Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 1218–1227

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

Clustering West Nile Virus Spatio-temporal data using ST-DBSCAN

Chimwayi. K. B[a], J Anuradha[b,*]

[a]*M-Tech Student, SCOPE, VIT, Vellore,Tamil Nadu,632014,India.*
[b]*Associate Professor, SCOPE, VIT, Vellore,Tamil Nadu,632014,India.*

Abstract

Spatio-temporal data mining has been the talk of the day due to high availability of spatio-temporal data from varied sources in diverse fields. Through many tracking devices, huge amounts of spatio-temporal data are being generated. In epidemiology, diseases, patterns and trends attached can be explored taking advantage of methods such as spatio-temporal clustering to discover new knowledge. In this paper Spatio-Temporal Density Based Spatial Clustering of Applications with Noise (ST-DBSCAN) is implemented and analysed on a public health dataset. Upon the implementation, results are analysed, loopholes spotted and a fuzzy version of ST-DBSCAN is proposed. The method is successfully applied to find spatio-temporal clusters in Chicago West Nile Virus (WNV) surveillance data for the period 2007 to 2017.The drawbacks in the original ST-DBSCAN are identified and solutions are proposed. ST-DBSCAN is an extension of the original Density Based Spatial Clustering of Applications with Noise (DBSCAN).

Keywords: Clustering; Spatial Data Mining; DBSCAN; ST-DBSCAN; ST-Fuzzy DBSCAN

1. Introduction

Knowledge discovery from spatio-temporal data has been gaining momentum is this era regardless of the complexity associated with dealing with the kind of data whereby temporal and spatial neighbours are to be considered to form meaningful clusters. Clustering is referred to the process of grouping data according to similarities. In data analysis, clustering is an important problem and many data scientists are using clustering to aid in identifying patterns for example to identify genes with similar expression patterns among many other diverse applications. Many clustering algorithms exist and they are usually grouped as partitioning, hierarchical and density based.

The major problem that has been noted for example with the k-means algorithm is the fact that it ignores the notion of outliers in datasets at hand. This in turn leads to the formation of clusters whereby all points have to belong to at least one cluster forgetting that some real world datasets for example in the field of anomaly detection have outliers and need not to be in a cluster as the points but may be noise points for certain anomalies. Density based clustering is a

* Corresponding author. Tel.: +919443130861
E-mail address: januradha@vit.ac.in

10.1016/j.procs.2018.05.037

form of finding clusters in data using the notion of identifying dense clusters of points which are separated from sparse clusters. The identification of dense clusters of points help to identify outliers in the data and to find also clusters of arbitrary shape.

Since the inception of DBSCAN [4], a lot of extensions to the algorithm have been proposed. Many extensions to the algorithm have been proposed, among them there is ST-DBSCAN and fuzzy extensions of DBSCAN. Density based clustering is usually used to cluster spatial attributes so as to find certain concentrations for objects in space, and if it is ST-DBSCAN finding clusters in space or in both space and time is possible. These will be objects that have close spatial locations, an example explaining the importance of the concept is being able to find concentrations of objects that have close spatial locations and their existence times as well.

The use of density based clustering in this work demands a proper understanding and understandable definition of distance and in this case spatial distance as well as spatio-temporal distance. The algorithm helps in finding spatial, spatio-temporal concentrations of objects and groups of points with similar spatio-temporal properties.

S. Kisilevich et al. [10] made a review on spatio-temporal clustering specifically on trajectory clustering. Spatio-temporal clustering is a relatively growing field in the domain of data mining and machine learning. It is defined as a process where objects are grouped according to temporal and spatial similarity. It has been noted that surveillance of mosquitos provide an essential tool in public health for evaluating risk of viruses as West Nile Virus and for efficient allocation of scarce public health resources. On the other hand proper evaluation and analysis enables proper justification of emergency control actions.

