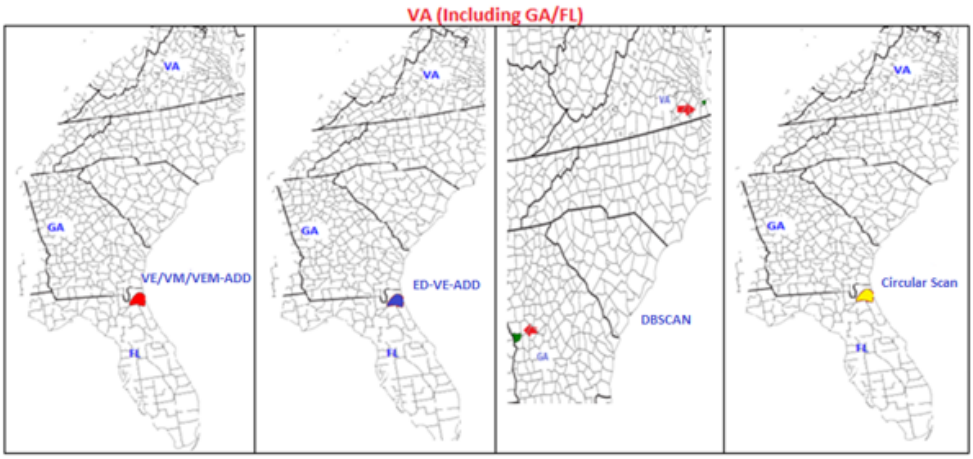
Algorithm	λ_{max}	ε	ε_{min}	ε_{max}
VE/VM/VEM-ADD	2.468	15 km	5 km	35 km
DBSCAN A-ADD	2.468	170,000	10,000	182,000
ED-VE-ADD	2.468	170,000	10,000	182,000

Algorithm	Likelihood Ratio	ε
Circular Scan	11688.66	13.06 KM

For the NJ Northern region, shown in figure 7, we see that all the algorithms except circular scan statistic based method get similar results finding counties of Essex, Hudson, Passaic, Union, and Bergen as the most anomalous window. Circular scan statistic based method finds only Essex and Hudson as the most anomalous. The anomalous window for this region based on CQ Crime Ranking consists of cities belonging to counties of Essex, Hudson, Passaic, and Union.

For VA Southern region, shown in figure 8, consisting of GA and FL, we see that all algorithms except DBSCAN find Duval County, FL to be the most anomalous. DBSCAN finds Muscogee, GA and Norfolk, VA to be the most anomalous (shown by red arrows in the above figure). According to the CQ Crime Ranking, Duval County is the most anomalous

Figure 8. Anomalous windows and related numbers for VA southern region



Algorithm	λ_{Cmax}	ε	ε_{Cmin}	ε_{Cmax}
VE/VM/VEM-ADD	1.59	0 km	5 km	50 km
DBSCAN A-ADD	1.08	90,000	10,000	167,000
ED-VE-ADD	1.59	0	10,000	167,000

Algorithm	Likelihood Ratio	ε
Circular Scan	4341.19	0 km

4.2.2 Accuracy/Precision/Recall and Validation with Crime Rankings

CQ- Congressional Quarterly (formerly Morgan Quitno) (MQ_Home MQ_acquiredpublishes the top 200 safe US city rankings every year. This is based on combined crime and not on any one crime type. This is the reason we discuss the results of only combined crime here. We referred to the list published in 1995 (Crime_Rankingsand used only the bottom 100 cities which are potentially the 100 most dangerous cities. For our experimental purposes, the cities found in these 100 are considered as known anomalies. We compare the results of our combined crime experiments for all the regions with these known anomalies and determine the accuracy matrix. We have not used the FBI website as the source of the data is FBI UCR and it would be a circular reference.

