Chicago Crime Data Graphical Analysis

Final Project Report

Sébastien Penet
Applied Mathematics
CentraleSupelec
sebastien.penet@student.ecp.fr

Meryem Ben-Goumi
Applied Mathematics
CentraleSupelec
meryem.bengoumi @student.ecp.fr

Pierric Maillard de la Morandais
Applied Mathematics
CentraleSupelec
pierric.maillard-de-la-morandais@student.ecp.fr

Pierre Meziane
Applied Mathematics
CentraleSupelec
pierre.meziane@student.ecp.fr

ABSTRACT

Our code and data is available here.

For our project, we started by reproducing an approach to crime prediction explained in the paper *Spatio-Temporal prediction of crimes using network* [1]. To do so, we first build a multi-layered network reflecting the crime activity and the social characteristics of Chicago's community areas. Such social characteristics include the number of schools and the number of registered 311 service calls. Once the network is built, the next step is to extract features and apply different types of regressors. The output is, for each month of the year 2015, the number of crimes in each community and for each crime type. The most accurate predictions are obtained using a Support Vector Regression.

In a second part, we focus more on optimizing the location of the police patrols by solving the optimization problem that consists in minimizing the sum of distances from the optimal police patrol location to the location of every crime in our dataset. We then solve the same problem using an approximation of the distance between two nodes. We can then evaluate this approximation by computing the distance between the solution node in the exact method and the approximative one, after defining a criterion which, if satisfied, validates our approximation.

MOTIVATION

Chicago city has one of the United States' highest murder rates. On average, approximately 10 people are shot every day in Chicago. Besides, more and more information about the city of chicago and its crime activity is publicly available. As such, data analysis techniques may help us reduce the number of crimes registered in the city of Chicago.

PROBLEM DEFINITION

Our study will be based on the dataset "Crimes - 2001 to present" available on the City of Chicago's website[1].
The objective of our work is to answer two questions:

- How could we perform a spatio-temporal prediction of crimes in Chicago?
- Where should police patrols be located in order to better address crimes?

RELATED WORK

A lot of researchers have already tackled the problem of predicting crimes. For our project, we decided to focus on one of the most advanced research work done on this topic : the paper *Spatio-Temporal prediction of crimes using network,* by Saroj Kumar Dash, Ilya Safro, Ravisutha Sakrepatna Srinivasamurthy. We consider their work particularly interesting in terms of graph theory because they introduce social networks in their prediction of crimes.

METHODOLOGY

To adress our problem, we followed the following steps:

1. Preparation of data: downloading it from Chicago's website, and selecting the parts that we would be interested in.
2. Spatio-Temporal Analysis of crimes: observing the geographic and temporal behavior of crime activity in Chicago
3. Prediction of Crimes based on Chicago's social activity: reproducing the approach described in the paper [2].
   - Building a network reflecting crime activity and social factors
   - Running Predictive models
   - Checking the accuracy of the predictions
4. Optimization of the location of police patrols in Chicago
   - Minimizing the sum of distance from the police patrol to all the crimes in the data set using an exact method
   - Minimizing the same sum using an approximation
   - Comparing both the above methods

1 Preparation of the Data

After downloading it from Chicago's website, let's start by opening the available data with pandas to represent it as a dataframe. Below is an overview of the raw data.

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	Ward	Community Area	I Co
0	10000092	HY189866	03/18/2015 07:44:00 PM	047XX W OHIO ST	041A	BATTERY	AGGRAVATED: HANDGUN	STREET	False	False	...	28.0	25.0	0
1	10000094	HY190059	03/18/2015 11:00:00 PM	066XX S MARSHFIELD AVE	4625	OTHER OFFENSE	PAROLE VIOLATION	STREET	True	False	...	15.0	67.0	
2	10000095	HY190052	03/18/2015 10:45:00 PM	044XX S LAKE PARK AVE	0486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	False	True	...	4.0	39.0	0
3	10000096	HY190054	03/18/2015 10:30:00 PM	051XX S MICHIGAN AVE	0460	BATTERY	SIMPLE	APARTMENT	False	False	...	3.0	40.0	0
4	10000097	HY189976	03/18/2015 09:00:00 PM	047XX W ADAMS ST	031A	ROBBERY	ARMED: HANDGUN	SIDEWALK	False	False	...	28.0	25.0	
5	10000098	HY190032	03/18/2015 10:00:00 PM	049XX S DREXEL BLVD	0460	BATTERY	SIMPLE	APARTMENT	False	False	...	4.0	39.0	0

Figure 1: **Head of Raw Data represented as a Dataframe**

Among others, the data contains the date of the crime, the location (down to the block number), the type of crime and if the crime lead to an arrest or not.

