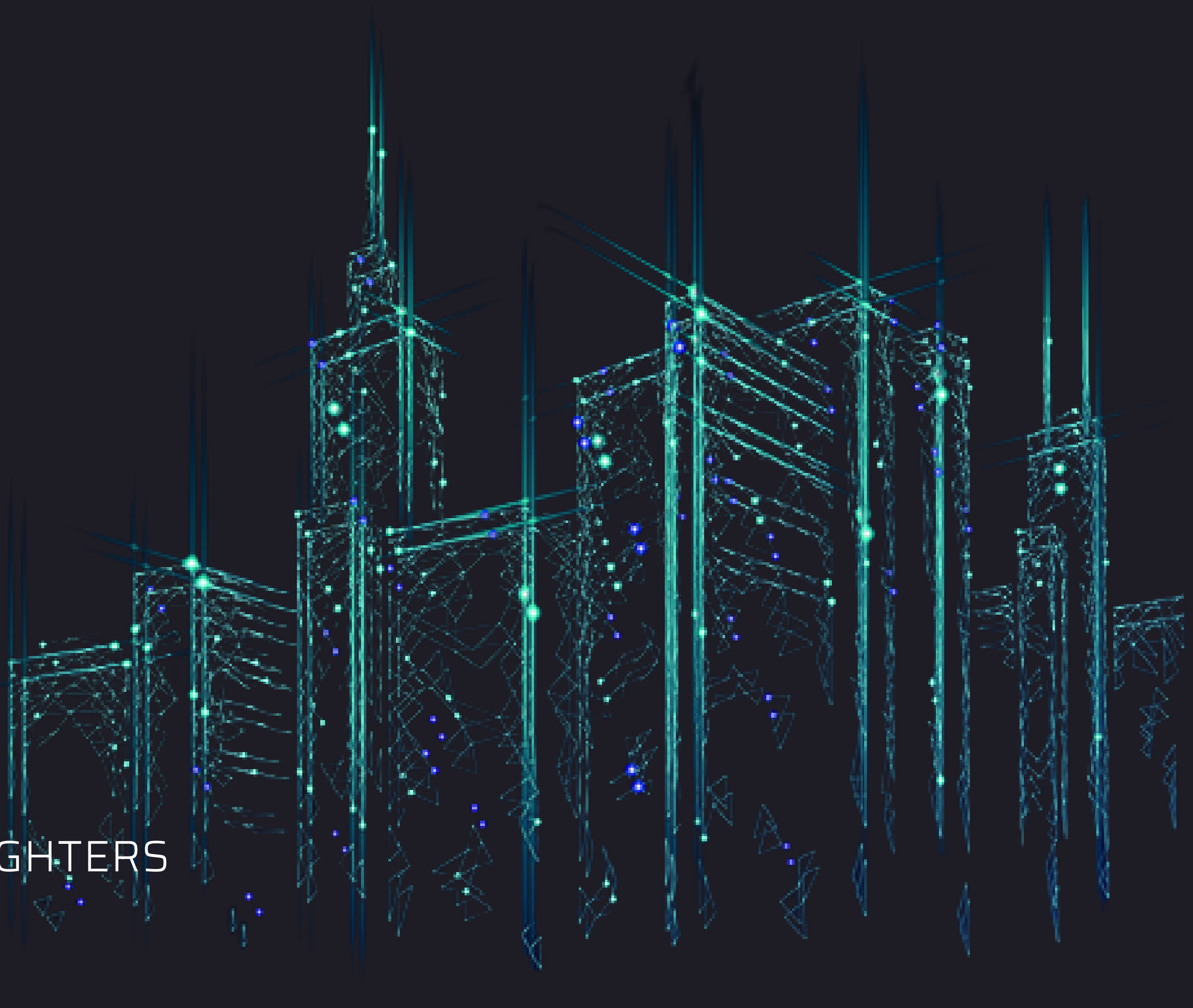
المدرسة الحسنية للأشغال العمومية
ECOLE HASSANIA DES TRAVAUX PUBLICS

SMART CITIES HACKATHON 2023

CITYGUARDIAN

Intelligent City Patrol Management



Made by

CRIMEFIGHTERS

ABSTRACT

In Casablanca, the economic capital, much like in other major metropolises, delinquency is increasing in proportion to demographic growth, a trend observed for decades. Some areas in Casablanca are currently recognized as high-risk zones, marked by a significant rise in assaults and thefts. Our City Guardian project is based on the intelligent distribution of police vans to these areas, with the aim of intervening rapidly and even preventing criminal acts, especially theft, violence, and various forms of aggression.

Keywords

AI, Optimal Path, ESP, GPS Module, Github, ArcGIS, API, Coordinates, Arduinu IDE, Mapping, Python, JavaScript.

THANKS

We would like to express our heartfelt gratitude to Geomatic for providing us with the opportunity to participate in the hackathon celebrating GIS Day. We also extend our thanks to our school, Hassania School of Public Works, for organizing this event centered around the theme of Smart City. Our sincere appreciation goes to all our professors whose guidance and support played a crucial role in our academic development. Their expertise and encouragement have been instrumental in our achievements, and we are truly grateful for the knowledge and skills gained through their teachings.

CONTENTS

LIST OF FIGURES

INTRODUCTION

Project Context

The origin of our project, ***CityGuardian***, an intelligent patrol van distribution platform, stems from an experience lived by one of our team members in Casablanca, in an area known for frequent assaults and thefts. While he was with his friends, criminals suddenly appeared to disrupt the safety of citizens. Fortunately, a police van happened to pass by at that critical moment. The arrival of this police van had an immediate deterrent effect, forcing the criminals to disperse and flee. This fortunate situation deeply inspired us to enhance security by introducing artificial intelligence for a more efficient distribution of patrol vans. Our goal is to optimize law enforcement interventions and prevent crimes through a more effective and technologically advanced approach.

Problem Statement

How can we effectively integrate artificial intelligence and Geographic Information System (GIS) tools to significantly enhance the efficiency of police interventions in cities, especially in high-risk areas, in order to advance towards the concept of a smart city and bolster urban security ?

Objectives

- Creating an autonomous system that dispatches police vans on patrols throughout the city's territory to cover areas where crimes are likely to occur.
- Developing an AI model that utilizes historical crime data within the city to predict future criminal activities.
- Facilitating swift police intervention during emergency calls by calculating the shortest path to the nearest patrolling police van.
- Integrating all of the aforementioned components into a cohesive system aimed at reinforcing public safety in the city.

Target Audience

- Law enforcement officers and operational leaders actively using the platform.
- Municipal authorities overseeing public safety and security.
- Emergency services teams benefiting from enhanced coordination during critical situations.
- Residents in high-risk urban areas experiencing improved security.
- Local businesses seeking a safer environment for employees and customers.
- Technology professionals interested in innovative security solutions.
- Government agencies responsible for public security and urban policy implementation.

NEEDS ANALYSIS

Functional Requirements

1. Integration of Artificial Intelligence (AI):
- Utilization of AI algorithms to predict high-risk areas and recommend patrol strategies.
- Predictive analysis of criminal trends for proactive intervention.

2. Optimized Patrol Management:
- Intelligent allocation system for patrols based on real-time criminal data analysis.
- Ability to dynamically adjust patrol routes based on emerging crime patterns.

3. Mapping System:
- Integration of Geographic Information System (GIS) for accurate mapping of criminal incidents.
- Data visualization features for an in-depth understanding of crime patterns.

4. User-Friendly Interface:
- Intuitive interface for police officers, facilitating efficient use of the platform.
- Interactive dashboards providing real-time information on patrols and incidents.