In this sub-section, we present and discuss the Accuracy, Precision and Recall numbers for different datasets, run for different algorithms. These metrics allow us to compare in a standardized manner the improvement in detection of true anomalies in the data as compared to other methods and baselines from crime rankings. We analyze the results with the help of graphs that follow the tables 4, 5, 6.

Table 4. *Accuracy*

Algorithm	NJ Northern	NY Northern	NY Southern	PA Allegheny	PA Rest	VA Southern
VE-ADD	90%	91.67	87.50%	76.92%	100%	70.58%
VM-ADD	90%	91.67%	87.50%	76.92%	100%	70.58%
VEM-ADD	90%	91.67%	87.50%	76.92%	100%	70.58%
ED-VE-ADD	90%	91.67%	87.50%	100.00%	100%	70.58%
DBSCAN	90%	83.34%	75%	69.23%	75%	64.70%
Circular Scan	80%	91.67%	100%	84.61%	100%	70.58%

Table 5. *Precision*

Algorithm	NJ Northern	NY Northern	NY Southern	PA Allegheny	PA Rest	VA Southern
VE-ADD	80%	100.00%	0.00%	25.00%	100%	100.00%
VM-ADD	80%	100.00%	0.00%	25.00%	100%	100.00%
VEM-ADD	80%	100.00%	0.00%	25.00%	100%	100.00%
ED-VE-ADD	80%	100.00%	0.00%	100.00%	100%	100.00%
DBSCAN	80%	50.00%	20%	0.00%	0%	50.00%
Circular Scan	100%	100.00%	100%	0.00%	100%	100.00%

Table 6. *Recall*

Algorithm	NJ Northern	NY Northern	NY Southern	PA Allegheny	PA Rest	VA Southern
VE-ADD	100.00%	50.00%	0.00%	100.00%	100.00%	16.00%
VM-ADD	100.00%	50.00%	0.00%	100.00%	100.00%	16.00%
VEM-ADD	100.00%	50.00%	0.00%	100.00%	100.00%	16.00%
ED-VE-ADD	100.00%	50.00%	0.00%	100.00%	100.00%	16.00%
DBSCAN	100.00%	50.00%	100.00%	0.00%	0.00%	16.00%
Circular Scan	50.00%	50.00%	100.00%	0.00%	100.00%	16.00%

In terms of accuracy (figure 9), precision (figure 10), and recall (figure 11), we can see that our algorithms consistently perform equal or better than DBSCAN. We also perform equal or better than circular scan statistic based method across all the datasets except for NY Southern and PA Allegheny.