Let's simplify the data by selecting only the fields that seem the most relevant ('ID', 'Date', 'Primary Type' ,'Arrest', 'Domestic', 'Latitude', 'Longitude').
Moreover, the initial data contains 6,779,002 rows. We will work with a fraction of this data to ensure the performance of our algorithms. It will also allow us to save a testing set to further confirm our findings on out-of-sample data after having built our models. For this purpose, all crimes being dated, we can simply focus on one particular time window. The available data ranges from 01/01/2001 01:00:00 AM to 12/31/2017 12:55:00 PM. We decide to focus on crimes dated between 01/01/2001 01:00:00 AM and 01/01/2010 01:00:00 AM.
As a next step, we can plot the geographic distribution of the crimes.

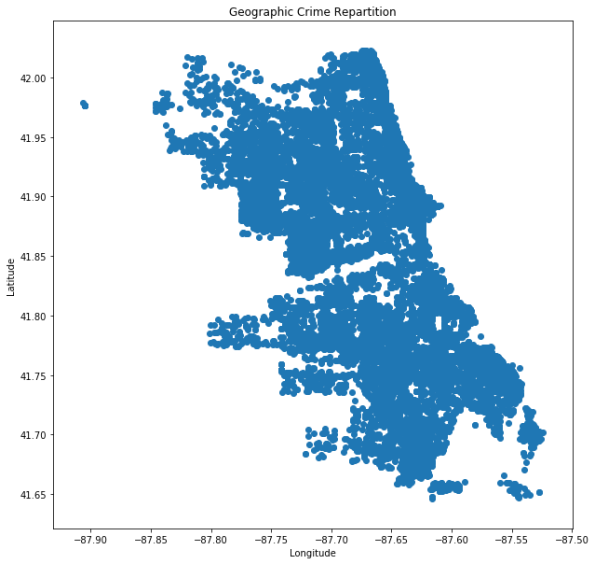


Figure 1: **Geographic Distribution of the Crimes between 01/01/2001 and 01/01/2010**

To further facilitate the analysis, let's now focus on one particular category of crime. Let's choose Narcotics for example, and replot our repartition.

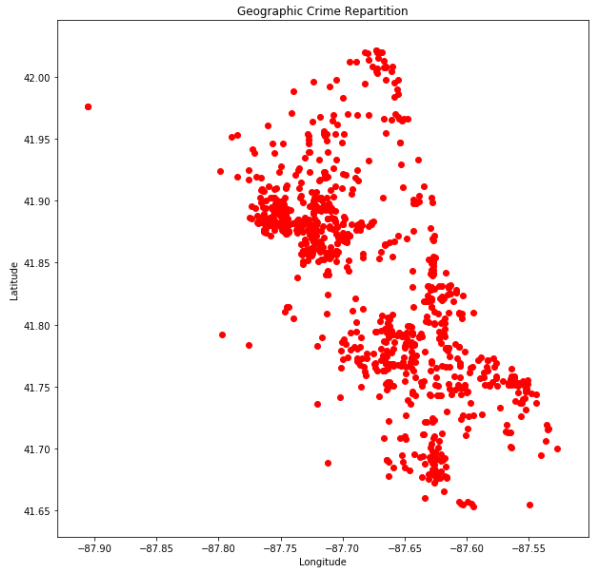


Figure 2: **Geographic Distribution of Crimes related to Narcotics between 01/01/2001 and 01/01/2010**

2 Spatio-Temporal Analysis of Crimes

2.1 Identification of Geographical Clusters

To identify geographical clusters in the above repartition, we can use a K-Nearest Neighbors algorithm.

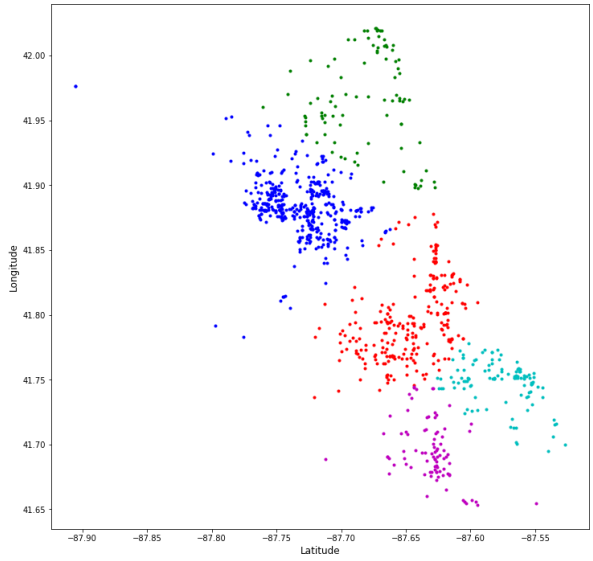


Figure 4: **Clusters of Crimes related to narcotics between 01/01/2001 and 01/01/2010**

2.2 Crime Seasonality

Plotting the number of crimes over several year gives us a clear idea of the seasonality of crime in Chicago, as shown by the below graph.