5. Real-Time Communication:
- Instant communication features between officers and the command center.
- Real-time notifications for critical incidents requiring immediate intervention.

Non-Functional Requirements

1. **ArcGIS Products:**
   - Needed for Spatial Analysis, Data Visualisation, & hosting an online server for communication between different elements of the system.

2. **Programming Languages:**
   - Using Python with ArcGIS libraries to automate various spatial functionalities..
   - Python is the chosen language for implementing predictive AI models.

3. **AI Model:**
   - Utilizing AI models to predict future crimes based on historical crime data.

4. **Important Data :**
   - Obtaining internet data on historical crimes, including crucial details such as the date and time of the incident, crime type, location of the crime (latitude and longitude in WGS84), and additional pertinent information.

CONCEPTION

The main objective of ***CityGuardian*** is to establish an intelligent distribution of van patrols throughout the city. This initiative aims to optimize law enforcement presence using strategies based on artificial intelligence and data analysis. To achieve our goals, we followed a systematic process.

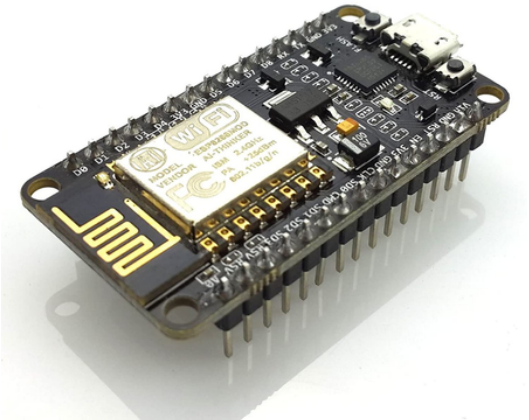
1.Data Collection :

Unfortunately, the lack of open-source data on crime rates in Casablanca led us to explore alternative sources. Hence, we chose to leverage Chicago's data to enrich our dataset. This decision was driven by the need to ensure quality and diversity suitable for developing robust predictive models, specifically tailored to our intervention context. Chicago's data proved to be highly beneficial, providing not only remarkable quality but also content rich in relevant indices for our analysis. Importantly, this Chicago data is open source, and you can find it here : https://data.cityofchicago.org/widgets/9hwr-2zxp?mobile_redirect=true.

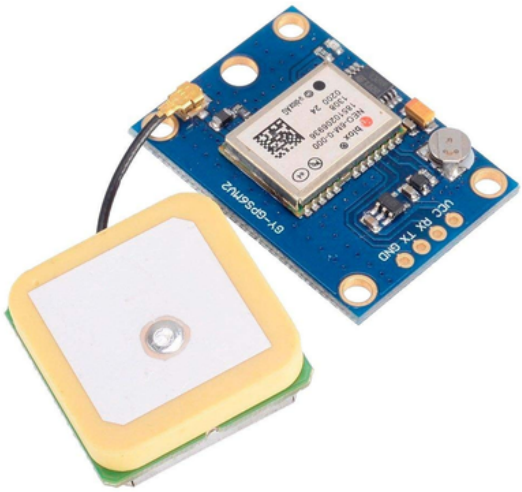
We would also need data about the roads of chicago if we want to implement algorithms about optimal paths. here's the source of chicago's roads data : https://data.cityofchicago.org/Transportation/Street-Center-Lines/6imu-meau

Furthermore, our system must ensure rapid police intervention in response to emergency calls. To simulate this scenario, we'll incorporate a GPS module along with an ESP8266 to transmit the caller's coordinates.

- GPS module : is a device that receives signals from satellites to determine its geographical position, allowing it to calculate latitude, longitude, and sometimes altitude.
- ESP8266 : is a low-cost Wi-Fi microchip developed by Espressif Systems. It's commonly used for connecting devices to the internet and enabling communication between them.



ESP8266



GPS module

2.Predictive Model :

The stage of predictive modeling through machine learning, employing a scalable approach, involves the use of a model capable of handling a variety of inputs and data sources. This approach ensures that the model can receive any data source and operate reliably, consistently achieving its objectives.

The predictive modeling step is carried out using Machine Learning algorithms :

- <u>Linear Regression</u>: aims to establish a linear relationship between a dependent variable and one or more independent variables.
- <u>Random Forest Regression</u>: uses an ensemble of decision trees to enhance the accuracy and stability of predictions.
- <u>Polynomial Regression</u>: extends linear regression by introducing polynomial terms to better fit non-linear relationships between variables.

2.Esri Products :

The utilization of Esri Products stands as an essential component in our process. Initially, it facilitates a visual representation of our data, enabling us to observe and analyze transformations seamlessly. This visual insight is pivotal in the testing of diverse prediction models, aiding in the identification of the most suitable ones for our purposes.

Moreover, these products offer a range of advanced spatial algorithms, such as Network Analysis, which provides valuable functionalities. For instance, it transforms road networks into graphs and nodes, enabling us to determine the shortest paths effectively.

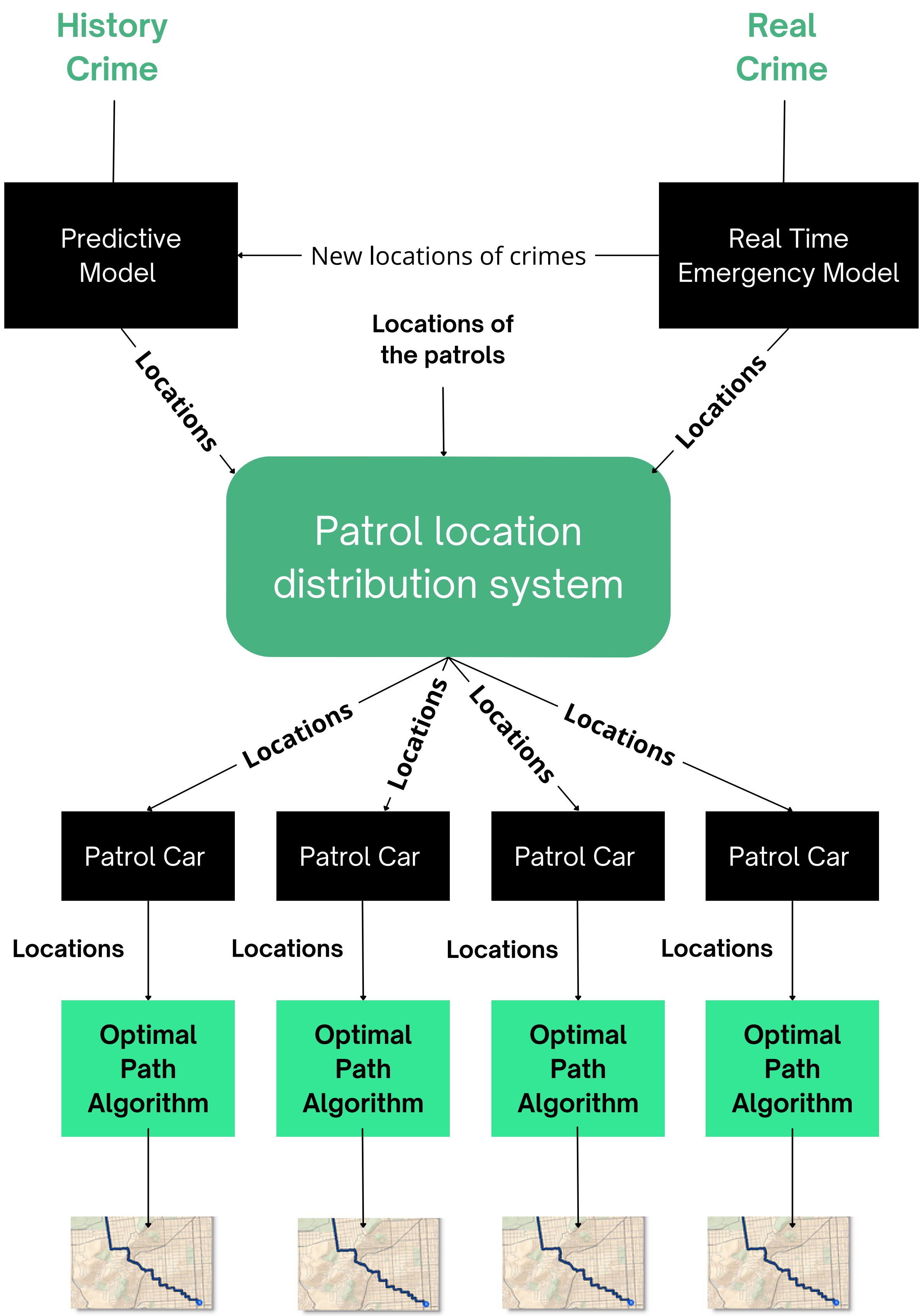
Furthermore, our licensing privileges have granted us access to ArcGIS servers, allowing seamless data uploads and fostering communication with our web application. This, in turn, empowers our web application to harness ArcGIS functionalities using the JavaScript API.