Figure 9. *Accuracy*

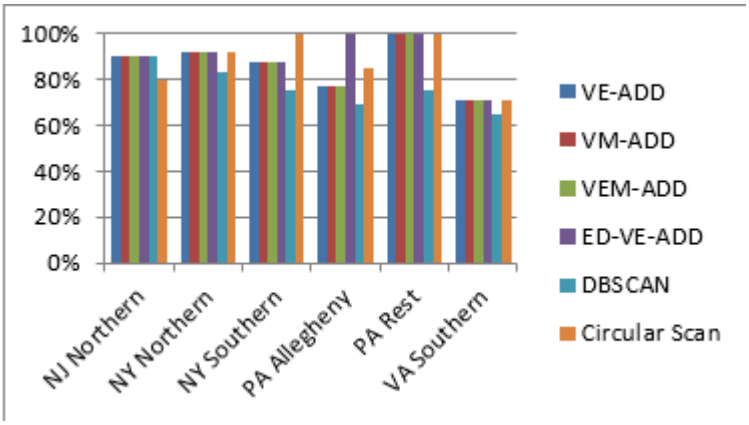


Figure 10. *Precision*

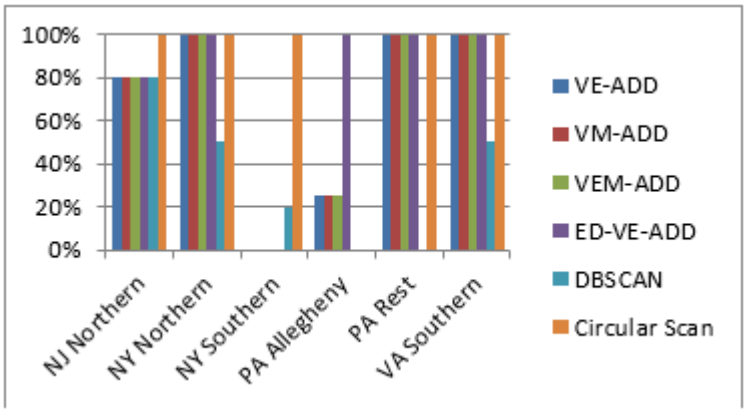
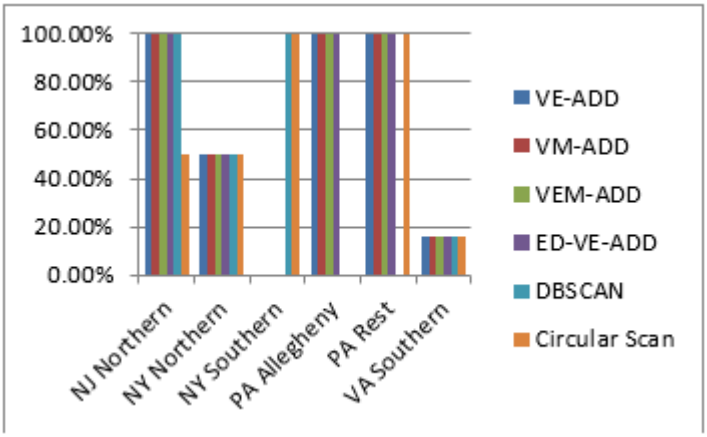


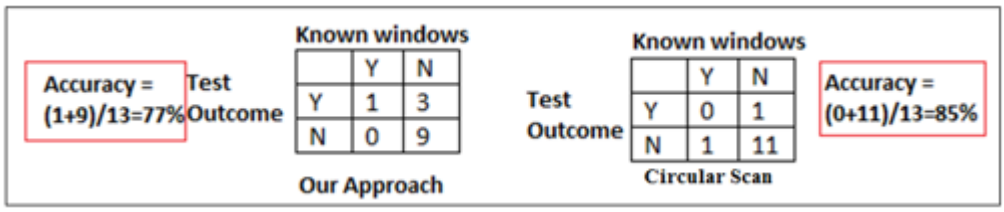
Figure 11. *Recall*



The precision and recall numbers for NY Southern for the circular scan statistic are both 100%. This means that circular scan statistic based method successfully detected the known anomaly 'Albany'. Our algorithm failed to identify 'Albany' but identified 'Tompkins' county as anomalous, hence our precision and recall are both 0%. What is interesting here is that 'Tompkins', which is highly unusual according to our approach, has a very large proportion of the total population as the attribute of interest, which is combined crime. Circular scan statistic based method finds 'Albany' to be the most unusual window and 'Tompkins' as the secondary window. Thus, both are reported by circular scan statistic based method. We do not compare all the secondary anomalies found by circular scan statistic based method, but looking only at this, we conclude that the anomalous window discovered is influenced by which test statistic is used.

Secondly, we see that the accuracy for PA Alleghany, shown in figure 12, for the circular scan statistic based algorithm is about 85% while for our algorithms it is about 77%. The accuracy matrix for our algorithm and circular scan statistic based algorithm is presented below:

Figure 12. *Accuracy Matrix for PA Alleghany*



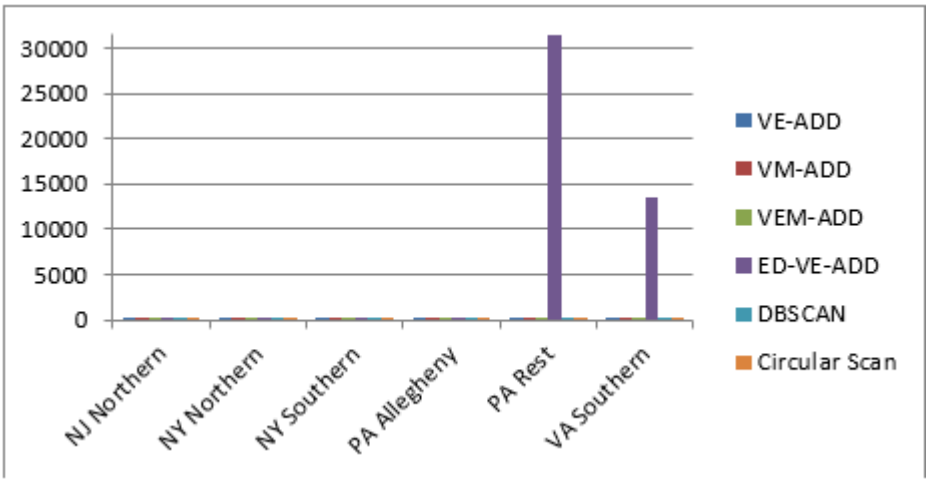
We see that despite our approach correctly identifying the one known window, our accuracy is lower. Conversely, though circular scan statistic based algorithm does not identify the one known window correctly, its accuracy is higher. This is because, though circular scan statistic based algorithm fails to identify one known anomaly, it correctly identifies the non-anomalies as N–N cell in the table, represented by the number 11 in figure 12. On the other hand, apart from correctly classifying the known anomaly, our algorithm identifies only 9 non-anomalies correctly out of 12. Thus, our accuracy is lower. But, if we look at the precision and recall numbers, both are 0 for circular scan statistic based algorithm, for PA Alleghany. Recall is 100% for our algorithms for correctly classifying one window. For this region, the ED-VE-ADD variation does better than our approach because it correctly classifies the known window as well as the non-anomalies, and does not misclassify any.

Thus, we see that our algorithm consistently does equal or better than others in most datasets. DBSCAN does not perform as well as VE/VM/VEM-ADD but incorporating the λ computation makes it a better anomaly detection algorithm. Even if the accuracy of circular scan statistic based algorithm is better in some places, it is either because of the difference in selection of test statistic or because the non-anomalies were correctly classified. However, overall, our algorithm does a better job at discovering the known anomaly.

4.2.3 Time

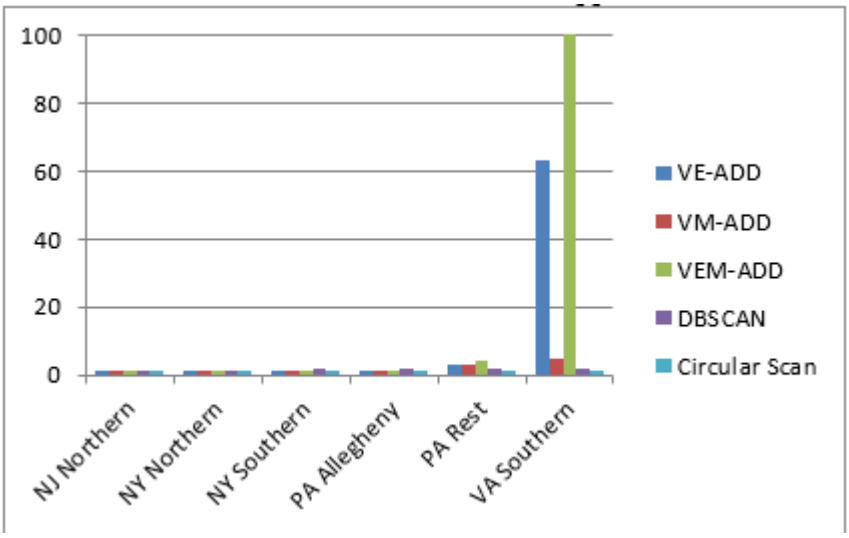
We present a comparison of the time taken for each approach to complete in figure 13. Almost all the algorithms finished within a second. Some of them took about 2 minutes to complete. ED-VE-ADD for PA Rest and VA Southern took the most time, about 8 hours and 4 hours respectively to run. This is because they have the greatest number of input nodes amongst other regions and moreover since we did not know the ε at which it found λ_{max} beforehand (for Euclidian distance), we varied the ε from a very low value to very high value.