For density based clustering methods, if a point is visited, the density is considered by counting the number of points within the epsilon specified. The object that have points more than the specified minimum points threshold form a cluster. Spatial Data Mining (SDM) which is the extraction of hidden information and patterns from spatial data can be broadly classified into supervised and unsupervised learning. This work greatly focuses on unsupervised classification well known as clustering. In this work the major aim is to cluster spatio-temporal data which has emerged in huge quantities in this case being surveillance spatio-temporal data. The rise of huge spatio-temporal data due to ubiquitous devices has helped researchers identify the greater need for new and improved spatial and spatio-temporal data mining algorithms for extracting useful interesting patterns from the data.

The core of this work is on using SDM for the enhancement of public health, considering that given any public health information for example surveillance data, the algorithm should be applied to find significant patterns necessary for enhancing decision making in the sector. The rest of this paper is organized as follows; the second section provides literature review preceded by application, then results and discussions which are then followed by conclusions and future work.

2. Literature survey

Considering that in real world applications we encounter different types of spatio-temporal data, Kisilevich et.al [10] provided a guide and framework for categorising this data. The categories of spatio-temporal data can be geo-referenced time series, geo-referenced variables, moving objects and spatio-temporal events. The type of data is clustered to find events that are intertwined or close together in space and time. Scan statistics is one method besides density based methods which can be used for clustering this kind of data. In [10] the main focus was to cluster trajectories at the same time reviewing approaches and methods of spatio-temporal clustering.

In [8] fuzzy extensions of the DBSCAN have been proposed and implemented for the generation of clusters which have distinct fuzzy characteristics. The work resulted in the relaxation of parameters which are generally considered for DBSCAN and therefore provides a ground to work on the proposed algorithm which is ST-FDBSCAN which mainly clusters spatio-temporal data to find fuzzy density characteristics in clusters.

Derya Birant et al. [2] proposed ST-DBSCAN as an extension of DBSCAN to cater for some downfalls encountered when using DBSCAN algorithm, the major being the ability to cluster according to spatial and non-spatial attributes (temporal) of a dataset.The proposed algorithm caters for changes in relation to identifying core points, noise points and clusters which are adjacent to each other. Authors in [2], made significant changes to the DBSCAN algorithm so as to cater for clustering spatio-temporal data according to the available spatial, non-spatial and temporal components available in a dataset. The second improvement laid down is to enable the clustering of points which do have different densities, which DBSCAN struggle on as it fails to detect some noise points. In the original paper, each cluster is

assigned a density factor so as to deal with the problem of not detecting noise points when different densities exist. In some instances the values of border points in a cluster may be entirely different with the values of the other border points in the cluster in an opposite side. Authors proposed the use of comparing the average value of a cluster with a new incoming value. A spatial data warehouse has been designed and the algorithm was implemented on different environmental spatio-temporal datasets.

Researchers in [3] worked on a survey paper where they gave an overview of work done in relation to spatio-temporal clustering. The work highlighted a significant work concerning spatio-temporal clustering. The algorithms highlighted and reviewed include ST-GRID, ST-DBSCAN and Fuzzy C-Means (FCM) among others. It has been noted that the main objectives when clustering spatio-temporal points is to find clusters which have a high density and these are generally termed hot spots. In research hot spots can be used for identifying social movements and outbreaks of diseases which is identified by much density of events both in space and time. Spatial statistics has been used to find hot-spots with techniques such as Spatial Scan Statistic (SaTScan).

In their paper [13], Moulavi et al. proposed Density Based Clustering Validation (DBCV). It is greatly noted in this paper that the most challenging aspect of density based clustering is validation. Authors therefore proposed a relative validation index measure called the DBCV. The proposed algorithm considers the concept of Hartigans model of density-contour trees [13] to compute the least dense region inside a particular cluster as well as the most dense region between the clusters which are then used to measure the within and between cluster density connectedness of clusters. The measure implemented for DBSCAN was compared with other cluster validation techniques.

An augmented fuzzy c-means for clustering spatio-temporal data has been proposed in [9] . An augmented distance function is used for optimizing performance, reconstruction and prediction error are used. Synthetic and real world datasets are used. Scan statistic has also been generalised to cater for spatio-temporal data. With scan statistic researchers have embarked in studies which include disease outbreak analysis, identification of events in twitter data and identifying crime hot spots.

