IDENTIFYING CLUSTERS OF VIOLENCE AND SEXUAL OFFENCES USING DENSITY-BASED ALGORITHMS

Pratibha Patel
Department of Civil, Environmental and Geomatic Engineering, University College London, UK

Abstract
Crime prediction is an initial condition for its prevention, and it can be predicted along those dimensions in which it is concentrated (Farrell and Pease, 2014). Therefore, it is crucial to spot and study these crime clusters. In this paper, the identification of the crime clusters is carried out using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Spatial-Temporal Density-Based Spatial Clustering of Applications with Noise (ST-DBSCAN) clustering algorithms. The parameters of the algorithms were tuned, and the clusters formed were validated using silhouette score. Further, an attempt is made to explain why these crime clusters occur in a particular region using the Index of Multiple Deprivation (IMD).

Keywords: crime, hotspot, clustering, DBSCAN, ST-DBSCAN, silhouette score, IMD.

1. INTRODUCTION

Spatial clustering is used in diverse fields such as disease outbreaks, transportation, plant species, etc. However, it is increasingly becoming vital for crime analysis as the crime rate increases drastically. Hierarchical, partitioning, density, and grid-based clustering are the four types of clustering (Peter and Antonysamy, 2011).

1.1 Literature Review

Density-based approaches work by distinguishing between high-density and low-density areas. These methods can quickly discover clusters of arbitrary shapes (Smiti and Elouedi, 2012). DBSCAN is well-known for its ease of use and efficiency. However, apart from its benefits, such as detecting clusters of any shape or size in massive geographic databases, DBSCAN also has drawbacks, such as a long calculation time. A user's power to decide the values of input parameters is challenging. Finally, DBSCAN has a significant flaw: it cannot detect varying density clusters (Akbar and Khan, 2014).

ST-DBSCAN, a new density-based clustering algorithm based on DBSCAN. It proposes three marginal extensions to DBSCAN related to the identification of (i) core objects, (ii) noise objects, and (iii) adjacent clusters. In contrast to the existing density-based clustering algorithms, this algorithm can discover clusters according to the non-spatial, spatial, and temporal values of the objects (Nagwani and Bhansali, 2010).

1.2 Study Area

Birmingham is the second-largest city in the United Kingdom and a metropolitan borough in the West Midlands metropolitan county (Britannica,2022). It consists of ten districts. Birmingham was chosen for this study because it is the most dangerous major city in West Midland (CrimeRate, 2022).

1.3 Data Source, Description and Preparation

The crime datasets were acquired from an open provider, i.e. data.police.uk website. The data from West Midland Police was taken. The crime data contains details of crime occurrences, including approximate point locations. The police force ensures data quality by doing format validation, automated testing, and manual verification and approval (data.police.uk, 2022). However, there are some issues in areas such as location accuracy (~60 to 97%), double counting of crime data, missing outcomes, and data updating. However, in this study, the analysis is carried out at the district level, and therefore the available data is good enough to carry out the analysis.

As the dataset contained data for seven boroughs of the West Midlands, it was filtered out for Birmingham. The datasets were in the WGS84 datum, i.e., lat-long (degrees); it was converted to the British National Grid in Easting-Northing(metres).

This was done because the parameters of clustering algorithms take distance as an input which is better expressed in a metric system.

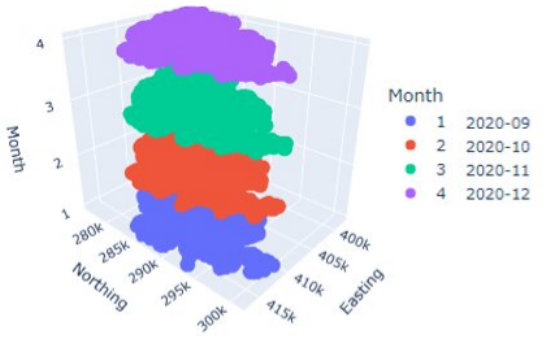

Fig:1 *Spatial-temporal data visualisation using 3D scatterplot*

2. EXPLORATORY SPATIO-TEMPORAL DATA ANALYSIS

The primary aim of an ESTDA is to examine the data for distribution, outliers and anomalies to direct further analysis (Natrella, 2010). To begin with, different types of crime were explored over months.