4.Web development :

Carried out using the ArcGIS for JavaScript API for web processing on the platform. This ensured the creation of an interactive and responsive user interface. Additionally, we integrated the ArcGIS Server API to establish seamless communication between the platform and the ArcGIS server, facilitating the exchange of essential data.

Given the extensive coding involved in our project, this report showcases only select parts of the code. The full implementation is available on our GitHub repository at this link: https://github.com/oussamakazoubi/CityGuardian.
All code references mentioned herein can be found in the repository

Here is the diagram representing the conception and implementation stages of our ***CityGuardian*** project :



Project Model Diagram

DEVELOPMENT

1. Data Aquisition:

a - CSV file:

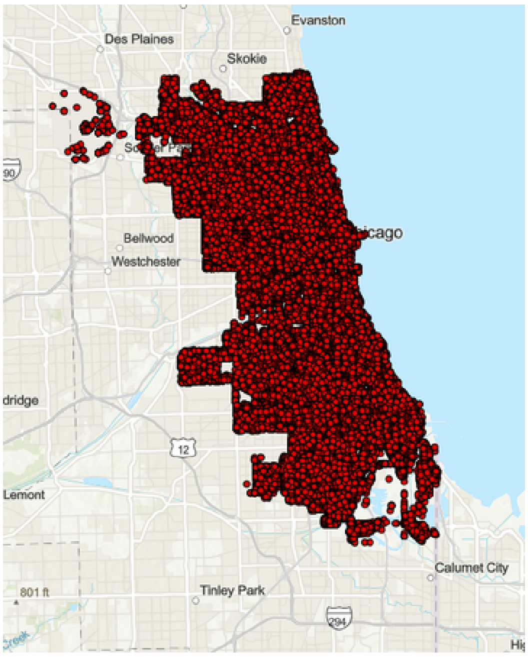
We obtained detailed crime data for Chicago from a government website: https://data.cityofchicago.org/widgets/9hwr-2zxp?mobile_redirect=true. The data was in the form of a CSV file named "Crimes_-_2022.csv." To visualize these incidents in ArcMap, we used the "Create Feature Class from XY Table" feature.

The table schema includes various fields which are :

ID, Case Number, Date, Block, IUCR, Primary Type, Description, Location Description, Arrest, Domestic, Beat, District, Ward, Community Area, FBI Code, X Coordinate, Y Coordinate, Year, Updated On, Latitude, Longitude, and Location.

F	G	H	I	J	K	L	M
Primary Type	Description	Location Desc	Arrest	Domestic	Beat	District	Ward
SEX OFFENSE	INDECENT SO	RESIDENCE	false	true	423	4	
OTHER OFFEN	HARASSMENT	RESIDENCE	false	true	724	7	
SEX OFFENSE	SEXUAL EXPL(	APARTMENT	true	false	324	3	
SEX OFFENSE	AGGRAVATED	RESIDENCE	false	false	1621	16	
WEAPONS VI(	RECKLESS FIR	STREET	false	false	733	7	
SEX OFFENSE	AGGRAVATED	RESIDENCE	false	true	423	4	
THEFT	OVER $500	COMMERCIAI	false	false	113	1	
BATTERY	DOMESTIC BA	HOSPITAL BU	true	true	1211	12	
BATTERY	DOMESTIC BA	RESIDENCE	true	true	2431	24	
OTHER OFFEN	VIOLATE ORD	WAREHOUSE	true	true	1822	18	
BATTERY	DOMESTIC BA	VEHICLE - CO	true	true	1834	18	
THEFT	OVER $500	APARTMENT	true	true	1914	19	
BATTERY	AGGRAVATED	SCHOOL - PUI	false	false	613	6	
CRIMINAL SE)	AGGRAVATED	RESIDENCE	true	true	813	8	
ROBBERY	STRONG ARM	APARTMENT	false	false	1411	14	
ROBBERY	AGGRAVATED	STREET	false	false	113	1	
BURGLARY	UNLAWFUL E	WAREHOUSE	true	false	1111	11	
ROBBERY	VEHICULAR H	STREET	true	false	1011	10	
SEX OFFENSE	CRIMINAL SE)	OTHER (SPECI	false	false	1023	10	
OTHER OFFEN	OTHER CRIME	GOVERNMEN	false	false	1231	12	
ARSON	BY FIRE	RESIDENCE - (	true	false	835	8	
THEFT	POCKET-PICKI	SPORTS AREN	false	false	132	1	
THEFT	$500 AND UN	STREET	false	false	713	7	
BATTERY	DOMESTIC BA	APARTMENT	true	true	131	1	
BATTERY	DOMESTIC BA	APARTMENT	true	true	2424	24	
BATTERY	DOMESTIC BA	APARTMENT	true	true	1224	12	
BATTERY	DOMESTIC BA	APARTMENT	true	true	1935	19	
BATTERY	DOMESTIC BA	APARTMENT	true	true	1924	19	
OFFENSE INV(	AGGRAVATED	RESIDENCE	false	true	1623	16	