In [7] clustering is performed on data for hotspots in peat land using DBSCAN algorithm. A web based clustering application is built in R for grouping the hotspots from the data using shiny framework. Researchers in [14] embarked on spatio-temporal clustering on road accident rather than diseases. This method provides a base and a working frame for this study since it has a focus on spatio-temporal analysis for WNV data. An assessment of spatial clustering of accidents and hotspots spatial densities was achieved by following Morans I method of spatial autocorrelation, point kernel densities and Getis-Ord Gi* statistics. Accidents were therefore compared in terms of spatial and temporal aspects in the particular city. Researchers in [12] had an aim of integrating WNV datasets so as to have a greater insight of the spatial distribution of the virus in Israel. However this study mainly used mapping whereby chrolopleth maps were used to show morbidity of humans with respect to WNV. Results showed a high prevalence to the disease as well as high risk areas.

Authors in [11] provided an important framework on challenges and research paths to be considered in the field of Spatial Data Mining with respect to Geographic Information System public health surveillance systems. Allen et.al [1] carried a study on monitoring influenza outbreaks using the social media platform Twitter. They focused on applying GIS and machine learning to classify tweets as a way of surveillance for influenza. Support Vector Machine (SVM) classifier has been trained by the use of manually tagged data (tweets) which were collected from a flu season. The spatial attributes of tweets were taken into consideration.

In [5, 6] authors provided an implementation of Compose Density between and within clusters (CDbw) which is a cluster validity index. This is important to be reviewed in literature for this algorithm since in the research field it has been used mostly for validating the original DBSCAN algorithm implementations. The measure assesses compactness, cohesion and separation of clustering algorithms.

3. Application

The aim of this work is to implement ST-DBSCAN algorithm on WNV dataset to find spatio-temporal patterns and to identify loopholes which provides a basis for proposing a fuzzy version of the algorithm. The objective has been attained using R for the implementation after various procedures have been done on the dataset. The application is therefore explored in the sections below.

3.1. Description of the dataset

The dataset under consideration which has been used as a representation of public health data is West Nile Virus traps surveillance data for the City of Chicago. Spatio-temporal clustering hence identifies clusters by grouping data objects according to spatial and temporal similarity. The data contains locations (spatial attribute for traps), temporal representing date of testing and other non-spatial attributes. The dataset has 13 features and 25295 instances for a 10 year period starting 2007 to 2017. The spatial temporal dataset represents a list of locations and test results for pools of mosquitoes which have been tested via the Chicago Department of Public Health (CDPH) Environmental Health program. The sample snapshot for the dataset has been provided in Fig. 1.

```
Console ~/
> chicago=read.csv("West_Nile_Virus__WNV__Mosquito_Test_Results (6).csv")
> head(chicago)
  SEASON.YEAR WEEK TEST.ID                BLOCK TRAP TRAP_TYPE             TEST.DATE NUMBER.OF.MOSQUITOES
1        2011   29   31550 100XX W OHARE AIRPORT T916    GRAVID 07/25/2011 12:07:00 AM                  3
2        2016   25   42613     58XX N PULASKI RD T027    GRAVID 06/22/2016 12:06:00 AM                  6
3        2007   27   20583    15XX W WEBSTER AVE T045    GRAVID       07-11-2007 03:07                  4
4        2009   38   28275     5XX S CENTRAL AVE T031    GRAVID 09/25/2009 12:09:00 AM                  4
5        2011   34   32254     58XX N PULASKI RD T027    GRAVID 08/26/2011 12:08:00 AM                  2
6        2012   30   34127       24XX E 105TH ST T128    GRAVID 07/27/2012 12:07:00 AM                 50
    RESULT            SPECIES LATITUDE LONGITUDE                                  LOCATION
1 negative          CULEX PIPIENS       NA        NA
2 negative         CULEX RESTUANS 41.98632 -87.72838 (41.986319851449004, -87.72837845617912)
3 negative CULEX PIPIENS/RESTUANS 41.92170 -87.66696  (41.92170457422864, -87.66696323469388)
4 negative          CULEX PIPIENS 41.87287 -87.76474   (41.87287286249572, -87.7647365320396)
5 negative         CULEX RESTUANS 41.98632 -87.72838 (41.986319851449004, -87.72837845617912)
6 positive CULEX PIPIENS/RESTUANS 41.70469 -87.56424  (41.704687213624375, -87.5642355621286)
>
```

Fig. 1 Snap shot of WNV dataset.