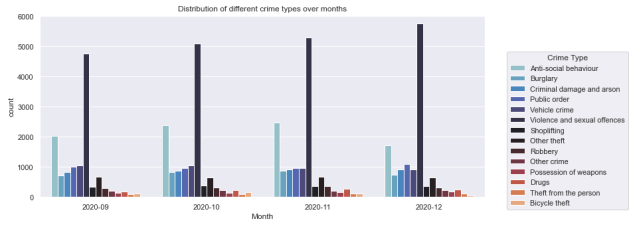

Fig:2 *Bar graph showing crime distribution over four months*

From fig 2 above, it is evident that violence and sexual offences form the primary type of crime with 20912(~40%) in four months. Therefore, it was considered in this study. The second most common crime is anti-social behaviour. Violence and sexual offences, and anti-social behaviour form ~60% of crime in Birmingham. Over four months, the violence and sexual offences cases show an increasing trend.

2.1 Kernel Density Estimation (KDE):

KDE is specifically helpful in detecting hot spots due to the series of estimations which are made over a grid placed on the entire point pattern. These estimations show the intensity at a particular location and therefore detect the highs and lows of point pattern densities (Kalinic and Krisp, 2018).

From plots (fig 3), it is evident that there is a higher probability of violence and sexual offences cluster occurring in the central region of Birmingham, particularly in the Centre-West. In other words, these areas of high probability density will form areas of high-density crime clusters. It includes the districts of Ladywood, Perry Barr, Hodge Hill, Erdington and Yardley. The districts in the south of Birmingham have a relatively lower probability, whereas the districts in the north of Birmingham have the most negligible probability.

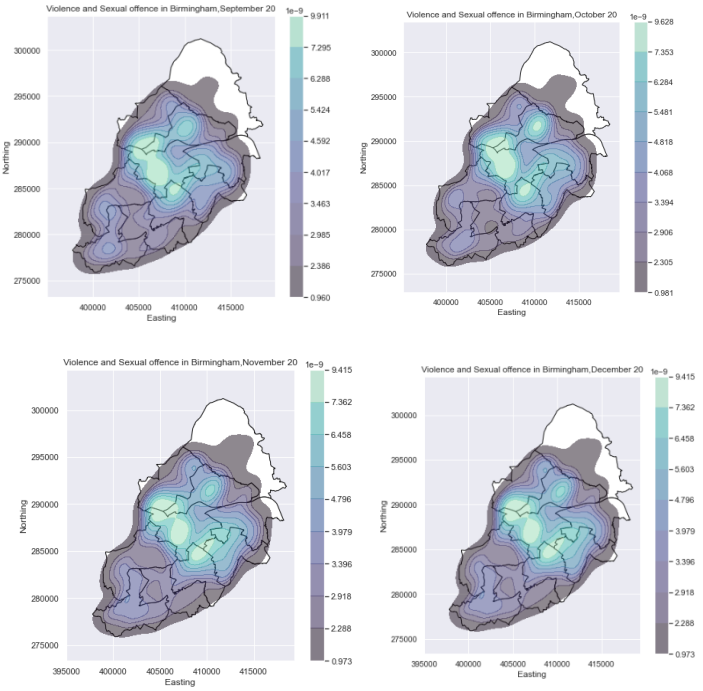

Fig 3: *KDE plots for four months of VSO data*

2.2 Getis-Ord Gi*:

The level of autocorrelation varies due to spatial heterogeneity. And therefore, it is vital to use measures such as LISA statistics which help inform the nature of the local distribution of crime.

Gi* compares local averages to global averages and helps identify if the local pattern of crime is different from what is generally observed across the whole study area. Here, Gi* is used to complement the KDE. The results of Getis-Ord Gi*s were prepared using ArcMap. The district of Ladywood has a hot spot with 95% confidence indicating more intense clustering of high values whereas the district of Sutton Field has a cold spot with 90% confidence implying more intense clustering of low values in the area. In the rest of the districts, the results are not significant.