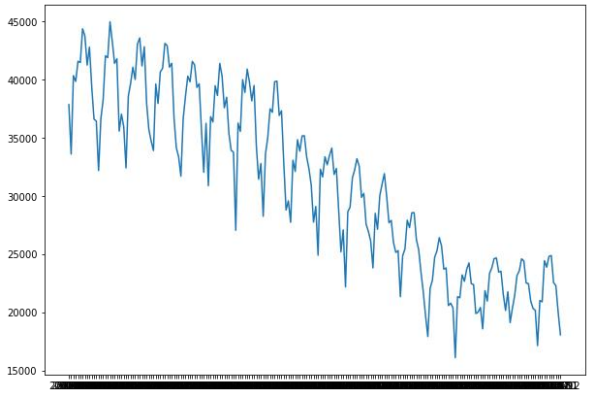


Figure 3: **Evolution of the number of crimes in Chicago on several years**

Let's note that this graph also shows a general decline in the number of crimes over the years.
A closer look to the evolution of crime on a given year provides us with a more precise analysis of the seasonality phenomenon.

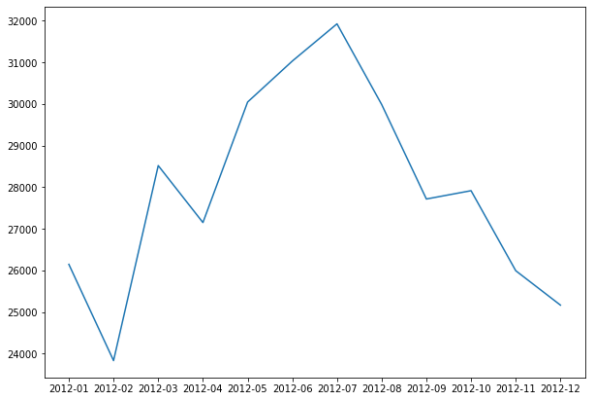


Figure 4: **Evolution of the number of crimes in Chicago on a given year**

This graph shows that the number of crimes is way higher in summer than it is in winter.

3 Prediction of Crimes based on Chicago's social activity

The objective of this part is to reproduce the approach described in the paper *Spacio-Temporal prediction of crimes using network,* by Saroj Kumar Dash, Ilya Safro, Ravisutha Sakrepatna Srinivasamurthy, using the code: https://github.com/Ravisutha/CrimePrediction.

Their approach consists in clustering the city of Chicago according to their 77 community areas ($\mathcal{C}$). The network built contains the crime data (crime types ($\mathcal{T}$) registered in the community area) and other social factors describing each community area: the set of schools ($\mathcal{S}$), police stations ($\mathcal{P}$) and 311 service requests ($\mathcal{R}$).

The graph associated to this multi-layered network is an undirected weighted graph ($\mathcal{V}, \mathcal{E}$), with different types of nodes and edges. The set of nodes $\mathcal{V}$ is composed of the six layers described above:

$$V = C \cup T \cup S \cup P \cup R$$

3.1 Building the Network

As an example, let's build the network for year 2014. The same approach could be reproduced for any other year for which crime data is available. We obtain a graph with 3407 nodes and 17696 edges:

- $|\mathcal{C}|$ = 77 communities in Chicago. Two community areas c - c are connected in the graph if they share a border in common.
- $|\mathcal{S}|$ = 2977 schools. There is an edge between a school s and the community c it is located in, weighted by the school average ACT score. As such, the weight of the edge s - c reflects the school's level of education.
- $|\mathcal{P}|$ = 24 police stations. Each crime type t is connected to the police station p corresponding to the crime location. A police station p is connected to the community areas corresponding to it.
- $|\mathcal{T}|$ = 321 crime types according to the primary & secondary descriptions provided by the Police Department in Chicago. A crime type t is connected to a community c and to a police station p if it is registered in it in 2014. The weight of the edges c - t and p - t is the number of crimes of such type.
- $|\mathcal{R}|$ = 8 types of 311 service requests. An edge c - r between a service request and a community, weighted by the number of requests registered in the community, reflects the reactivity of the community in solving its problems.