3.2. Data pre-processing

For effective spatio-temporal clustering proper data pre-processing was essential. The dataset had missing coordinates whereby the automatic geocoder did not give spatial locations for some traps. Using geocoding in R, the missing coordinates were dealt with for the 8 trap locations that had missing values. Three attributes of the dataset had missing values before geocoding. As part of pre-processing, dealing with dates and time in the dataset is an important procedure. The dates and time from the raw dataset had different formats such that it was difficult to deal with the time classes in R. Pre-processing was done to ensure proper representation of time using R based time object.

3.3. Implementation and validation of ST-DBSCAN

According to [4], it has been noted that the benefits of using DBSCAN include the ability to find arbitrary shaped clusters, unlike partitioning clustering methods there is no need to specify the number of clusters in advance and at the same time the method works quite well for clustering large datasets. Using K-Nearest Neighbor (KNN) distance plot the epsilon 1 for spatial parameter was determined. The epsilon 2 for temporal attribute was set considering the time in the actual dataset considering that the trapped mosquitoes were tested monthly or weekly from the month of May till September each year. Validation of ST-DBSCAN remains a challenging task unlike validating other methods of clustering which are based on centroids. To use external validation for the dataset proved as challenging provided that there was no predefined expected number of clusters nor was the expected number of clusters known since the data has never been used for clustering. ST-DBSCAN according to the authors requires epsilon 1 to represent spatial attributes, epsilon 2 to represent temporal attributes.

Table 1 shows the spatio-temporal clusters formed by the algorithm on WNV data considering the Epsilon1 (spatial), Epsilon 2 (temporal) and Minimum Points. The spatial threshold has been set to 0.02 whereas the temporal threshold has been set to 7776000 seconds which equate to 90 days using the knowledge of the dataset and application field. Spatial distances are less since the traps are all located in different blocks which are in one city, which is in Chicago. As shown in Table 1, the algorithm has been implemented with different parameters of Eps and minimun points. The best parameter chosen for cluster formation is 10 as the minimum points. The data points for each dataset are tabulated with the clusters formed as well as the number of positive and negative cases in the clusters.

According to the clusters formed, the years that have clusters which have a total of more than 100 cases of mosquitos which tested positive are 2007, 2012, 2013, 2014, 2016 and 2017, with 2012 being highest. For each

Table 1: Clusters formed for the 10 year period using ST-DBSCAN with 2 different parameters for each year.

Year	Data Points	Eps1-Spatial	Eps2-Temporal	Minimum Points	Clusters	Noise Points	Positive Cases	Negative Cases
2007	1690	0.02	7776000	15	24	274	141	1549
2007	1690	0.02	7776000	10	39	94	141	1549
2008	1182	0.02	7776000	15	29	300	41	1141
2008	1182	0.02	7776000	10	49	51	41	1141
2009	1516	0.02	7776000	15	41	191	14	1498
2009	1516	0.02	7776000	10	53	54	14	1498
2010	1796	0.02	7776000	15	52	166	52	1744
2010	1796	0.02	7776000	10	64	15	52	1744
2011	1328	0.02	7776000	15	39	272	28	1300
2011	1328	0.02	7776000	10	56	68	28	1300
2012	1689	0.02	7776000	15	49	166	281	1408
2012	1689	0.02	7776000	10	61	17	281	1408
2013	1365	0.02	7776000	15	46	210	121	1244
2013	1365	0.02	7776000	10	62	9	121	1244
2014	1622	0.02	7776000	15	88	24	150	1472
2014	1622	0.02	7776000	10	58	24	150	1472
2015	1068	0.02	7776000	15	26	321	74	994
2015	1068	0.02	7776000	10	42	127	74	994
2016	1125	0.02	7776000	15	29	297	221	904
2016	**1125**	**0.02**	**7776000**	**10**	**45**	**114**	**221**	**904**
2017	966	0.02	7776000	15	20	385	111	855
2017	**966**	**0.02**	**7776000**	**10**	**41**	**129**	**111**	**855**