Figure 13. *Time for execution*



The graphs in figure 14 for time have been plotted not considering the run times for the ED-VE-ADD approach. The graphs show that our algorithms take a little longer to run as compared to the DBSCAN and the circular scan statistic approaches.

Figure 14. *Graphs for time not considering ED-VE-ADD run time*



5. CONCLUSION AND FUTURE WORK

In this paper, we proposed an approach to bridge the gap between scan statistics based methods and clustering techniques for the discovery of spatial anomalous windows. Our method utilized the well-studied DBSCAN approach to identify spatial anomalous windows.

Our results indicated that we effectively bridged the gap between scan statistics based approach and spatial clustering method. We presented several variations on the algorithms to adapt the DBSCAN parameter settings so that we do not have to provide a single value for the input parameters. We presented detailed experimental results in real world crime datasets for each of the variations of algorithms as well as for circular scan statistic based method (Kulldorff 1997and DBSCAN (Ester et al. 1998to illustrate the efficacy of our approach. In addition, we validated our ðndings with certain known crime trends in these real world datasets, indicating the efficacy of our approach. Our results indicated that we generally attain equal or high accuracy as compared to the scan statistics based approaches and higher accuracy than pure clustering approach, in real world crime data.

One major limitation is that we presented our algorithms for univariate data that find arbitrary shaped anomalous windows, however, we need to expand this to multivariate data. Additionally, we consider a univariate test statistic; however, in the future we would like to compare results of multivariate test statistics. In this paper, we looked at only one test statistic to compare anomalous behavior. It would be interesting to see the comparative outcomes of various test statistics in the discovery of anomalous windows. Lastly, we would like to study the results of this approach applied to other real-world datasets beyond the crime domain.

FUNDING AGENCY

The publisher has waived the Open Access Processing charge for this article.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Record*, *22*(2), 207–216. doi:10.1145/170036.170072

Alimohammadi, H., & Chen, S. N. (2022). Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis. *Expert Systems with Applications*, *191*, 116371. doi:10.1016/j.eswa.2021.116371

Amarbayasgalan, T., Jargalsaikhan, B., & Ryu, K. H. (2018). Unsupervised novelty detection using deep autoencoders with density based clustering. *Applied Sciences (Basel, Switzerland)*, *8*(9), 1468. doi:10.3390/app8091468

Crime Rankings. (n.d.). *America's Safest Places*. Available from: http://www.morganquitno.com/rank.htm

Das, K., & Schneider, J. (2007). Detecting anomalous records in categorical datasets. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 220-229.

Emadi, H. S., & Mazinani, S. M. (2018). A novel anomaly detection algorithm using DBSCAN and SVM in wireless sensor networks. *Wireless Personal Communications*, *98*(2), 2025–2035.

Ester, M. (1998). Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Mining and Knowledge Discovery*, *2*(2), 169–194.

FIPS Codes. (n.d.). *State and County Codes and Names*. Available from: https://www.census.gov/geo/tigerline/append_a.pdf

Han, J. (2005). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc.

Harel, D., & Koren, Y. (2001). Clustering spatial data using random walks. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 281-286.

Janeja, V. P., & Atluri, V. (2008). Random Walks to Identify Anomalous Free-Form Spatial Scan Windows. *IEEE Transactions on Knowledge and Data Engineering*, *20*(10), 1378–1392.