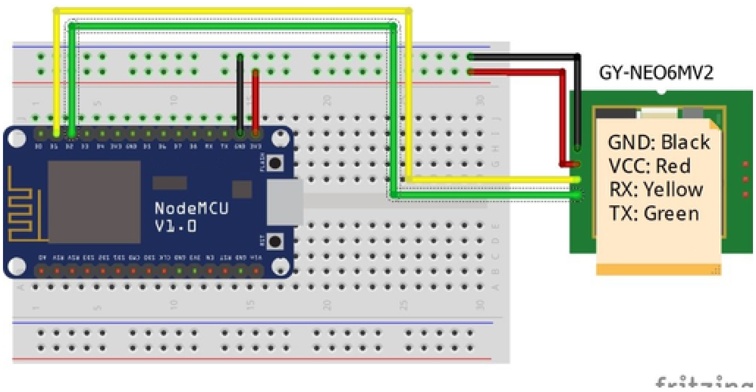
Chicago Crimes 2022 CSV file



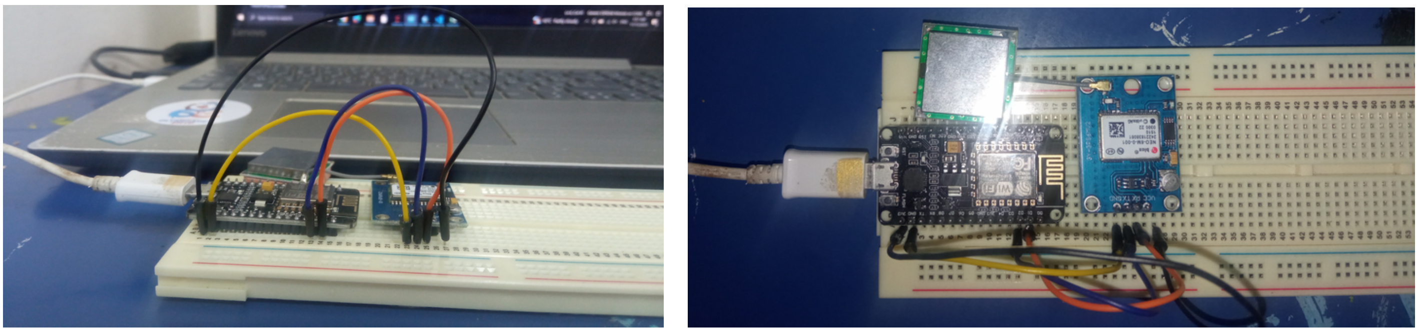
Chicago Crimes Shapefile

b - GPS Coordinates :

We plan to utilize a GPS module in conjunction with the ESP8266 microcontroller to simulate tracking the location of an individual in distress who is reaching out to the police for help. Furthermore, our mobile phones can serve a similar purpose, given their GPS capabilities.

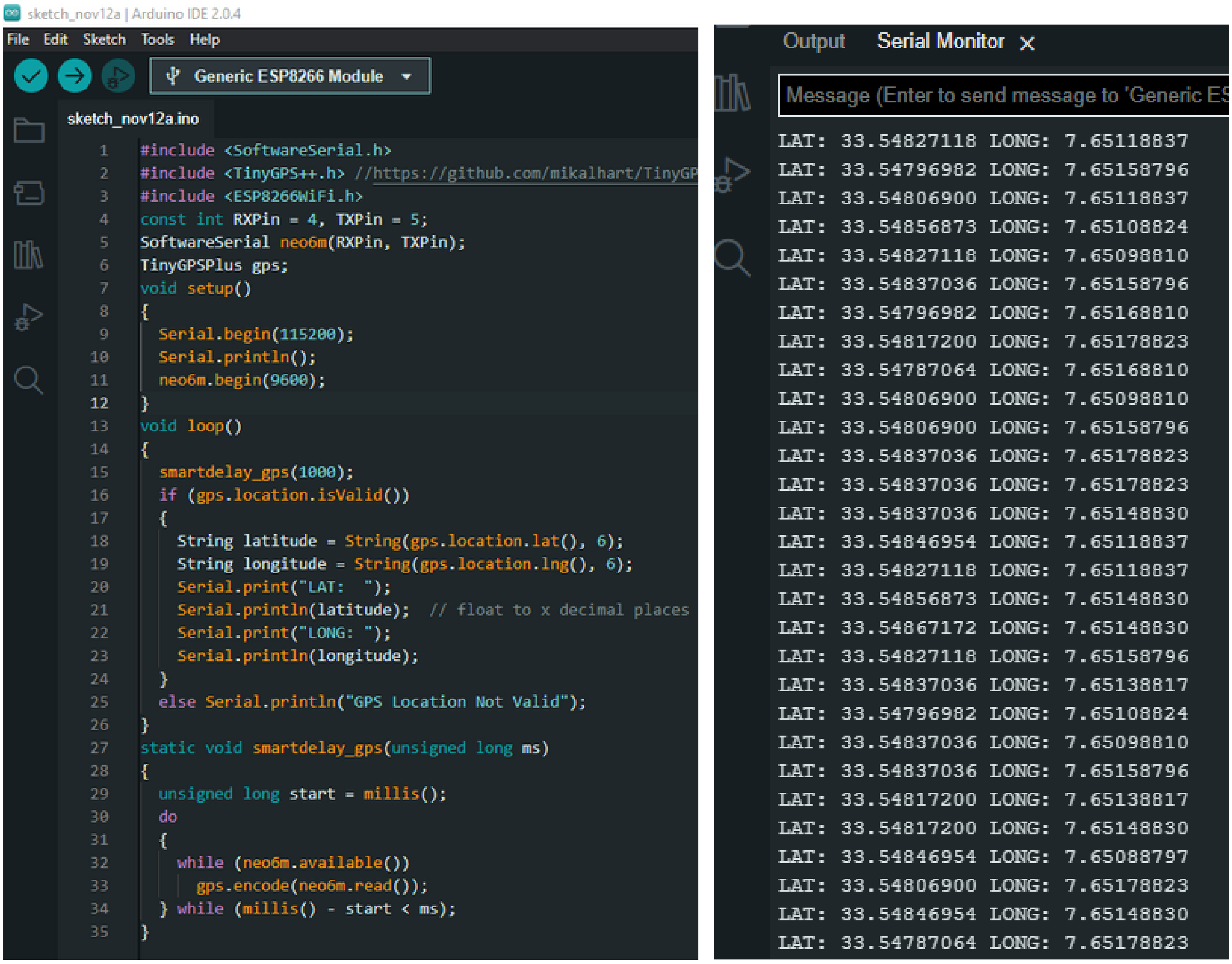


GPS Module / ESP circuit diagram



GPS/ESP circuit assembly

We plan to utilize a GPS module in conjunction with the ESP8266 microcontroller to simulate tracking the location of an individual in distress who is reaching out to the police for help. Furthermore, our mobile phones can serve a similar purpose, given their GPS capabilities.



ESP code and resulting coordinates

Additionally, by updating the ESP code, we enable this electrical device to directly transmit its coordinates to our ArcGIS server, allowing seamless access by our applications. The updated code is available on our GitHub page

2. Predictive AI Model:

three important phases are premordial, This initial phase involves the acquisition of relevant data to fuel our system, luckily we already have enough of it to teach our model (the CSV file).

Phase 2 - Data Processing:

Data processing is a crucial stage that is divided into several sub-steps:

- Cleaning: Removal of outliers, handling missing data, and correction of errors to ensure data quality.
- Normalization: Standardization of units and scales to ensure consistent comparability.
- Encoding: Conversion of categorical data into numerical formats for optimal integration into models.
-

Phase 3 - Data Splitting:

This stage separates the data into three distinct sets:

- Training: Utilized to train the model.
- Validation: Employed to fine-tune model parameters and prevent overfitting.
- Test: Reserved for evaluating model performance on unseen data.
-