year, 2 different minimum points have been used. The highlighted years show the clusters formed from 10 minimum points, the actual one used for clustering.

```
Console ~/
> wnvpos=read.csv("2017.csv")
> wnvpos$date=parse_date_time(wnvpos$date, orders="mdy HMS", tz="GMT")
> clusters=stdbscan(data = wnvpos,0.02,7776000,minpts = 10,countmode = 1:nrow(wnvpos))
> print.dbscan(clusters)
dbscan Pts=966 MinPts= eps=0.02
         0  1  2  3   4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
border 129  0  0  0   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
seed     0 13 21 11 178 21 21 48 19 17 25 14 13 41 14 21 17 16 23 15 14 13 12 10 14 16 13 14 14 19
total  129 13 21 11 178 21 21 48 19 17 25 14 13 41 14 21 17 16 23 15 14 13 12 10 14 16 13 14 14 19
       30 31 32 33 34 35 36 37 38 39 40 41
border  0  0  0  0  0  0  0  0  0  0  0  0
seed   16 10 11 15 17 15 11 12 13 10 10 10
total  16 10 11 15 17 15 11 12 13 10 10 10
```

Fig. 2 Clusters from 2017 WNV data.

```
> wnvpos=read.csv("2016.csv")
> wnvpos$date=parse_date_time(wnvpos$date, orders="mdy HMS", tz="GMT")
> clusters=stdbscan(data = wnvpos,0.02,7776000,minpts = 10,countmode = 1:nrow(wnvpos))
> print.dbscan(clusters)
dbscan Pts=1125 MinPts= eps=0.02
         0  1  2  3  4   5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
border 114  0  0  0  0   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
seed     0 32 25 23 54 158 49 17 18 16 12 31 20 24 14 20 12 13 15 19 16 25 15 30 20 21 37 23 17 18
total  114 32 25 23 54 158 49 17 18 16 12 31 20 24 14 20 12 13 15 19 16 25 15 30 20 21 37 23 17 18
       30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
border  0  0  0  0  5  0  0  0  0  4  1  4  2  3  3  3
seed   14 20 11 10  5 15 21 29 11  7 11  6 11  7  7  7
total  14 20 11 10 10 15 21 29 11 11 12 10 13 10 10 10
```

Fig. 3 Clusters from 2016 WNV data.

3.4. Visualizations

Visualisation provides a way of determining the meaning and usefulness of the cluster solutions. Figure 2 and Figure 3 show 41 and 45 clusters respectively for 2017 and 2016 WNV datasets.

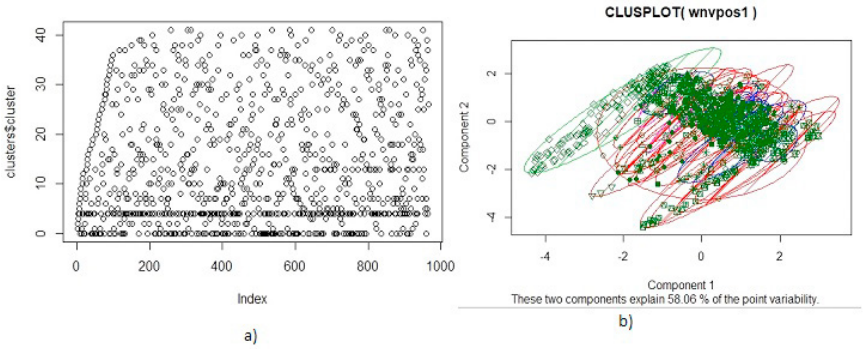

Fig. 4: (a) 2017 Clusters (b) Clusters using clusplot for 2017.