3.2 Network Visualization & Analysis

a. Graph visualization

Running the *make_network.py* script for year 2014 generates a graph file *network_2014.graphml.* To visualize the graph and extract statistics, we use Gephi software. Applying a ForceAtlas 2 graph layout algorithm using Gephi provides the visualization below:

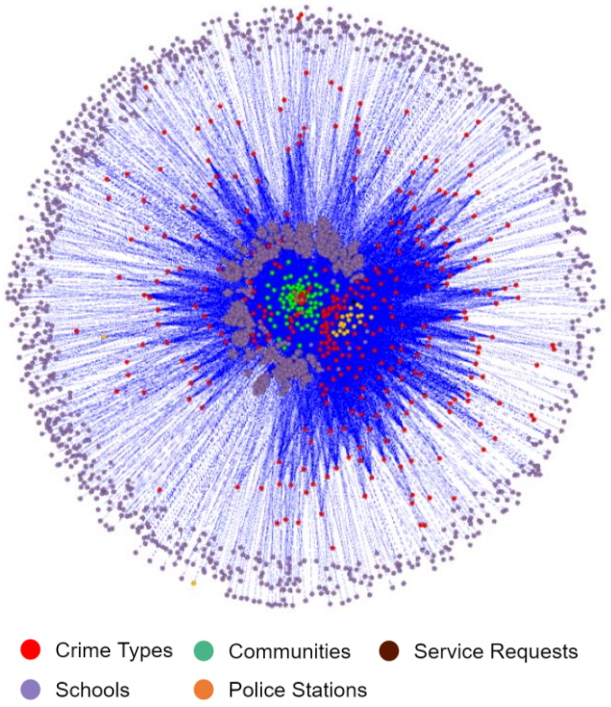


Figure 5: **Visualization of the multi-layered graph for year 2014.**

b. Graph Statistics

The graph obtained for year 2014 has the following statistics, computed using Gephi:

Network Overview	
Number of Nodes	3407
Number of Edges	17696
Average Degree	10.388
Average Weighted Degree	451.71
Network Diameter	4
Graph Density	0.003
Modularity	0.249
Average Path Length	3.672
Number of Triangles	7644
Average Clustering Coefficient	0.019

Table 1: **Graph Statistics**

3.3 Feature extraction

a. Extracting data from the Network

The features extracted for each community are given by the network's edges: the number of crimes of each primary type registered in it, the number of schools and police stations in the community, and the number of 311 service calls registered in it.

b. Similarity between communities

To add more features reflecting the crime activity and social factors of a community, we select the two most similar communities.

To do so, the first step is to compute the similarity between all pairs of communities, thus generating a similarity matrix.
The formula used for computing the similarity is the **Random Walk Similarity**, which is often used in network analytics for weighted and undirected graphs. Based on a Markov-chain model of the Random Walk algorithm, this similarity reflects the probability of a random walk to constitute a path from one node to another. Such probability is computed using the pseudo-inverse of the Laplacian of the graph. Hence, the similarity between nodes increases when the number of paths connecting these two nodes increases and when the length of the path decreases.

After computing the similarities between all communities, the next step is to select, for each community ***c***, the two most similar communities ***c1*** and ***c2***. Then, we add the features of ***c1*** and ***c2*** to the features of ***c***. These new features allow a better understanding of the community's crime and social activity. In fact, we realize that two communities can be very similar in terms of crime activity without necessarily being geographically close to each other. It happens when two communities that are geographically "far" from each other but have comparable social factors.

3.3 Predicting crimes

a. Predictive Models

To predict the number of crimes for a given type, for all communities, for all months of 2015, we reproduce two of the three approaches explained in the paper *Spacio-Temporal prediction of crimes using network* :

- **Polynomial regression** of degree 2, which consists in trying to fit a quadratic polynomial function to the dataset.
- **Support Vector Regression**, which is similar to the renowned machine learning classifier "Supper Vector Machine" (SVM). The difference with the classifier is

that the prediction is not a class or category but a number of crimes.

The training used to fit the model is the dataset available for years 2011 to 2014. The testing set is the year 2015. To perform the predictions, we used the script coded by the authors of the paper mentioned above *predict.py*.*

b. Top 10 Crime Types

The table below shows the actual registered number of crimes and the predictions given by both methods for year 2015, for the ten types of crime that have the highest number :

Crime Type	Actual Number	SVR Prediction	Polynomial Prediction
Theft	57319	67230	67049
Battery	17041	55715	55084
Criminal Damage	28671	32492	34876
Narcotics	23837	33708	35643
Assault	17041	18616	20609
Deceptive Practice	15676	13548	16516
Burglary	13183	19777	21358
Motor Vehicle Theft	10070	14054	15189
Robbery	9638	11991	12503
Criminal Trespass	6400	7950	9353

Table 2: **Predictions & Actual Number of crimes in 2015 for top 10 crime types.**