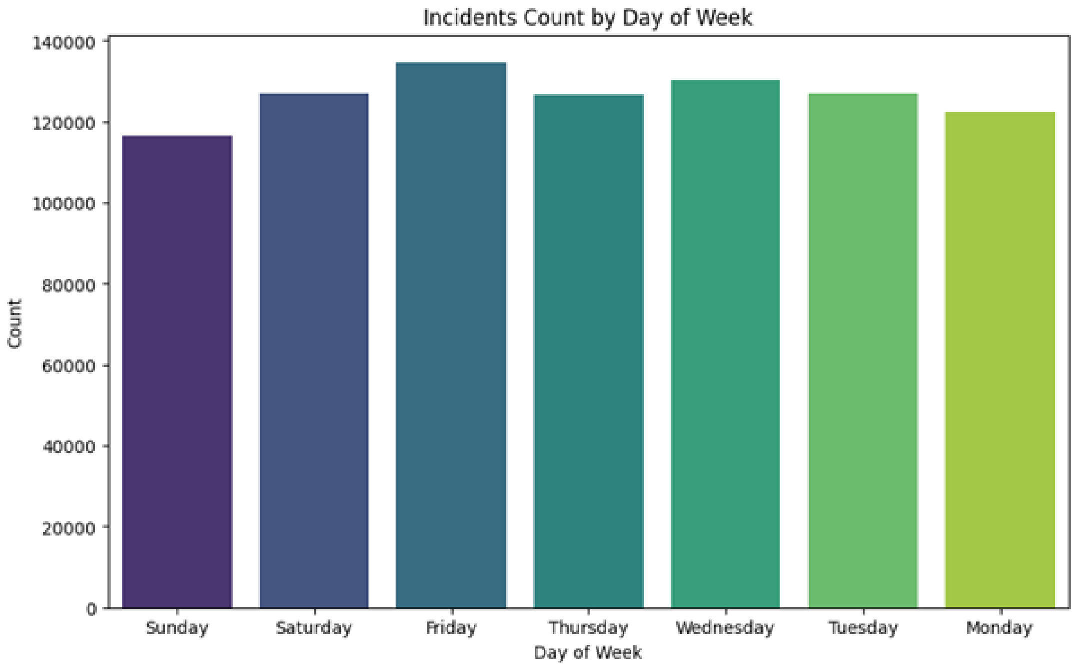
Phae 4 - Data Visualization:

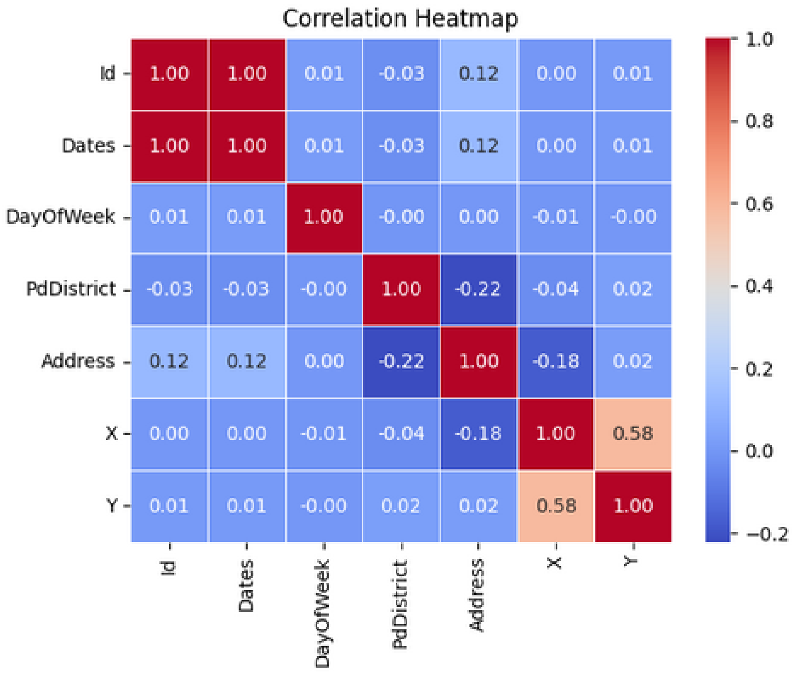
Effective data visualization is crucial for understanding data distribution and spotting trends. This includes:

- Exploratory Data Analysis (EDA): A thorough exploration of data to reveal patterns (distribution) and relationships (correlation).
- Visualization: Using graphs and charts to make information easily understandable.

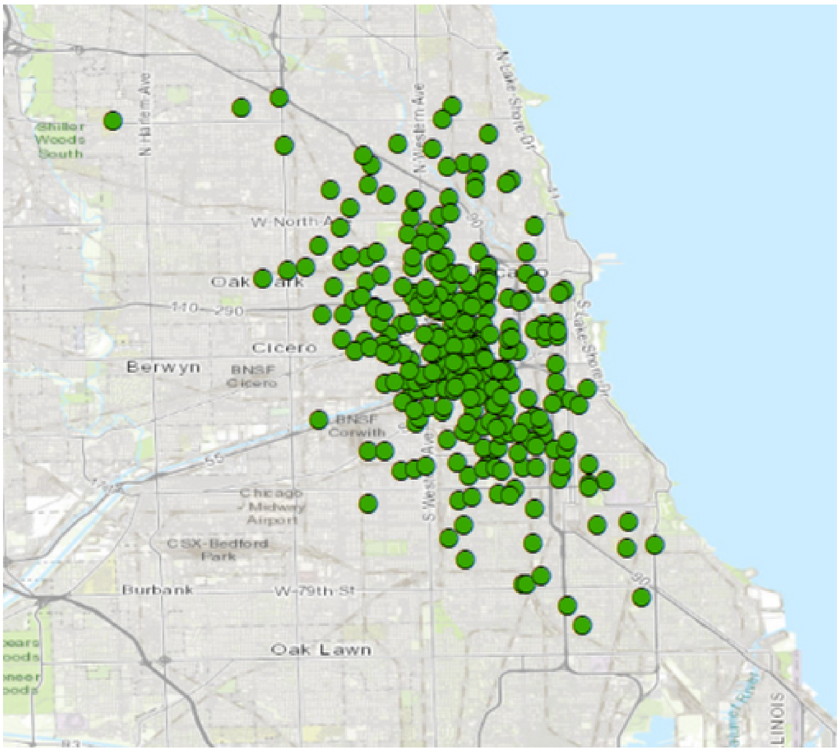
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from datetime import datetime

# existing data loading and preprocessing code
data_san = pd.read_excel(r'C:\Users\Oussama\Desktop\datachicago\mini chicago.xlsx')
data_san['Date'] = pd.to_datetime(data_san['Date'])
data_san['Hour'] = data_san['Date'].dt.hour
data_san['Month'] = data_san['Date'].dt.month
data_san['Day'] = data_san['Date'].dt.day
data_san['Minute'] = data_san['Date'].dt.minute