3.5. Interpretation of clusters

The obtained cluster solutions are interpreted to determine what they have in common at the same time observing how objects in one cluster differ from the ones in another. To analyse the results randomly, in 2017 data, cluster 1 has no positive cases and has all elements in the cluster belonging to spatial point (-87.70024; 41.93279) at a time stamp 2017-06-16 17:36:00. Cluster 4 also has about 57 points which are positive cases and the spatio-temporal combination of the points is point (-87.90732; 41.97416) and timestamp 2017-06-19 17:36:00. For all the clusters formed, the bottom line is to ascertain that at a certain trap, at a particular time of testing, what quantity, species have positive or negative tests for the WNV. Figure 4 a) gives a plot of the cluster which has been formed and their index. Of the clusters formed and represented in plot 4 b) 58 percent of the information about multivariate data in the clusters has been captured by the plot and hence the clusters are quite clear though they can be better.

3.6. Cluster validation

In validating clusters formed, there is a greater need to determine if the groupings formed are real otherwise the clusters might be just highlighting the unique aspects of the dataset or technique used. Cluster validation is a term used for the processes of designing a procedure to evaluate the goodness of clustering algorithm results. The method can help by avoiding the issue of finding patterns in random data.

Cluster validation can be internal, external or relative. Internal validation evaluates goodness by considering internal information of the clustering process rather than looking at external information. On the other angle, external way of validating clusters is a process of comparing clustering results with an externally known result. This means that the extent of cluster labels matching with supplied external labels is assessed. On another extent relative cluster validation assess clustering by means of varying different parameter values for one particular algorithm at hand. For example when dealing with DBSCAN it means different parameters for epsilon and minimum points are employed till an optimal number of clusters is identified.

Table. 2 shows how relative validation was and can be applied for the algorithm. Different values for the parameters have been considered. Some parameters such as a high value of minimum points reduces the number of clusters but increase noise points.

CDbw clustering index has been applied on ST-DBSCAN results. The values are highlighted in table 3. The higher the value of CDbw for a clustering algorithm, the better. The index for the algorithm is low, however in this case the dataset used is one hence there is no comparison with an entirely different dataset in this case.

From tabulated results of CDbw, there is a huge difference in the CDbw value as the number of points in a dataset increase as well as the number of clusters.

Table 2: Relative validation (varying parameters).

Eps1	Eps2	Minimum Points	Clusters	Noise	Total Points
0.02	7776000	15	59	245	1796
0.02	7776000	10	75	44	1796
0.02	7776000	5	80	11	1796
0.02	2160060	15	27	977	1796
0.02	2160060	10	60	380	1796
0.02	2160060	5	80	52	1796
0.02	2592000	15	39	632	1796
0.02	2592000	10	71	128	1796
0.02	2592000	5	79	24	1796
0.02	**6048000**	**15**	**59**	**245**	**1796**
0.02	6048000	10	75	44	1796
0.02	6048000	5	80	11	1796
0.01	7776000	15	58	211	1796
0.01	7776000	10	72	36	1796
0.01	7776000	5	77	3	1796
0.01	2160060	15	28	897	1796
0.01	2160060	10	59	332	1796
0.01	2160060	5	28	45	1796

Table 3: CDbw validation measure.

Dataset	Clusters	CDbw	Cohesion	Compactness	Seperation
WNV 2017	41	0.0005033044	0.01928815	0.01927963	1.353447
WNV 2016	45	8.402172	0.006292881	0.006290759	2.122458

4. Results and Discussions

The biggest challenge encountered on implementing this clustering algorithm is lack of documented evaluation measures and methods for ST-DBSCAN. There exist no particular evaluation measure (internal or external validation) for ST-DBSCAN. Applying external clustering validity for example in the dataset at hand proved challenging . For the implementation of ST-DBSCAN , authors designed their own data warehouse and mentioned the clusters formed, however it has proved challenging to access the data and implement the algorithm as a way of validating and determining correctness. Existing density based clustering validation techniques do not cater for the spatio-temporal aspect of the algorithm implemented. This method has not been so perfect for the clustering of WNV data since the clusters seem to be adjacent to each other. The density based algorithm works well if the clusters are distant from each other. In the pseudo code provided in the ST-DBSCAN paper, the major challenge is that the issue of density factor has been stressed as an improvement but is not accounted for in the pseudo code. This makes it difficult to evaluate and to compare. The issue of determining the distance for temporal attributes is not very clear. The algorithm implemented on WNV therefore stands as the base of the ST-Fuzzy DBSCAN (ST-FDBSCAN) implementation.