We notice that we have the same ten crime types that have the highest number for both methods and for the actual number. However, they are not ranked the same way depending on the prediction method.

c. Accuracy Measure

To compare the predictions obtained with the two methods, we plotted below a box plot representing the ratio of predicted over actual values for all months of the year 2015:

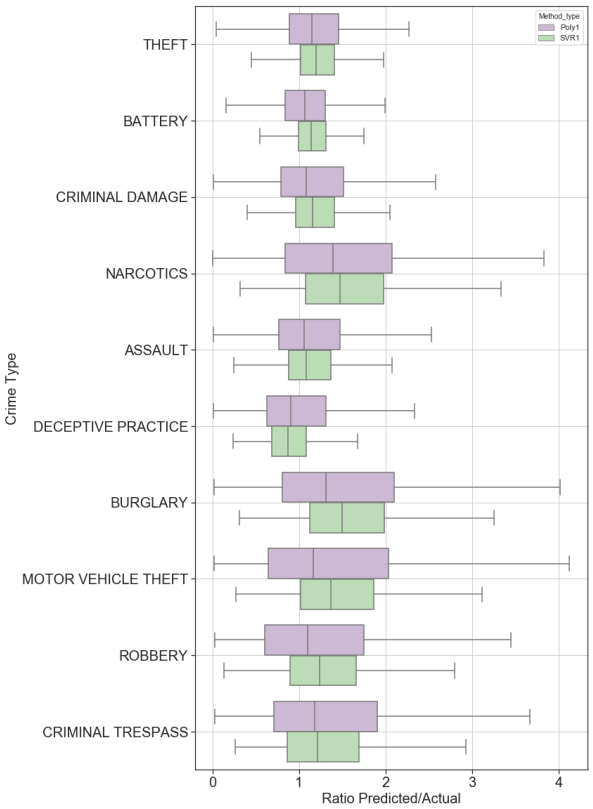


Figure 6: **Comparison of Predicted/Actual ratio for the top 10 crime types**

As we can it in the boxplots above, the Support Vector Regression provides better results than the Polynomial Regression.

d. Crime Seasonality

In the first section, we discussed the seasonality of crimes in Chicago : the number seemed to increase in summer, and decrease in Winter. We can see in the figures plotted below that the same observation can be made for the predictions obtained with SVR Model.

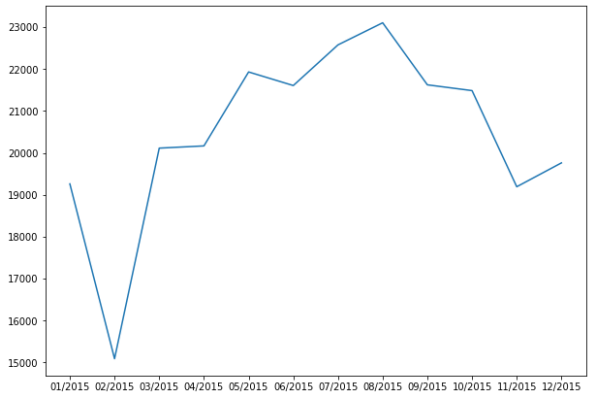


Figure 7: **Actual Number of crimes for different months of year 2015**

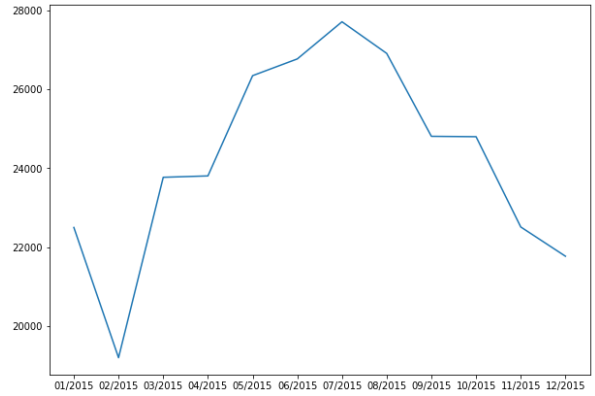


Figure 8: **Number of crimes for different months of year 2015 predicted using SVR method**

The fact that we observe a seasonality phenomenon confirms the quality of the SVR method.

It should be noted that the seasonality phenomenon is not reflected in all crime types. In fact, for "**theft**" type, we note an increase in summer & a decrease in winter. However, for "**narcotics**", this correlation with the weather is not observable, as it is shown in the figures plotted below.