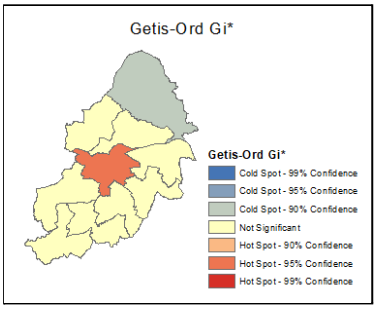

Fig 4: *Getis-Ord Gi* analysis*

3. METHODOLOGY AND RESULTS

In this study, density-based clustering was chosen because it can deal with noise, i.e., not all points need to be part of a cluster, and clusters can have arbitrary shapes, which prove to be handy when dealing with large spatial databases. The basic idea behind these types of clustering methods is that clusters are dense regions in the data separated by regions of lower object density.

3.1 DBSCAN

DBSCAN uses the density of points to generate clusters of connected points and later evaluates whether the other points are connected to those clusters. In contrast, other clustering algorithms such as k-means use the distance of points to the cluster centre. Therefore, it takes two parameters as input, i.e. Minimum points (minPts) and epsilon (eps). minPts is defined as the minimum number of points required to form a cluster and eps is defined as the maximum distance within which two points are considered reachable. These parameters are susceptible and must be appropriately set depending on the domain-specific knowledge.

DBSCAN classifies the points into three categories, namely, core points, reachable points(border) and outliers(noise points):

Core Point: A point is considered a core point if it has more than a specified number of points (minimum points) within epsilon. These points lie inside the cluster.

Border/Reachable Points: A point is a border point if it has fewer than minimum points within the epsilon, but it lies in the neighbourhood of the core point.

Outliers/Noise Point: A point that is neither core nor a border point is called noise.

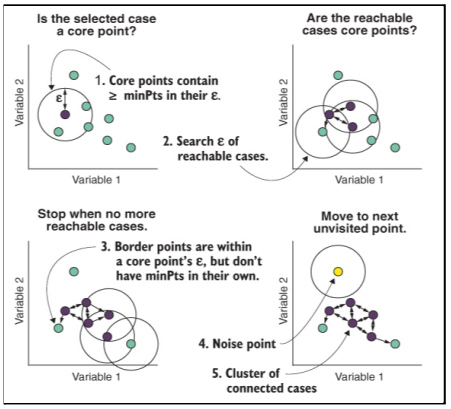

Fig 5: *DBSCAN algorithm*

A case is selected randomly, and if its epsilon radius (ϵ) contains minPts cases, then it is a core point. Reachable cases of this core point are evaluated the same way until there are no more reachable cases. This network of density-connected cases is called a cluster. Cases that are reachable from core points but are not themselves core points are border points. The algorithm moves on to the next unvisited case. Cases that are neither core nor border points are labelled as noise (Rhys, 2015).

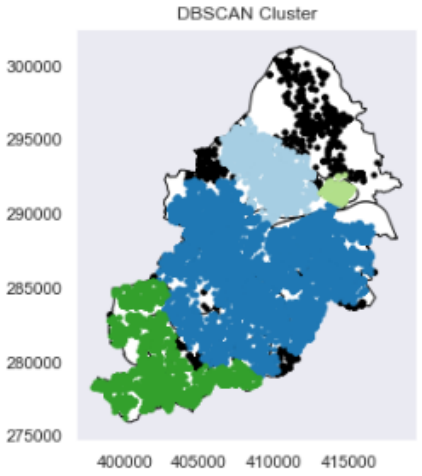

Fig 6: *Output of DBSCAN for VSO between September 2020-December 2020*

Epsilon values of 845 and the minimum point of 155 were chosen using the silhouette score. The DBSCAN algorithm forms four clusters. There is one large cluster in the centre which includes the districts of Ladywood, Hodge Hill, Yardley, and Hall Green. There is one prominent cluster in the North and South of the largest cluster. Black colour denotes the outliers or the noise point. The clusters are formed in the centre and South of Birmingham as seen in the KDE plot.

3.2 ST-DBSCAN

ST-DBSCAN is an extension of DBSCAN. It takes three parameters as input rather than two parameters as in the case of DBSCAN. The three parameters are S-Eps (spatial reachable interval), T-Eps(temporal reachable interval) and minPts (minimum number of reachable points) (Singh and Meshram,2018).