4.1. Proposed algorithm

An ST-FDBSCAN algorithm is proposed in this section for the generation of clusters with fuzzy cores and neighbourhoods. Crisp ST-DBSCAN algorithm usually fail to put into groups, fuzzy nature of cluster borders which overlap and are of varying size. The proposed algorithm aims to generate spatio-temporal clusters which have fuzzy overlapping borders and fuzzy cores. Similar to WNV dataset, most real world datasets have a common attribute of having faint borders and overlapping borders. This requires an efficient algorithm which has soft constraints to cater for such. Specifying approximate values for the ST-DBSCAN algorithm can be useful in some applications whereby it

may be difficult to set clear boundaries when forming clusters. The parameters used to cater for soft approximations are epsilon maximum, epsilon minimum, minimum points minimum and minimum points maximum for spatial and temporal attributes respectively. Equation (1) determines the density (dens) of clusters.

$$dens(d) = \sum_{(d_i \in neigh(d, \in_{max}))} \mu_{dist}(d, d_i) \tag{1}$$

If μ_{minD} (dens (d)) > 0 then the point d belongs to the fuzzycore of a certain cluster with a membership degree Fuzzycore(d) = μ_{minD} (dens (d)). If μ_{minD} (dens (d)) = 0 , then d is a border or a noise point.

$$\mu_{b(d)} = min_{di \in neighfcore(d)}(min(fuzzycore_c(d_i), \mu_{dist}(d, d_i))) \tag{2}$$

where $neighfcore(d) = d_i \; s.t. fuzzycore_c(d_i) > 0 \wedge \mu_{dist}(d, d_i) > 0$. The proposed ST-FDBSCAN procedure is given as Algorithm 1 and is outlined below.

Algorithm 1: ST-Fuzzy DBSCAN

```
ST-FDBSCAN (D, Eps1_max, Eps2_max, Eps1_min, Eps2_min, MinPts_max, MinPts_min)
input : D: dataset of spatio-temporal points
        Eps1_max, Eps2_max, Eps1_min,Eps2_min: soft constraints on the distance around a point
        leading to the definition of fuzzy neighbourhood size, here Eps1 is for
        spatial data and Eps2 is for non-spatial data (temporal/ other)
        MinPts_Min , MinPts_Max: constraint on the density around a point to be considered as fuzzy
        core point
C = φ
Clusters = φ
for ∀d ∈ D s.t.d is unvisited do
    mark d as visited
    nPts=regionQuery(d,Eps1_max,Eps2_max)
    dens(d)= equation(1)
    if μ minD(dens(d)) == 0 then
        mark d as NOISE
    else
        C=next cluster
        Clusters=Clusters ∪
         expandClusterFuzzy(d,npts,C,Eps1_max,Eps2_max,Eps1_min,Eps2_min,MinPts_max, MinPts_min)
    end
    return Clusters
end
expandClusterFuzzy(d,npts,C,Eps1_max,Eps2_max,Eps1_min,Eps2_min,MinPts_max, MinPts_min)
input : d: point which is marked as visited
        npts: points in fuzzy neighbourhood of d
        Eps1_max, Eps2_max, Eps1_min, Eps2_min: soft constraints on distance around spatial and
        non_spatial points so as to compute the approximate neighbourhood
        MinPts_min,MinPts_max: soft constraint on the density for consideration on the fuzzy core point.
        add d to C as core with membership μ minD(dens(d))
for ∀d′ ∈ npts do
    mark d′ as visited
    if μMinD(dens(d′)) > 0 then
        npts′ =regionQuery(d′,Eps1_max,Eps2_max)
        npts=npts ∩ npts′
        add d′ toC as core with membership μMinD(dens(d′))
    else
        add d′ to C(as border point with membership in equation(2)
    end
end
return C
```