# clean Null data
data_san = data_san.drop(['Date'], axis=1)
data_san = data_san.dropna()

# Split into features and targets
F = data_san.drop(['X', 'Y'], axis=1)
T = data_san[['X', 'Y']]

# Model training
F_train, F_test, T_train, T_test = train_test_split(F, T, test_size=1, random_state=42)
model_R = RandomForestRegressor(n_estimators=100)
model_R.fit(F_train, T_train)
```

Predictive AI Model (Predict_Path.py).





To achieve clarity, the primary purpose of our AI model is to predict crime locations at a given time [t → (x, y)]. Our choice of the RandomForestRegressor wasn't arbitrary. Initially, we used Linear Regression, which erroneously associated time and place in a linear manner, linking the change in time proportionately to the change in location. Subsequently, we attempted polynomial regression, increasing the degree each time, but the continuous nature of the output function resulted in predicted points being too close together for small variations in time, which didn't align with the characteristics of Chicago's crime data. After multiple calculations, the RandomForest Regressor emerged as the most suitable model. Our main objective is to generate numerous predicted crime locations within a small variation of time and have police vans patrol these areas to ensure public safety. In the following section, we'll explore how to optimally dispatch these vans.



Predicted Crimes in Chicago around 5pm

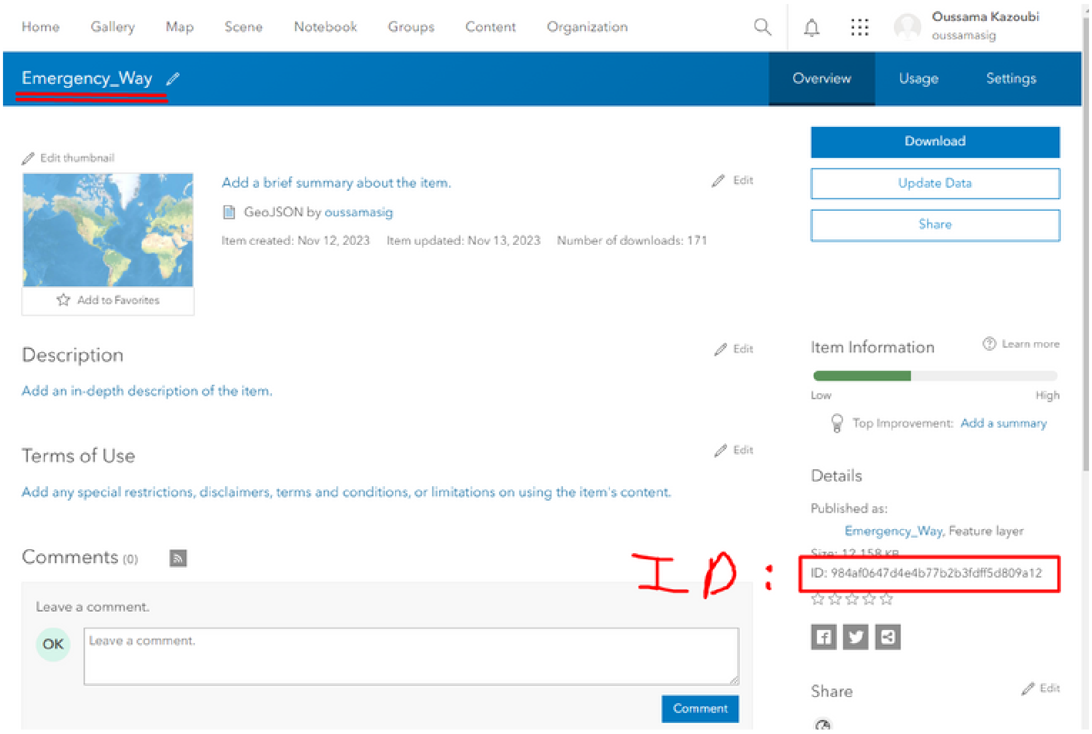
3. Patrol Paths distributor

a - ArcGIS Server:

For this section and the ongoing project, our approach involves developing a web application to monitor real-time police dispatches and patrols, simulating emergency interventions. The initial phase involves obtaining predicted crimes, calculated independently by the web hosting server (localhost). We've selected the ArcGIS server as a bridge between the localhost and the machine performing geospatial data processing. Once a process is executed by the machine, it transmits the results to the ArcGIS server. Subsequently, our web app retrieves the required data for visualization or additional treatments.

```
gis = GIS("https://www.arcgis.com", "oussamasig", "[redacted]) Password
predicted_coords = multiple_predictions(hour=8,minutes=5, day=5, month=5, nbr=300)
item_id = "984af0647d4e4b77b2b3fdff5d809a12"
update_json_with_predictions(predicted_coords, item_id)
```

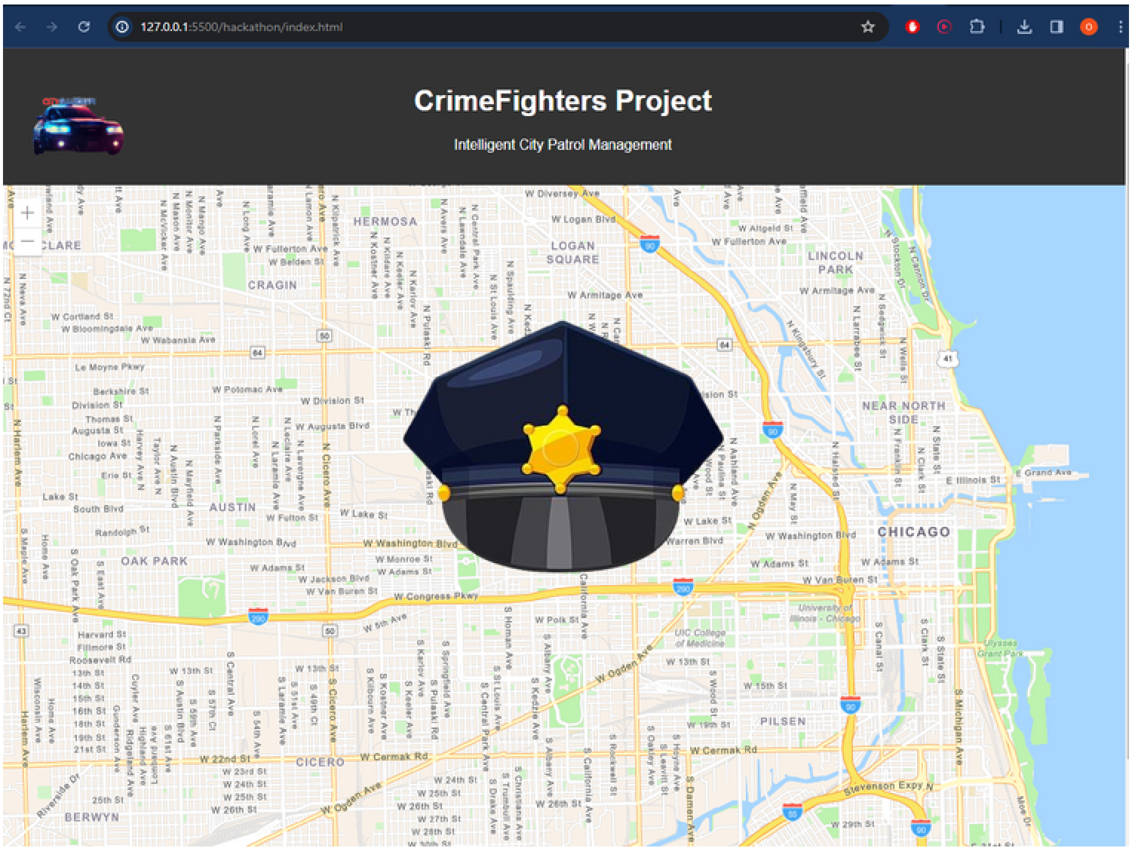
updating a GeoJson file in the arcgis server with predicted crimes data using its id



Viewing The Predicted data in Arcgis Server Interface

b - ArcGIS JavaScript API:

For this section and the ongoing project, our approach involves developing a web application to monitor real-time police dispatches and patrols, simulating emergency interventions. The initial phase involves obtaining predicted crimes, calculated independently by the web hosting server (localhost). We've selected the ArcGIS server as a bridge between the localhost and the machine performing geospatial data processing. Once a process is executed by the machine, it transmits the results to the ArcGIS server. Subsequently, our web app retrieves the required data for visualization or additional treatments. To Achieve that we must use ArcGIS API for JavaScript , to create arcgis objects , operate on them or use ArcGIS functions.



The Front Page of our Web App (index.html)

```
// Create a graphics layer for the route
const routeLayer = new GraphicsLayer();
view.map.add(routeLayer);

// Create a graphics layer for the police cars
const policeCarsLayer = new GraphicsLayer();
view.map.add(policeCarsLayer);

// Symbol for the police car
const carIconUrl = "car.png";
const carSymbol = new PictureMarkerSymbol({
    url: carIconUrl,
    width: "32px",
    height: "32px"
});