Janeja, V. P., & Palanisamy, R. (2012). *Multi Domain Anomaly Detection in Spatial Datasets. Knowledge and Information Systems- An International Journal*.

Kulldorff, M. (1997). A spatial scan statistic. *Communications in Statistics. Theory and Methods*, *26*(6), 1481–1496.

Kulldorff, M. (1999). Spatial Scan Statistics: Models,Calculations, and Applications. In J. Glaz & N. Balakrishnan (Eds.), *Scan Statistics and Applications* (pp. 303–322). Birkhäuser Boston.

Kulldorff, M. (2007). *Multivariate scan statistics for disease surveillance.* Wiley.

Kulldorff, M. (2010). *SaTScan User Guide for version 9.0*. Academic Press.

Morgan Quitno Home Page. (n.d.). Available from: http://www.morganquitno.com/index.htm

Morgan Quitno Press Acquired by CQ Press. (n.d.). Available from: http://www.morganquitno.com/presscq.htm

Mungekar, D., Joshi, H., Kankekar, A., Nair, P., & Das, P. (2021). Crime Analysis using DBSCAN Algorithm. *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, 628-635. doi: doi:10.1109/ICIRCA51532.2021.9544719

Naus, J. I. (1965). The Distribution of the Size of the Maximum Cluster of Points on a Line. *Journal of the American Statistical Association*, *60*(310), 532–538.

Redmond, M. (2011). *Communities and Crime Unnormalized Data Set*. UCI Machine Learning Repository. Available from: http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized

Shaojun, L., Xing, Z., Lipeng, X., Jijie, S., & Dongsheng, L. (2022). DBSCAN Based Parameter Optimization of KPCA for Fault Diagnosis. In L. Yan, H. Duan, & X. Yu (Eds.), Advances in Guidance, Navigation and Control. Lecture Notes in Electrical Engineering (Vol. 644). Springer. https://doi.org/10.1007/978-981-15-8155-7_355.

Sheridan, K., Puranik, T. G., Mangortey, E., Pinon-Fischer, O. J., Kirby, M., & Mavris, D. N. (2020). An application of dbscan clustering for flight anomaly detection during the approach phase. In *AIAA Scitech 2020 Forum* (p. 1851). Academic Press.

Shi, L., & Janeja, V. P. (2009). Anomalous window discovery through scan statistics for linear intersecting paths (SSLIP). *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 767-776.

Tango, T. (2021). Spatial scan statistics can be dangerous. *Statistical Methods in Medical Research*, *30*(1), 75–86.

US Census Bureau. (n.d.). *Census Regions and Divisions of the United States*. US Census Bureau. Available from: https://www.census.gov/geo/www/us_regdiv.pdf

Xie, Y., & Shekhar, S. (2019, August). Significant DBSCAN towards statistically robust clustering. In *Proceedings of the 16th International Symposium on Spatial and Temporal Databases* (pp. 31-40). Academic Press.

Youngser, P. (2010). *Anomaly Detection using Scan Statistics on Time Series Hypergraphs*. Academic Press.

Zhang, L., Lin, J., & Karim, R. (2018). Adaptive kernel density-based anomaly detection for nonlinear systems. *Knowledge-Based Systems*, *139*, 50–63.

APPENDIX A - ALGORITHMS

The getCombinations method is called for each scan window in Core Node and Neighbors Set. We outline the algorithm for getCombinations from lines 7 to 12 in algorithm 4 below. In line 11, we keep the core node in each scan window as 'prefix' and take all the possible combinations of the neighbors. As discussed above, we add this prefix (core node) to the combinations of neighbors to ensure contiguous spatial nodes. Line 16 eliminates duplicates before finally adding the newly made scan windows to the Combination Set. 'allowedWindowSize' on lines 3 and 17 is the maximum size of scan window that is allowed. Here, we set it to half the size of the nodes. The circular scan statistic also uses a similar technique to limit cluster size. Unlike circular scan statistic though, there is no penalty on large scan windows when forming the Core Node and Neighbors Set. Any number of points that fall within an ε neighborhood for a core point form a scan window
In this algorithm 'i' is the scan window which is compared with all the 'j' subsequent scan windows. Unions are formed and checks applied from lines 17 to 21. As in combinations, we use 'allowedWindowSize' to limit the size of the scan window.