In other words, ST-DBSCAN has two epsilons. Eps1 is the spatial attribute distance parameter (latitude and longitude). Eps2 is the non-spatial attribute distance parameter (time) MinPts is the minimum number of points within Eps1 and Eps2 distance of a point (Birant and Kut, 2006). If an area is dense, it will contain more than the minimum number of points.

In fig 7, ST-DBSCAN is plotted for the four months: September 2020, October 2020, November 2020 and December 2020. In the first month, eight clusters are recognised, out of which, one (black) is the outlier. In the next month (October), the cluster will reduce to 6. Furthermore, in the last two months, it has become 5.

The common trend in all these months is that no clusters were formed in the North of Birmingham, and there is always one big cluster in the centre of Birmingham, as seen in the KDE plot where density was high. If all these ST-DBSCAN results are aggregated together, we will find the same result as the DBSCAN algorithm. It can be concluded from the results above that the density cluster changes slightly when the temporal neighbourhood is changed.

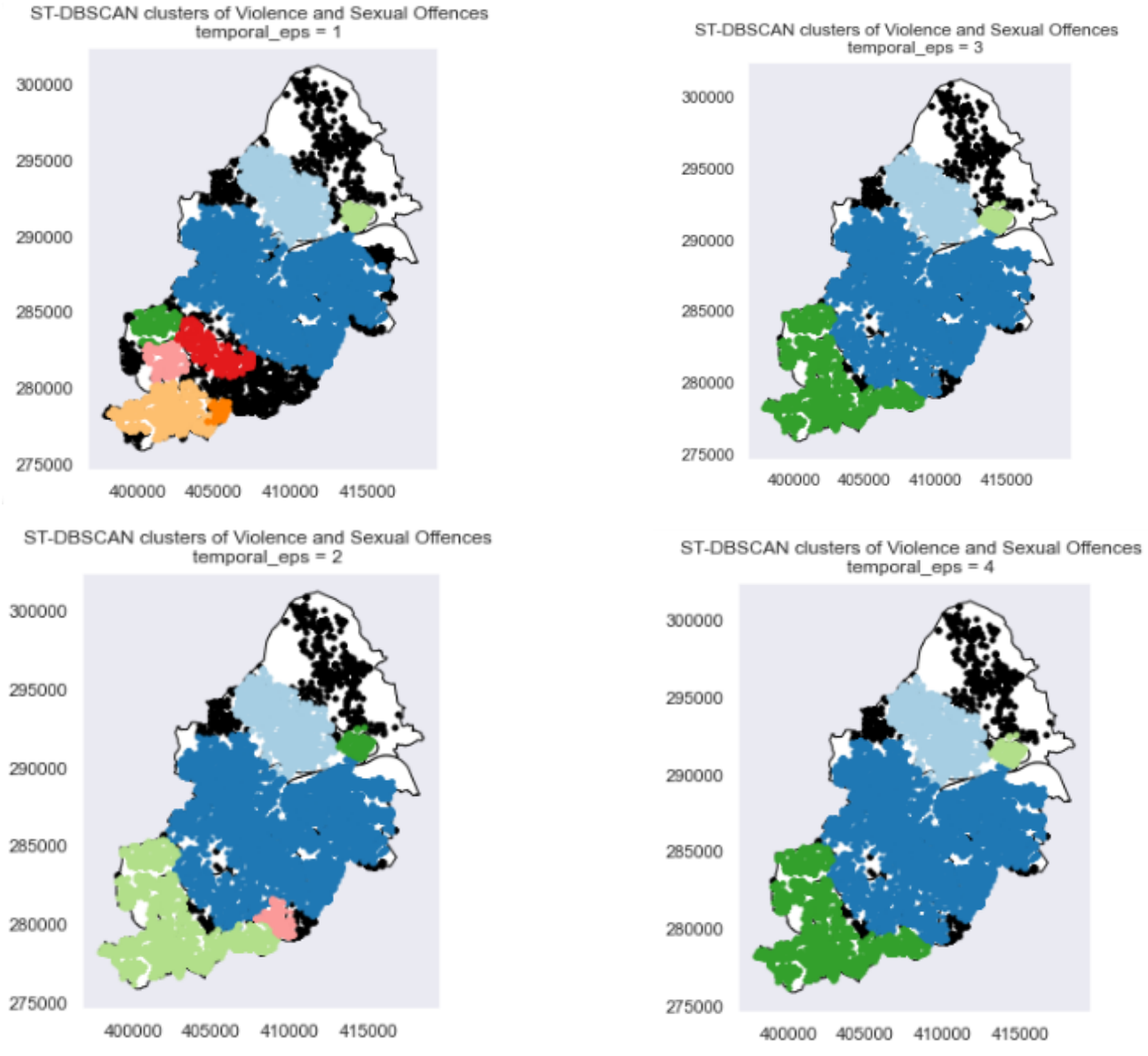

Fig 7: *Result of ST-DBSCAN*

3.3 Validating clusters:

Methods like elbow plots were not used to find the optimal parameters of the algorithms. There are various cluster validation methods, such as the Silhouette score or Dunn Index, which can be used to determine how good clustering is. However, these indexes alone cannot help determine the best parameters as they do not consider the various underlying factors responsible for the occurrence of crime in a particular area. Therefore, the silhouette score and knowledge of the data were used to tune the model.

3.3.1 Silhouette score: It allows us to study and understand the separation between the clusters formed. In other words, it measures how close each object in one cluster is to another object in another cluster (Ogbuabor and Ugwoke, 2018). The value ranges from -1 to +1, where -1 indicates that the objects are not properly clustered while +1 means the correct clustering of objects.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

where a is the mean intra-cluster distance; b is the mean nearest-cluster distance for each sample (Thinsungnoen, et al., 2015).