// Function to create a route between multiple points
function createOptimalPath(points, color) {
// Convert array of arrays to an array of Point geometries
const pointGeometries = points.map(point => new Point({ x: point[0], y: point[1] }));

// Log the points to the console

// Define the route task
const routeTask = new RouteTask({
    url: "https://route.arcgis.com/arcgis/rest/services/World/Route/NAServer/Route_World",
    apiKey: apiKey // Include the API key for authentication
});
```

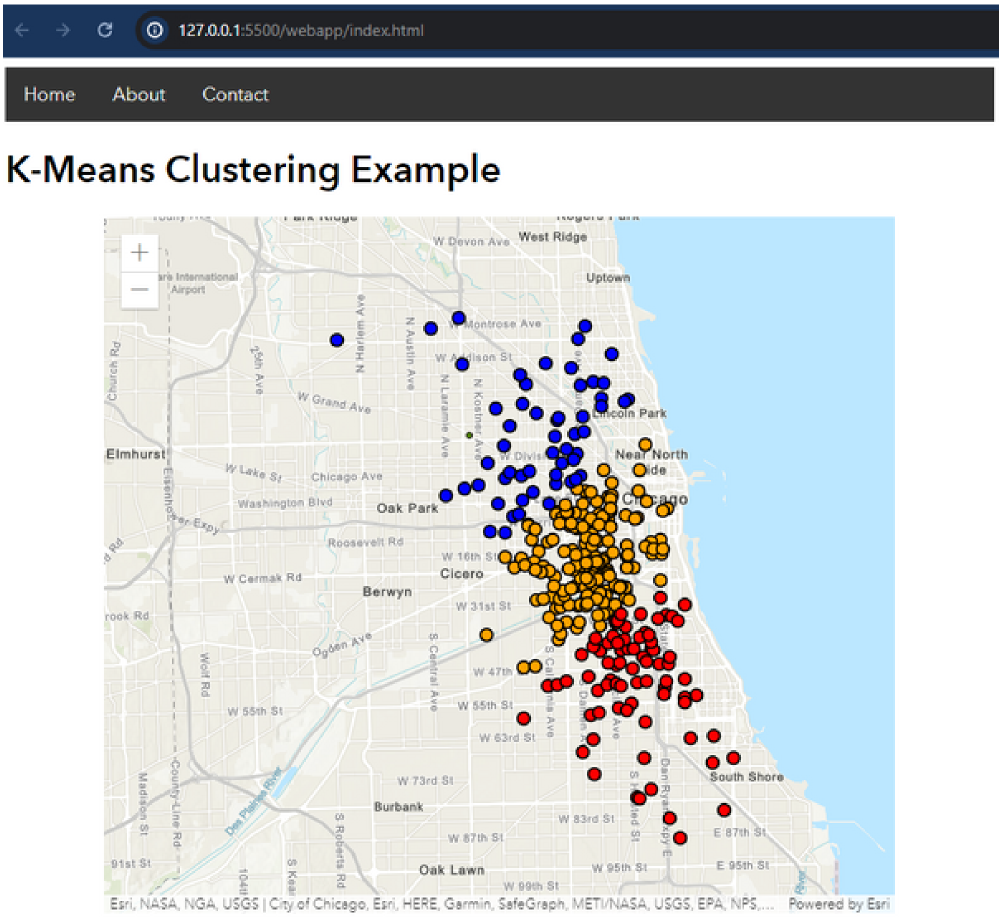
Example for ArcGIS JS API Usage (all_cars.html)

c - Intelligent Points Distribution:

The subsequent stage involves assigning a set of points (predicted crime coordinates) to each police van and determining the most efficient route that covers these points. For the point assignment, we'll utilize an AI clustering algorithm known as "K-Means."

```
    const k = 4; // Number of clusters

    const clusters = kmeans(a, k, distances.euclidean, 10, function(snapshot) {
        // This callback function will be called every 10 iterations (adjust as needed)

    });

// Create points for the first car
const splitIndex = Math.floor(a.length / 2);
const listPointsCar1 = a.slice(0, splitIndex).map(point => new Point({ longitude: point[0], latitude: point[1] }));