Figure 15. Algorithm 1: Anomaly detection with DBSCAN (ADD

```
Algorithm 1: Anomaly Detection with DBSCAN (ADD)

Inputs: (ε, Min-nodes, SetofNodes)
Each line in input file has
   (x, y) location of node,
   attr_of_interest,
   base_population

FOR i IN (0 TO SetofNodes.SIZE)
   Compute λmax;
END FOR

getCoreNodeandNeighborsSet (ε, Min-nodes);//uses DBSCAN
getCombinationSet (ε, Min-nodes);
getUnionSets (ε, Min-nodes);

Compute λmax across CoreNodeandNeighborsSet, CombinationSet and Union Sets
and report associated scan window
```

Figure 16. Algorithm 2: Generate combinations

```
Algorithm 2: Generate Combinations

getCombinationSet (ε,Min-nodes)

allowedWindowSize = SetofNodes.SIZE/2          //limit the size of scan window

FOR i IN (0 TO CoreNodeandNeighborsSet. SIZE)
   getCombinations (i); //these would be entries in Combination Set
END FOR
-----------------------------------------------------------------------------
getCombinations (i) // i is scan window

RETURN IF only one node IN scan window i //already accounted for
nodeList = all nodes IN scan window i
j = 0
prefix = nodeList [j]  //core node
combination (prefix, nodeList, j+1) // getCombinations
-----------------------------------------------------------------------------
combination (prefix, nodeList, i)

IF (nodeList.SIZE > i)
   newScanWindow.ADD(prefix + nodeList[i])
   IF newScanWindow ! IN combinationSet   //eliminate duplicates
      AND newScanWindow. SIZE < allowedWindowSize THEN
      ADD newScanWindow to the combinationSet
   END IF

   combination(prefix, nodeList, i+1)
   prefix.ADD(nodeList[i])
   combination(prefix, nodeList, i+1)

END IF //combination
```

Figure 17. Algorithm 3: Form unions

```
Algorithm 3: Form Unions

getUnionSets (ε, Min-nodes)

allowedWindowSize = SetofNodes.SIZE/2          //limit the size of scan window
currentSet = 1
// starts with CombinationSet number, goes on till the stopping condition is reached

WHILE(currentSet.SIZE > 0)
// Stop if No entries in current Set

   formUnion (currentSet);
   //this method will create scan windows for New Union Set

   currentSet++;
END WHILE
-----------------------------------------------------------------------------------------------

formUnion(A)
//A is givenWindowSet using which unions are derived

FOR each scan window i IN A //(givenWindowSet)
  FOR all scan windows j from i+1 onwards
  //compare with subsequent scan windows only
     newScanWindow = UNION (all nodes IN scan window i +
                            all nodes IN scan window j)
     IF newScanWindow !IN all previous Sets  // Check 1
        AND newScanWindow !IN currentSet  //Check 2
        AND newScanWindow. SIZE < allowedWindowSize THEN
           ADD newScanWindow to currentSet+1
     END IF
  END FOR
END FOR //formUnion
```

Figure 18. Algorithm 4: VE-ADD

```
Algorithm 4: VE-ADD

Inputs: (ε_min, ε_max, ε_step, Min-nodes, SetofNodes)
// Accept min, max and step in which ε is to be varied

Each line in input file has
   (x, y) location of node,
   attr_of_interest,
   base_population

FOR i IN (0 TO SetofNodes.SIZE)
   Compute λ_max;
END FOR

ε = ε_min;
WHILE (ε <= ε_max)
   getCoreNodeandNeighborsSet (ε, Min-nodes);
   getCombinationSet (ε, Min-nodes);
   getUnionSets (ε, Min-nodes);
   ε = ε + ε_step;
END WHILE
```

