Hazard-Free Navigation - (*Considering Safety and Efficiency*)

Spatial Group 20:-

Sr.No	Name	Student ID	Email ID
1	Aditya Reddy Borra	862393087	aborr011@ucr.edu
2	Harish Kumar Manepalli	862395420	hmane001@ucr.edu
3	Sree Charan Reddy Gangireddy	862327313	sgang011@ucr.edu
4	Srikar Kilambi	862395242	skila001@ucr.edu
5	Venkata Sai Kumar Gottumukkala	862393318	vgott001@ucr.edu

Project category : Spatial Application.

INTRODUCTION

1.1 Overview

Our proposal for the project is to find the safest and the shortest paths between two locations in a city, which go through the least number of crime affected/prevalent areas. We developed a user interface which will solve the problem to find the shortest path between any two locations in San Francisco, California, minimizing the crime areas analyzed from real-time crime dataset.

1.2 Problem Definition

Traditionally, route optimization prioritizes speed over safety, neglecting potential risks. This project focuses on leveraging data analysis to heighten public awareness and security. Its goal is to identify the safest route among available options, addressing prevalent security concerns in the city. While the shortest path may be the norm, minimizing travel through hazardous areas can be life-preserving. Users often prioritize personal safety over the shortest distance traveled.

As maps are commonly used for navigation and crime rates are escalating, ensuring people's safety becomes imperative. This initiative involves integrating safety data into map services, empowering users to factor safety considerations into their route decisions.

1.3 Project-related work

Shortest path algorithm:

- Using Shortest Path Algorithms to Improve Safety in Navigation- This paper proposes a modified Dijkstra's algorithm that takes into account crime data to find the safest path between two points.
- Short Path Algorithms for Large-Scale Road Networks- This paper evaluates the performance of different shortest path algorithms for large-scale road networks.
- Shortest path finding in GIS using node combination & Dijkstra Algorithm- This paper proposes a method for finding the shortest path in a GIS using node combination and Dijkstra's algorithm.

Spatial Crime Analysis:

- Geospatial Big Data Handling Theory and Methods: A Review and Research Challenges: This paper discusses the challenges and methods for handling geospatial big data, which is relevant to integrating OpenStreetMap (OSM) data into our application.
- Spatial And temporal analysis of Crime - This paper identifies hotspots by forming a grid with circles of set radius at each node, counting and ranking incidents within these circles, refining the process with hexagons, and selecting either the single node with the most incidents or all equally high-count nodes.
- Three Decades of GIS Application in Spatial Crime Analysis: Present Global Status and Emerging Trends: This paper reviews 30 years of using GIS in spatial crime analysis, covering its evolution in crime mapping, hot spot analysis, and other methods. It also highlights new trends like using machine learning and AI for crime prediction and prevention strategies.

Hotspot detection:

- Spatiotemporal analysis and hotspots detection of COVID-19 using geographic information system: This study applies spatiotemporal analysis and GIS to identify COVID-19 hotspots in March and April 2020, using techniques like Kernel Density Estimation and Spatial Autocorrelation (Moran's I).
- Comparative Study of Approaches for Detecting Crime Hotspots with Considering Concentration and Shape Characteristics :In this paper several approaches of hotspot detection were compared while simultaneously considering the concentration and shape characteristics.Out of the comparison they found an algorithm which worked the best.

1.4 Proposed Solution

Developed a web application hosted on Flask which takes user inputs as addresses for source & destination, and returns an interactive map with the safest route. A backend algorithm has been developed to compute this safest route using OSMnx and our own implementation using Dijkstra and A* algorithm to get the shortest path. We used NetworkX to compute the shortest path.We also wrote our own algorithm to do the same.

The process can be summarized as below:

A	Initially we loaded the dataset into the map to generate a heatmap of this crime data to visualize the crime density of the full dataset.
B	Generate top 200 Hotspots from this data using k-means clustering and euclidean distance parameter.
C	Used OSMnx module to download San Francisco City map as nodes and also converted latitude and longitude pairs of the hotspots to nodes
D	Request users for a source and destination and again convert those to nodes. Initially gave a route without considering any crime data
E	Removed the generated hotspots clusters from the map network and regenerated the shortest path again from the modified Graph.
F	Used NetworkX module for algorithm creation and modification on graphs. Developed Dijkstra, A* and NetworkX built-in algorithm to calculate the safest shortest path.
G	Integration of frontend & backend implementation on Flask Framework to generate visualizations on an interactive maps