let p=[[0,0,255],[0,255,0],[255,0,0],[100,0,0],[0,100,0],[255,100,0],[0,100,255],[50,50,50],[50,10,80],[80,30,10],[0,150,20],[0,8,200],[],[],[],[]];
// Create points for the second car
const listPointsCar2 = a.slice(splitIndex).map(point => new Point({ longitude: point[0], latitude: point[1] }));
```

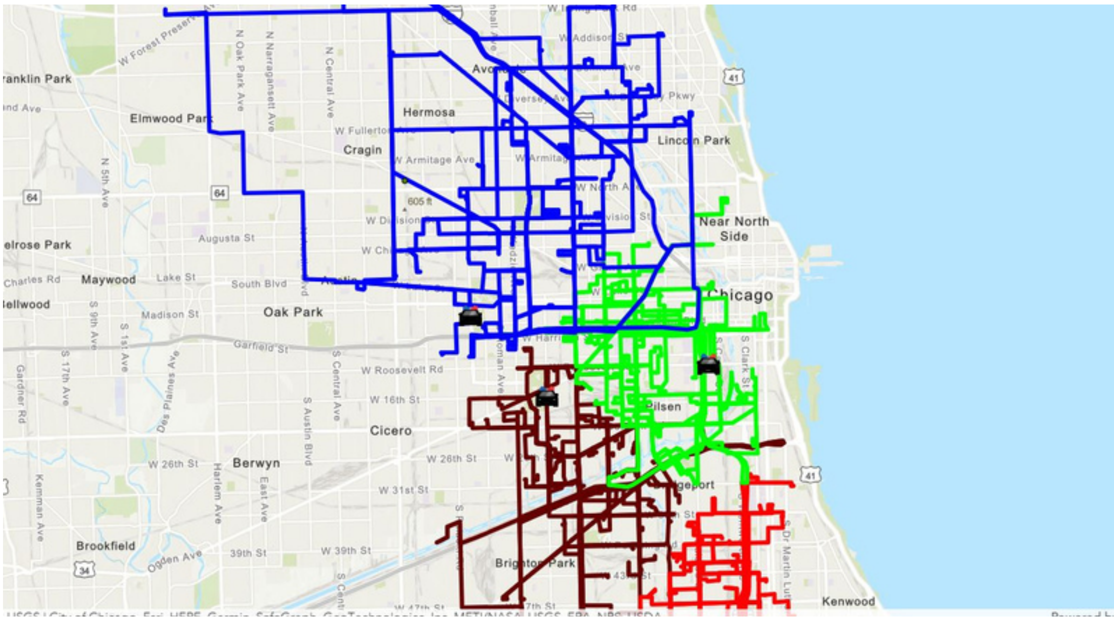
Creating Points distributions for 3 police vans (all_cars.html)



Distribution Result (for 3 Vans)

d - Optimal Patrol Path:

Now, our goal is to efficiently guide the police vans through their designated points. To achieve this, we'll craft a custom function called 'createOptimalPath' leveraging the functionalities available within the ArcGIS JS API. To enhance visualization, we'll scale up the number of points and the sets of points, corresponding to the increased number of police vans, ensuring a more comprehensive representation.



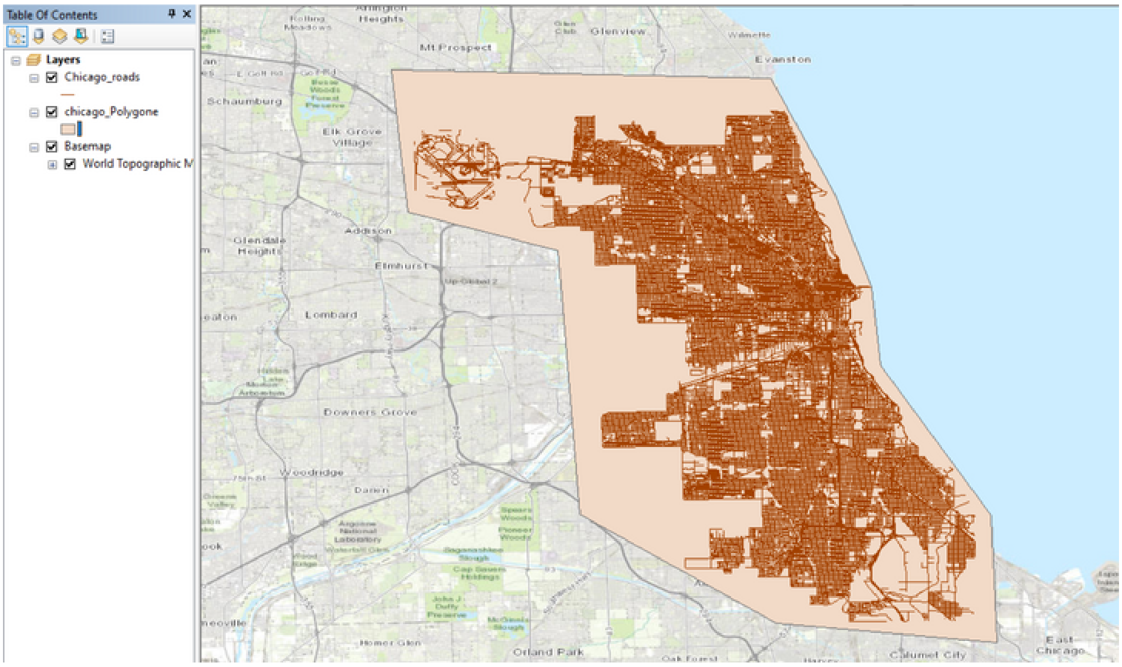
Police Vans Path Assignements

In this illustration, there are four distinct colored paths, each assigned to a police van (represented by the police icon on the map). On the webpage, real-time movement of the police icons is observable as they navigate through these paths, signifying their coverage of the designated locations. Further details and visual representation of the police vans' movements can be seen in the accompanying video provided with this report.

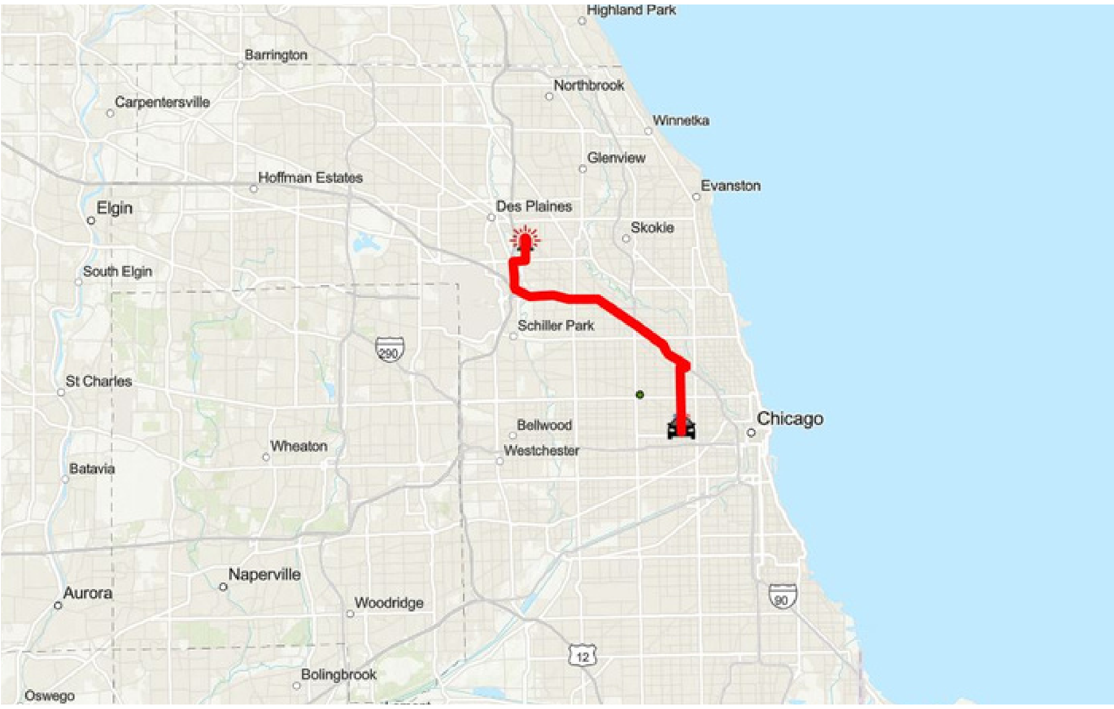
Performing calculations of such magnitude directly within the localhost JS instructions may occasionally slow down the web application's operations. In such scenarios, an alternative approach involves performing calculations on another device utilizing ArcGIS Python libraries and then uploading the results to the ArcGIS server for retrieval by the web application. For determining the optimal path through the predicted crime points, we employed Network Analysis functions to transform Chicago roads into a network of nodes and used ArcGIS's built-in functions to derive the most efficient route. The reason we're using two devices instead of one device that do the calculationsand then directly upload it to the web application, is that we are working as a group and so each one of us is developping some solutions so the only way is communication via a network and we chose ArcGIS server.

4. Emergency Intervention

In times of emergency—whether due to a citizen's call for assistance or a declared urgent police intervention—we rely on our web page to swiftly determine the shortest path to the location. However, when aiming to use the GPS module to simulate an emergency call, it presented a challenge. The Chicago-centric data and road network rendered the GPS coordinates invalid. To simulate the emergency call or police dispatch, we took an alternative route. We delineated Chicago using a polygon in ArcMap and created a Python script to generate a random point within this boundary. This random point was then transmitted to the ArcGIS server, visualized in our web application, and the shortest path was calculated in response.ough the predicted crime points, we employed Network Analysis functions to transform Chicago roads into a network of nodes and used ArcGIS's built-in functions to derive the most efficient route. The reason we're using two devices instead of one device that do the calculationsand then directly upload it to the web application, is that we are working as a group and so each one of us is developping some solutions so the only way is communication via a network and we chose ArcGIS server.



Chicago polygone



Shortest path towards the emergency location

CONCLUSION

This project represents an innovative step in the realm of public safety. By integrating AI, spatial analysis, and real-time response systems, it seeks to optimize law enforcement strategies. The implementation of predictive crime models, along with the rapid emergency response system using GPS technology, promises a more effective and efficient approach to address public safety concerns. Our Project can easily be implemented into Moroccan's cities, all that's needed is the crime's data and police vans with gps receptors and connected to the server that's hosting the web application. It was a really fun experience tackeling this problematic together and each one of our team is thinking and implementing solutions to bring our design to life. The hardest challenge during this competition was getting to learn new tools in a very small amount of time in order to finish the project in time, and we still had a lot of room to further improve our project. However, the time barrier is what made this competition unique and interesting and we are looking forward to seeing the other participants' projects and learning new things.

Team crimeFighters

BOUHAOUS Youssef
EL ABDIOUI Lamiae
KAZOUBI Oussama
MEGDEL Mohamed