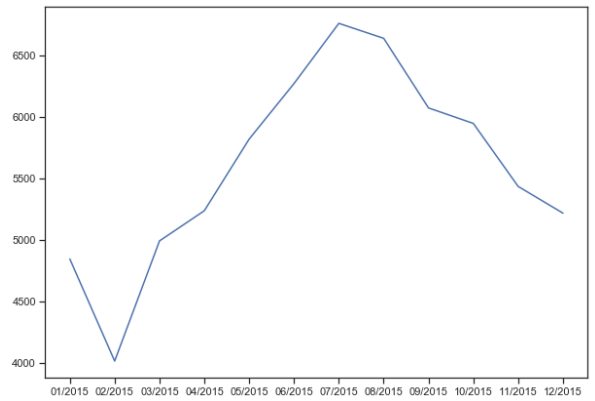


Figure 9: **Number of "theft" crimes for different months of year 2015 predicted using SVR method**

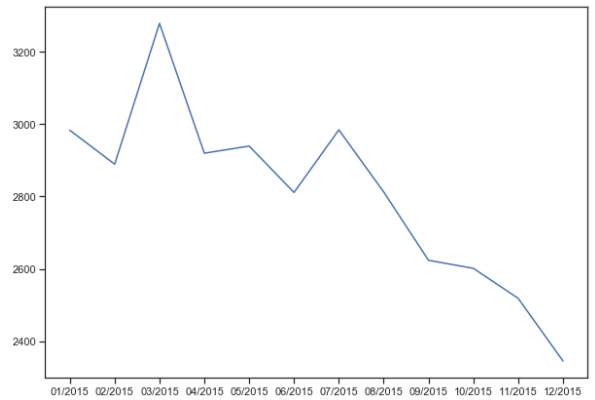


Figure 10: **Number of "Narcotics" crimes for different months of year 2015 predicted using SVR method**

3.4 Further Enhancements

A possible enhancement of the prediction is to predict crimes with specific geospatial coordinates instead of doing so for each community. Doing so would require to have access to social characteristics per geospatial coordinates.

3.1 Second approach: Custom Clustering

4 Computing the Best Possible Police Patrol

Our goal is now to try and identify the optimal locations for police patrols in order to ensure the most efficient crime prevention. We aim at positioning the patrols in order to reduce the shortest paths from the patrols to the main clusters of crime.

To simplify our approach, we are going to proceed to the clustering of the crimes, and place exactly one police patrol in each of these clusters.
This will allow us to reduce our optimization problem to the positioning of a unique patrol, problem way easier to reduce than the optimization of several patrols.

To proceed to our clustering, we use a previously the KMeans algorithm, using a number of 100 clusters, which is going to provide us with a problem relatively quick to solve.
The clustering using the Kmeans algorithm is given below.

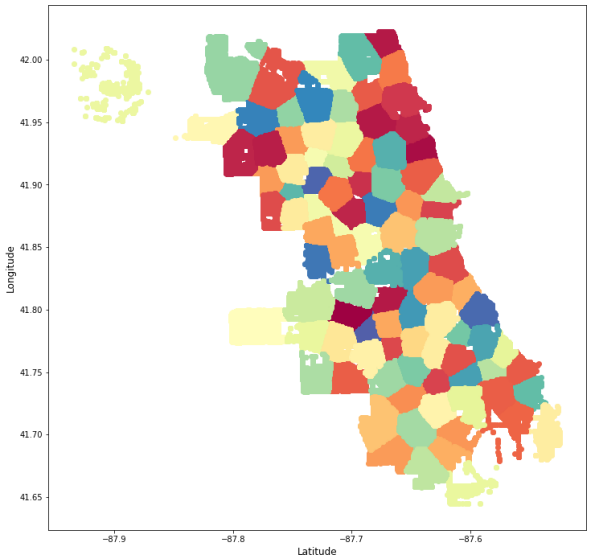


Figure 11: **Crime Clustering using the KMeans algorithm**

We can then choose one of these clusters on which we are going to focus our analysis.
For example, let us choose the fifth cluster whom map representation is given below.

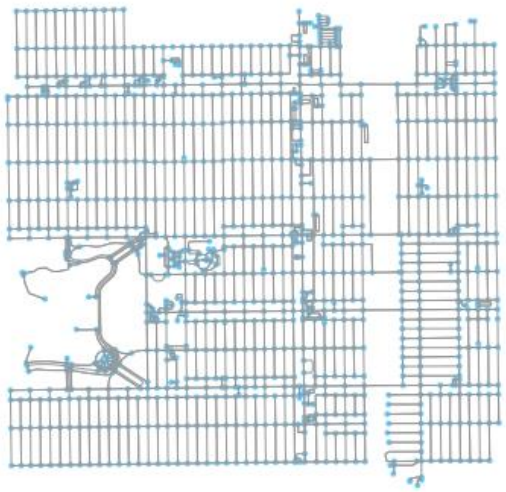


Figure 12: **Map Representation of the Fifth Cluster**

We can then compute the ideal police station location for a given year in this particular district by solving the exact optimization problem given below:

$$X_1^* = \underset{X}{\text{Argmin}} \sum_{X' \in G} d_1(X, X')$$

Where d_1 is the actual street distance (the distance "seen" by a police patrol using existing streets of Chicago) and G is the set of all crimes.