5. Conclusions and future work

In this contribution, ST-DBSCAN has been implemented on a public health dataset with the aim of finding patterns and upon the implementation it has been noted that there is need for proper frameworks for validating ST-DBSCAN algorithm, at the same time a fuzzy version of the algorithm which can cater for spatio-temporal clustering is laid down. Computing the algorithm for large datasets proved to be a task that requires much computing power hence in the future many advanced techniques for addressing this are required so that it is possible to execute in parallel. As the ST-FDBSCAN is proposed, it is an important aspect to cater for the performance of the algorithm since when

parameters are added, computation time also increases. Upon the implementation of ST-DBSCAN on WNV data, it has been noted that there exist challenges in validating density based algorithms. As noted there exists no spatio-temporal datasets which are generic and already used for the algorithm for which authors know the number of clusters expected for external validation. Future research directions embark on a research journey which will work on the issue of determining validation measures specifically meant for ST-FDBSCAN which is being proposed. After the successful implementation of ST-DBSCAN to identify patterns in WVN data, it has been noted that there is need to cater for properly clustering uncertain datasets, which at the current time ST-DBSCAN is unable to do. Using CDbw validation measure, for the WNV dataset it is noted that the performance is quite lower. For lesser clusters on the dataset, the meausure is very low. The evaluation of ST-DBSCAN on WNV data therefore provided a framework for implementing a fuzzy version of the algorithm called ST-FDBSCAN with a proper validation technique meant for assessing clusters formed from spatio-temporal datasets.

References

[1] Allen, C., Tsou, M.H., Aslam, A., Nagel, A., Gawron, J.M., 2016. Applying gis and machine learning methods to twitter data for multiscale surveillance of influenza. PloS one 11, e0157734.
[2] Birant, D., Kut, A., 2007. St-dbscan: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering 60, 208–221.
[3] Dhundale, V.V., Takalikar, M., . Survey on spatio-temporal clustering .
[4] Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise., in: Kdd, pp. 226–231.
[5] Halkidi, M., Vazirgiannis, M., 2002. Clustering validity assessment using multi representatives, in: Proceedings of the Hellenic Conference on Artificial Intelligence, SETN, pp. 237–249.
[6] Halkidi, M., Vazirgiannis, M., 2008. A density-based cluster validity approach using multi-representatives. Pattern Recognition Letters 29, 773–786.
[7] Hermawati, R., Sitanggang, I.S., 2016. Web-based clustering application using shiny framework and dbscan algorithm for hotspots data in peatland in sumatra. Procedia Environmental Sciences 33, 317–323.
[8] Ienco, D., Bordogna, G., 2016. Fuzzy extensions of the dbscan clustering algorithm. Soft Computing , 1–12.
[9] Izakian, H., Pedrycz, W., Jamal, I., 2013. Clustering spatiotemporal data: An augmented fuzzy c-means. IEEE transactions on fuzzy systems 21, 855–868.
[10] Kisilevich, S., Mansmann, F., Nanni, M., Rinzivillo, S., 2009. Spatio-temporal clustering, in: Data mining and knowledge discovery handbook. Springer, pp. 855–874.
[11] Luan, H., Law, J., 2014. Web gis-based public health surveillance systems: a systematic review. ISPRS International Journal of Geo-Information 3, 481–506.
[12] Lustig, Y., Kaufman, Z., Mendelson, E., Orshan, L., Anis, E., Glazer, Y., Cohen, D., Shohat, T., Bassal, R., 2017. Spatial distribution of west nile virus in humans and mosquitoes in israel, 2000–2014. International Journal of Infectious Diseases 64, 20–26.
[13] Moulavi, D., Jaskowiak, P.A., Campello, R.J., Zimek, A., Sander, J., 2014. Density-based clustering validation, in: Proceedings of the 2014 SIAM International Conference on Data Mining, SIAM. pp. 839–847.
[14] Prasannakumar, V., Vijith, H., Charutha, R., Geetha, N., 2011. Spatio-temporal clustering of road accidents: Gis based analysis and assessment. Procedia-Social and Behavioral Sciences 21, 317–325.