A good clustering is one which minimises the distance within the clusters (called intra-cluster) and maximises the distance between the clusters (called inter-cluster). In this study, the best silhouette score was 0.17 where the epsilon value was 845 and minPts was 165. This value indicates that the clusters are overlapping. This indicates results are not good enough.

4. DISCUSSION

4.1 Understanding the location of crime clusters using the Index of Multiple Deprivation (IMD)

The Index of Multiple Deprivation is based on 39 distinct variables organised over seven subdomains of deprivation, which are integrated and weighted to generate the Index of Multiple Deprivation (Birmingham City Council, 2019). It includes indicators such as income, employment, education, health, living environment, barriers to housing and services and so on.

It was found that Birmingham has high levels of deprivation, with 43 percent of the population living in LSOAs in the top 10% of England's most deprived regions (Birmingham City Council, 2019).

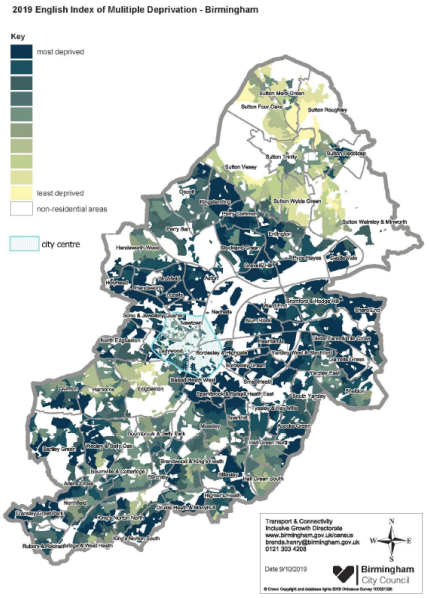

Fig8: *Index of Multiple Deprivation in Birmingham, 2019 (Source: Birmingham City Council, 2019).*

From fig 8, it is evident that deprivation is concentrated in a ring around the city centre. However, significant deprivation spread across the city, particularly to the East and the outer city to the South, West and Northeast. District Sutton Field (in North) has got the least deprived areas.

Crime occurs in a particular region due to the interplay of complex socio-economic factors. Therefore, using the IMD can help give meaningful insights into why the crime occurs in a particular place. On comparing the crime analysis maps produced in the study with the IMD index map, it can be said that the most deprived district is likely to have a high density of crimes Ladywood) whereas the least deprived areas (Sutton Field) are less likely to have crime.

4.2 Model Strength

Unlike other clustering methods, the density-based algorithms like DBSCAN AND ST-DBSCAN have the advantages in many fields, such as:
(i) There is no need to specify the number of clusters in the data a priori. Instead, the algorithm classifies on its own.
(ii) They have the concept of outliers, so not all the points (objects) are required to be part of the cluster.