ANALYSIS

2.1 Software Requirements

Framework : Flask
Front End : HTML, CSS, Javascript
Back End : Python
Libraries: OSMnx, Networkx, scikit-learn, pandas, pickle

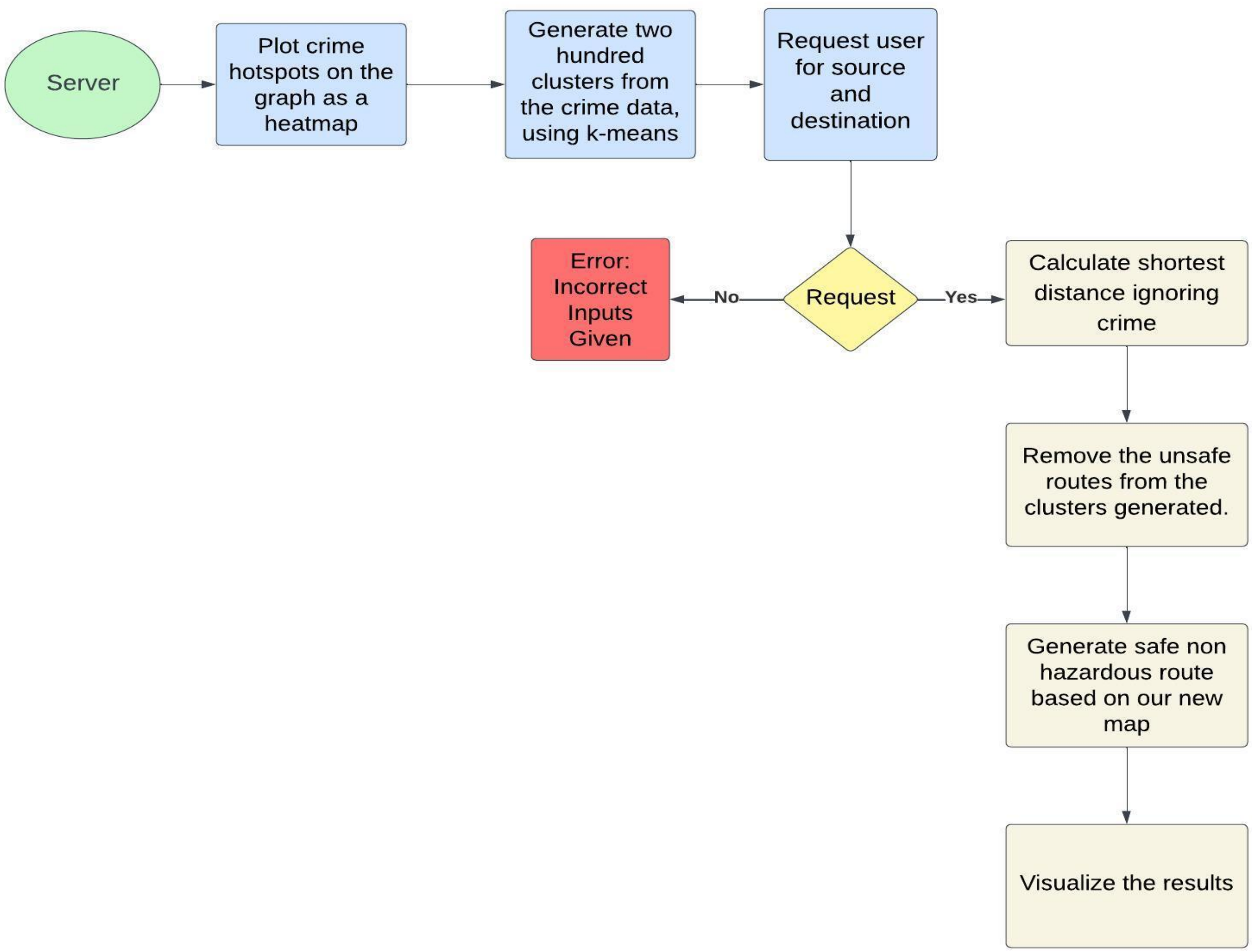
2.2 Use Cases

- **Crime Hotspots:** With crime rates on the rise, public safety, particularly for women, is a growing concern. It's crucial for travelers to be informed about the safest routes. We're employing algorithms to determine routes that are both safe and efficient, enhancing travel safety.
- **Traffic Congestion:** Offering alternative routes to bypass heavy traffic areas or suggesting optimal times to travel can minimize congestion-related accidents and delays.