Figure 13: **Representation of the d1 distance between two addresses in Chicago**

The solution is represented below, with each green dot corresponding to a crime and the red dot being the solution of the optimization problem (i.e. minimizing the sum of distances of all the paths drawn in blue).

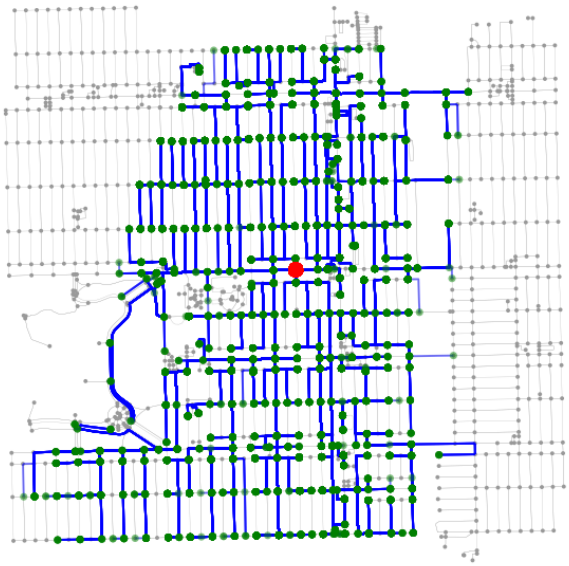


Figure 14: **Solution Node in the Fifth Cluster using actual street distances**

Another less cost-intensive approach is to solve the above using a distance as the crow flies, i.e. computing distances in straight lines instead of taking into account existing streets of the city of Chicago.
This new problem can be formulated as shown below:

$$X_2^* = \underset{X}{\text{Argmin}} \sum_{X' \in G} d_2(X, X')$$

Where d_2 is the usual distance in R2.
An example of the d_2 distance is represented below.

Figure 15: **Representation of the d2 distance between two addresses in Chicago**

In this situation, the solution node is computed in a very straightforward manner. In fact, it simply corresponds to the average node (which is the center of our cluster as considered previously).
The solution in this case is given below.

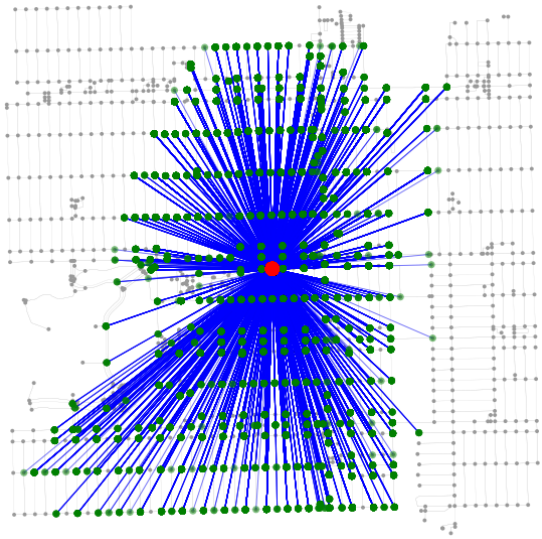


Figure 16: **Solution Node in the Fifth Cluster using straight line distances**

EVALUATION

To evaluate the quality of the above approximation, it can be interesting to compute the distance between the solution node given by the exact method and the solution node given by the approximation.
We must then define a threshold for this distance, above which we can consider that the approximation is not suitable for this problem.
We decide to consider the approximation to be good if the distance D between our solution nodes is less than half the average distance between the police patrol (in the exact case), and each crime.

In the situation described above, the distance between the two solutions nodes is given by

$$D = d_1(X_1^*, X_2^*)$$

where we chose d_1 as this distance is the most representative for this problem as it is the distance actually usable by a police car.

In our case, we get $D = 352.63\ m$

The average distance between the patrol and each crime is 792.51 m so our approximation is satisfying.

CONCLUSIONS

As a conclusion, the last approach is a fairly good approximation. We will then choose this approach over the first one when computing the optimal police patrol for a larger number of crimes. Note that a compromise shall be made between the anteriority of the crimes (crimes that happened 10 years ago are less representative than crimes that happened one week ago), and the number of crimes taken into account (the more crime, the more accurate the solution).
As a next step, the approach on police patrols could be applied to predictions obtained in the third section. To be able to do so, the predictive approach should be improved in order to perform predictions per geospatial coordinates rather than per community.