(iii) they have arbitrary shapes. So, it can find a cluster surrounded by another cluster.

The main difference between DBSCAN and ST-DBSCAN lies in how they define the search boundary (Choi and Hong, 2021). The latter is better as it includes another parameter (temporal epsilon) to deal with spatial and temporal aspects simultaneously. However, the DBSCAN takes less time to compute than ST-DBSCAN, and it also requires comparatively less computational power.

4.3 Limitations

The results of DBSCAN and ST-DBSCAN are sensitive, and therefore, the quality of the model depends on how well the parameters are tuned. For instance, the larger the value of epsilon, the more significant the size of the cluster and vice-versa. Similarly, if minPts are kept too small, more points will be considered noise or outlier. However, if minPts is kept higher, then there may be a case where smaller clusters might be incorporated into the more prominent clusters. Therefore, these models require intuition/expert knowledge to set the parameter values, which is quite challenging.
Moreover, the parameters are fixed for the whole dataset. Due to this, they cannot identify the clusters of varying densities. This becomes quite an important issue when dealing with geographic data such as crime.

In this study, only four months were considered due to many points data (in case of violence and sexual offences). More months should be used better to understand the seasonal or annual variation of clusters. Additionally, the data is available monthly, which restricts the study from monitoring the temporal changes in a day. Finally, running these models with massive datasets requires more computational power, another challenge.

For tuning, a silhouette score was used. It has a constraint that it is only defined if the number of labels is 2<= n_labels <= n_samples -1(sklearn). When it was looped through all the values defined in the range for eps and minPts, if the condition was violated even for one of the combinations, none of the results was processed. Therefore, this may have resulted in losing some better results. Tuning using this is quite cumbersome and requires time and computational power.

5. CONCLUSION AND FUTURE WORK

In this study, we can confidently predict high clustering of crimes in the district of Ladywood. And therefore, it is essential to use domain-specific knowledge to tune the parameters. Further, the analysis should be carried out at the ward level as spatial heterogeneity exists. At the district level, results get relatively aggregated and give a general picture of the examined phenomena.
It is essential to use domain-specific knowledge to tune the parameters. Further, the analysis should be carried out at the ward level as spatial heterogeneity exists. At the district level, results get relatively aggregated and give a general picture of the examined phenomena. To overcome the limitation of DBSCAN, the relative density for the clusters should be defined like in the KDE.

The well-tuned models will be an excellent tool in estimating the clusters and further leading to crime prevention.

6. REFERENCES

1. Choi, C. and Hong, S.-Y. (2021). MDST-DBSCAN: A Density-Based Clustering Method for Multidimensional Spatiotemporal Data. *ISPRS International Journal of Geo-Information*, 10(6), p.391. doi:10.3390/ijgi10060391.
2. Economic Policy Council (2019). *Deprivation in Birmingham*. [online] Birmingham: Birmingham City Council, p.10. Available at: https://www.birmingham.gov.uk/download/downloads/id/2533/index_of_deprivation_2019.pdf.
3. Akbar, S. and Khan, M.N.A. (2014). Critical Analysis of Density-based Spatial Clustering of Applications with Noise (DBSCAN) Techniques. *International Journal of Database Theory and Application*, 7(5), pp.17–28. doi:10.14257/ijdta.2014.7.5.02.
4. Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996, August. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, No. 34, pp. 226-231).
5. Chelly, Z., Smiti, A. and Elouedi, Z. (2012). COID-FDCM: The Fuzzy Maintained Dendritic Cell Classification Method. *Artificial Intelligence and Soft Computing*, pp.233–241. doi:10.1007/978-3-642-29350-4_28.
6. Thinsungnoen, T., Kaoungku, N., Durongdumronchai, P., Kerdprasop, K. and Kerdprasop, N. (2015). *The Clustering Validity with Silhouette and Sum of Squared Errors*. [online] www2.ia-engineers.org. Available at: https://www2.ia-engineers.org/conference/index.php/iciae/iciae2015/paper/view/576 [Accessed 11 May 2022].
7. The Editors of Encyclopaedia Britannica (2019). Birmingham | History, Population, Map, & Facts. In: *Encyclopædia Britannica*. [online] Available at: https://www.britannica.com/place/Birmingham-England
8. crimerate.co.uk. (n.d.). *Birmingham Crime and Safety Statistics*. [online] Available at: https://crimerate.co.uk/west-midlands/birmingham..
9. data.police.uk. (n.d.). *About | data.police.uk*. [online] Available at: https://data.police.uk/about/.
10. Nist.gov. (2019). *1.3.5.17. Detection of Outliers*. [online] Available at: https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm.
11. Advances in Computing Science. (n.d.). [online] Available at: https://rcs.cic.ipn.mx/2014_83/RCS_83_2014.pdf#page=34 [Accessed 11 May 2022].
12. ResearchGate. (n.d.). *(PDF) Repeat Victimisation*. [online] Available at: https://www.researchgate.net/publication/274312437_Repeat_Victimization.
13. Scikit-learn.org. (2019). *sklearn.metrics.silhouette_score — scikit-learn 0.21.3 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.