Figure 19. Algorithm 5: VM-ADD

```
Algorithm 5: VM-ADD

Inputs: (ε, Min-nodes_min, Min-nodes_max, Min-nodes_step, SetofNodes)
// Accept min, max and step in which Min-nodes is increased

Each line in input file has
   (x, y) location of node,
   attr_of_interest,
   base_population

FOR i IN (0 TO SetofNodes.SIZE)
   Compute λ_max;
END FOR

Min-nodes = Min-nodes_min;
WHILE (Min-nodes <= Min-nodes_max)
   getCoreNodeandNeighborsSet (ε, Min-nodes);
   getCombinationSet (ε, Min-nodes);
   getUnionSets (ε, Min-nodes);
   Min-nodes = Min-nodes + Min-nodes_step;
END WHILE
```

Figure 20. Algorithm 6: VEM-ADD

```
Algorithm 6: VEM-ADD

Inputs: (ε_min, ε_max, ε_step, Min-nodes_min, Min-nodes_max, Min-nodes_step, SetofNodes)
// Accept min, max and step of ε and min-nodes both

Each line in input file has
   (x, y) location of node,
   attr_of_interest,
   base_population

FOR i IN (0 to SetofNodes.SIZE)
   Compute λ_max;
END FOR

ε = ε_min;
Min-nodes = Min-nodes_min;

WHILE (ε <= ε_max)
   WHILE (Min-nodes<= Min-nodes_max)
      getCoreNodeandNeighborsSet (ε, Min-nodes);
      getCombinationSet (ε, Min-nodes);
      getUnionSets (ε, Min-nodes);
      Min-nodes = Min-nodes + Min-nodes_step;
   END WHILE
   ε = ε + ε_step;
   Min-nodes = Min-nodes_min; // Reset, value increased in earlier WHILE LOOP
END WHILE
```

Vandana Janeja is Professor and Chair of the Information Systems department at the University of Maryland Baltimore County (UMBC). She heads the MData lab at UMBC. She is member of the UMBC ADVANCE Executive committee focusing on diversity in STEM and is a member of the ADVANCE leadership cohort (2020-2021). She is a UMBC innovation fellow (2020-2022) advancing the ideas of including ethics in data science. She served as an expert at NSF supporting data science activities in the CISE directorate (2018-9/2021). Her research is in the area of data science with a focus on data heterogeneity across multiple domain datasets. Her research has been funded through federal, state and private organizations including NSF, U.S. Army Corps of Engineers, MD State Highway Administration, CISCO. She holds a Ph.D. in Information Technology from Rutgers University. She completed her MBA from Rutgers University and MS in Computer Science from New Jersey Institute of Technology.

Prerna Mohod joined the University of Maryland, Medical Center (UMMC) as an intern, later joining as a full-time employee working as a Systems Analyst. She worked for the Radiology IT department. She studied the radiology workflow and was part of a project team to validate a workflow lexicon on behalf of the University of Maryland School of Medicine (UMSOM) and the Society for Imaging Informatics in Medicine (SIIM). The goal was to measure and improve radiology workflow, patient safety and clinical quality across the U.S. by defining a standard, measurable data model for clinical workflow. Based on this work, she co-authored the paper 'Workflow Lexicons in Healthcare: Validation of the SWIM Lexicon' published in the Journal of Digital Imaging in June 2017. Apart from report-generating work for operational and analytical purposes for the leadership, she assisted physicians (radiologists) with their research by retrieving, cleaning, and collating data as per their requirements. These data sets were used for research and clinical analyses by the radiologists. Prerna has been a speaker at the ---- SIIM conference (Society for Imaging Informatics in Medicine) held every year at different locations and conducted a data mining learning lab there. Prerna wrote this paper as part of her thesis work at UMBC.