REFERENCES

[1] Crimes - 2001 to present
https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2
[2] Saroj Kumar Dash, Ilya Safro and Ravisutha Sakrepatna Srinivasamurthy , Spatio-temporal prediction of crimes using network analytic approach
[3] Natarajan Meghanathan, Using Machine Learning Algorithms to Analyze Crime Data
[4] Mami Kajita and Seiji Kajita, Crime Prediction by Data-Driven Green's Function method

APPENDICE: Analysis of the Graph built in Section 3.2

a. Graph Modularity: community Detection

Here, we use community detection algorithm based on edge weights. It detects 7 communities:

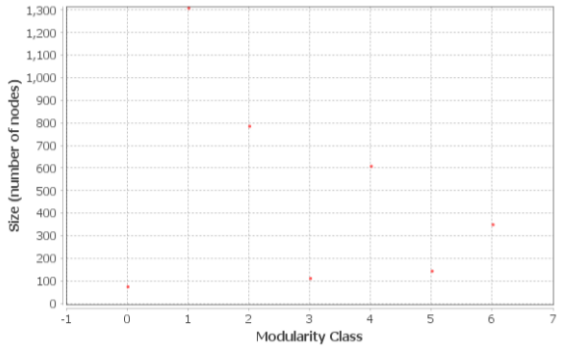
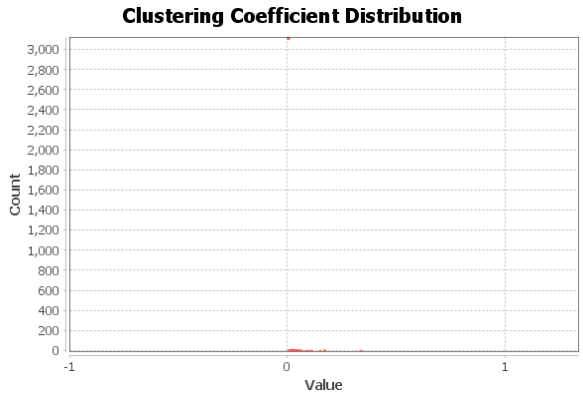


Figure 6: **Size distribution of the graph.**

With a modularity of 0.249, the graph does not really have a significant community structure.

b. Triangles and Clustering Coefficient

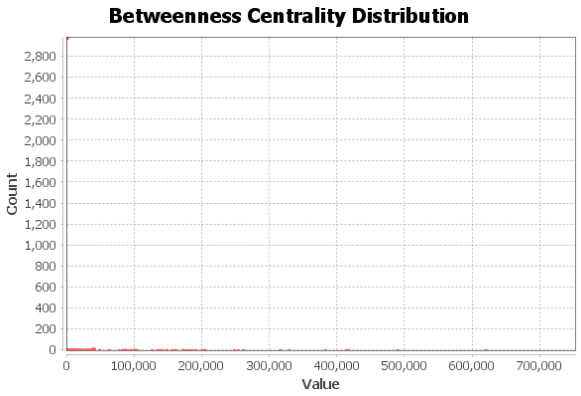
The graph has a total number of 7644 triangles. As it can be seen in the distribution plotted below, a large majority of nodes have a clustering coefficient of 0. Thus, the resulting average clustering coefficient of the graph is relatively low : 0.019. This indicated that the nodes are not so well embedded in the network.



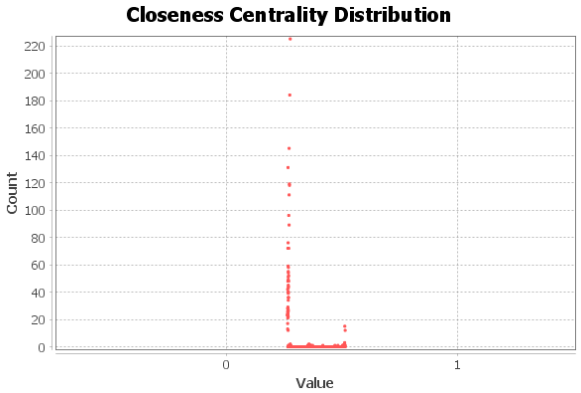
c. Network Diameter

The network has a diameter of 4; which corresponds to the longest graph distance between any two nodes, considering that the distance between two connected nodes is 1. The Average Path length is 3.672.

The Betweenness Centrality Distribution plotted below shows how often a node appears on shortest paths between nodes in the network



Then, the Closeness Centrality is the average distance from a given starting node to all other nodes in the network. It measures the Centrality of a node, meaning how close it is to other nodes.



Finally, the eccentricity is the distance from a given starting node to the farthest node in the network :