Code: Code can be downloaded from Github as well along with files required using this link: https://github.com/pratibha1708/crime_clustering.git

Instruction

The code is run in a jupyter notebook. Open the code, import the required libraries, and open the files. Put the .csv files, .shp and code in the same folder. And run the script.

Tuning parameters may take 30-40minutes, may skip that part if want to and can see the output in the notebook.

CODE TRANSCRIPT

```
#importing libraries

import pandas as pd

import geopandas as gpd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

from sklearn.cluster import DBSCAN

from sklearn.metrics import silhouette_score

from itertools import product

from st_dbscan import ST_DBSCAN

from pyproj import CRS

from pyproj import Transformer

import plotly.express as px

#adding data

data_1 = pd.read_csv('sept_2020.csv')

data_2 = pd.read_csv('oct_2020.csv')

data_3 = pd.read_csv('nov_2020.csv')

data_4 = pd.read_csv('dec_2020.csv')
```

Pre-processing data

```
# Data is available for entire midlands, filtering data for Birmingham

data = pd.concat([ data_1, data_2 ,data_3, data_4])

data = pd.concat([ data_1, data_2 ,data_3, data_4])

data = data.drop(data.columns[[0,2,3,10,11]], axis=1)

null_data = data[data.isnull().any(axis=1)]

null_data

data = data[data['LSOA code'].notna()]

data['LSOA name'].nunique()

data =data[data['LSOA name'].str.startswith('Birmingham')]

data
```

EDA: Crime Type

```
sns.set(rc={'figure.figsize': (12,5)})

g = sns.countplot(x = 'Month', data = data, hue = 'Crime type', palette= 'icefire').set(title='Distribution of different crime types over months')

plt.legend(bbox_to_anchor=(1.05, 0.8), loc='upper left', borderaxespad=0, title='Crime Type')

vso =data[data['Crime type'].str.startswith('Violence')]

Vso

vso.index = np.arange(1,len(vso) + 1)

Vso

# Transforming lat long to easting and northing

transformer = Transformer.from_crs('EPSG:4326', 'EPSG:27700')

vso['Easting'], vso['Northing'] = transformer.transform(vso['Latitude'].values, vso['Longitude'].values)

vso

# Pre-processing shapefile

west_midlands_shp = gpd.read_file('2021_06_08_Initial_Proposals_West_Midlands.shp')

birm_boroughs = ['Sutton Coldfield', 'Birmingham Hodge Hill', 'Birmingham Erdington','Birmingham Perry Barr', 'Birmingham Ladywood','Birmingham Edgbaston','Birmingham Yardley','Birmingham Hall Green','Birmingham Northfield','Birmingham Selly Oak']

bs_to_chk = west_midlands_shp['Constituen']

for i in bs_to_chk:

    if i not in birm_boroughs:


west_midlands_shp.drop(west_midlands_shp[west_midlands_shp['Constituen'] == i].index[0], inplace=True)

west_midlands_shp.plot()

west_midlands_shp.to_file('birmingham')

pts = pd.DataFrame({'latitude': vso['Latitude'].to_list(), 'longitude': vso['Longitude'].to_list()})

geometry = gpd.points_from_xy(pts['latitude'], pts['longitude'])

west_midlands_shp = gpd.read_file('birmingham')

bir_shp = west_midlands_shp

bir_shp
```

KERNEL DENSITY ESTIMATION (KDE)

```
# dividing data into different months for KDE plot

sep_20 = vso.iloc[:4767,]

oct_20 = vso.iloc[4767:9859,]

nov_20 = vso.iloc[9859:15161,]

dec_20 = vso.iloc[15161:,]

#function to plot KDE for different months

def kde(data,month,num):
```

```
    i, num = plt.subplots(1, figsize=(7, 7))

    bir_shp.plot(color='white', edgecolor='black',ax=num)

    sns.kdeplot(
ax=num,x=data['Easting'],y=data['Northing'],data=data, fill=True,
cmap='mako', alpha = 0.5, cbar=True,).set(title ='Violence and
Sexual offence in Birmingham,' + str(month));

    plt.show()
```

##Making data suitable for clustering

```
# converting months to numricals

vso['Month'] = [str(i).replace("2020-09", "1") for i in vso['Month']]

vso['Month'] = [str(i).replace("2020-10", "2") for i in vso['Month']]

vso['Month'] = [str(i).replace("2020-11", "3") for i in vso['Month']]

vso['Month'] = [str(i).replace("2020-12", "4") for i in vso['Month']]

vso_final = vso.drop(vso.columns[[1,2,3,4,5,6]], axis=1)

vso_final

# interactive Spatial-temporal data visualation

df = vso

fig = px.scatter_3d(df, x='Easting', y='Northing', z='Month',

              color='Month')

fig.show()

#converting dataframe to array

vso_array = vso_final.to_numpy()

vso_array

#
Source:https://github.com/eren-ck/st_dbscan/blob/master/demo/d
emo.ipynb



def plot(vso_array, labels, title):

    colors =
['#a6cee3','#1f78b4','#b2df8a','#33a02c','#fb9a99','#e31a1c','#fdbf
6f','#ff7f00','#cab2d6','#6a3d9a']

    bir_shp.plot(color='white', edgecolor='black')



    plt.grid()

    for i in range(-1, len(set(labels))):

        if i == -1:

            col = [0, 0, 0, 1]

        else:

            col = colors[i % len(colors)]



        clust = vso_array[np.where(labels==i)]



        plt.title( title)

        plt.scatter(clust[:,0], clust[:,1], c=[col], s=10,)

    plt.show()
```

DBSCAN

```
dbscan_model= DBSCAN(eps=845, min_samples=155,
metric='euclidean' ).fit(vso_final)

plot(vso_array[:,1:3], dbscan_model.labels_, 'DBSCAN Cluster')
```

ST-DBSCAN

```
temporal_eps=[1,2,3,4]

for st_eps2 in temporal_eps:

    st_dbscan = ST_DBSCAN(eps1 = 845, eps2 = st_eps2,
min_samples = 155)

    st_dbscan.fit(vso_array)

    plot(vso_array[:,1:3], st_dbscan.labels,'ST-DBSCAN clusters of
Violence and Sexual Offences \n'+'  temporal_eps = ' +
str(st_eps2))
```

Tuning hyper-parameters of DBSCAN using silhouette score

```
#Source:
https://towardsdatascience.com/explaining-dbscan-clustering-18eaf5c8
3b31

pca_eps_values = np.arange(700,1200,20)

pca_min_samples = np.arange(400, 700, 50)

pca_dbscan_params = list(product(pca_eps_values,
pca_min_samples))

pca_no_of_clusters = []

pca_db_index = []

pca_epsvalues = []

pca_min_samp = []

for p in pca_dbscan_params:

    pca_dbscan_cluster = DBSCAN(eps=p[0],
min_samples=p[1]).fit(vso_final)

    pca_epsvalues.append(p[0])

    pca_min_samp.append(p[1])


pca_no_of_clusters.append(len(np.unique(pca_dbscan_cluster.la
bels_)))

    pca_db_index.append(silhouette_score(vso_final,
pca_dbscan_cluster.labels_))

pca_eps_min = list(zip(pca_no_of_clusters, pca_db_index,
pca_epsvalues, pca_min_samp))

pca_eps_min_df = pd.DataFrame(pca_eps_min,
columns=['no_of_clusters', 'silhouette score', 'epsilon_values',
'minimum_points'])

pca_eps_min_df
```