- **Pollution Hotspots**: Air quality is a major concern in urban areas. We can use environmental data to identify areas with high pollution levels. Then we can suggest alternative routes for people with health concerns, like asthma, to avoid these pollution hotspots.
- **Wildlife Crossing Hotspots:** In areas where wildlife-vehicle collisions are frequent, we can use historical data to pinpoint these hotspots. We then provide alternative routes for drivers, reducing the risk of accidents and protecting wildlife.

DESIGN & IMPLEMENTATION

3.1 Design



3.2 Implementation

The application we developed for safe navigation in San Francisco utilizes a comprehensive approach by analyzing crime data to ensure hazard-free routes. Initially, the crime dataset is processed to create a heatmap, providing a visual representation of crime density. This data is further refined using k-means clustering to identify the top 200 crime hotspots based on Euclidean distance.

The OSMnx module is employed to obtain the city's graph network, converting geographic coordinates into nodes within this network. Users can input their start and end points, which are also converted into nodes. The application first calculates the shortest path without considering the crime data.

To enhance safety, the nodes representing crime hotspots are removed from the city's graph network. The application then recalculates the shortest path, ensuring it avoids areas with high crime rates. The NetworkX module supports this process, with algorithms such as Dijkstra and A* being adapted to find the safest and shortest path.

Finally, the Flask Framework bridges the backend and frontend, providing users with interactive maps to visualize their safe routes. This integration showcases the application's utility in real-world navigation, prioritizing user safety through informed route optimization.

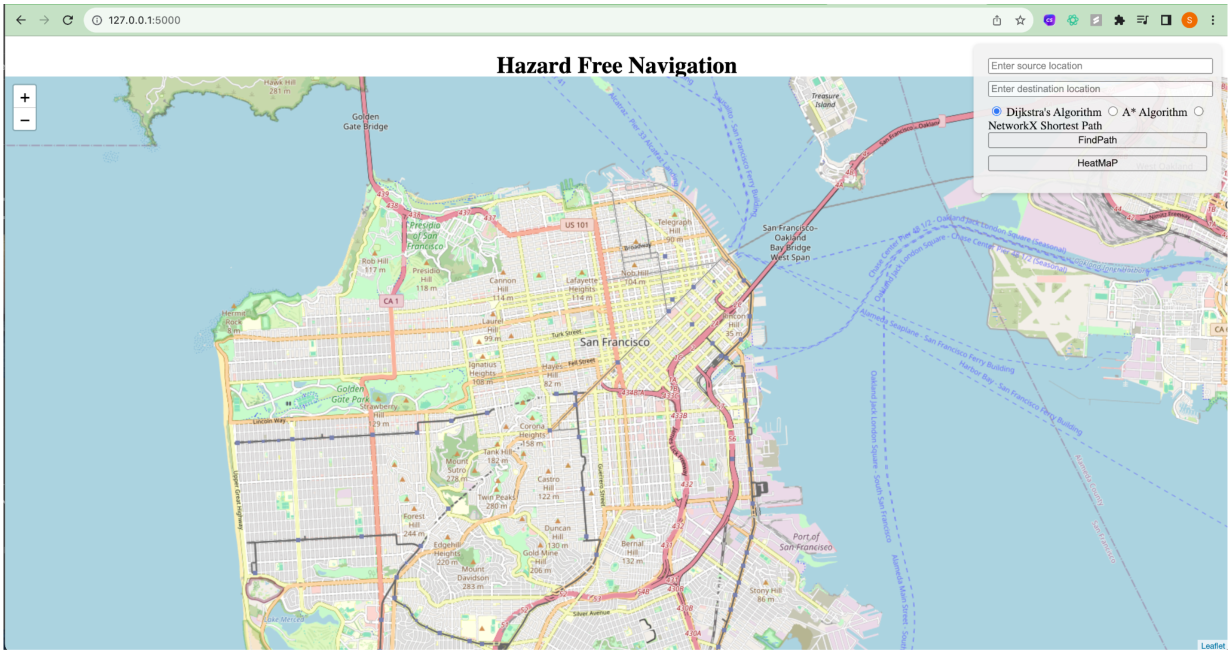
3.3 Technology Selection

- **NetworkX**: We have used NetworkX to find the shortest path from source to destination. Modifying the graph to compute the best path based on hotspots.
- **OSMnx**: Used for both generating the graphical representation of San Francisco and converting text addresses to geographic coordinates.
- **Folium**: To present the shortest route on the map (UI) using the Django framework.
- **Leaflet:** Enables us to plot the paths and hotspots on the map, providing an intuitive visual guide.
- **Pickle:** Employed for loading the pre-processed graph and hotspots data.

3.4 Working of Algorithm

Frontend & Backend:

In our application's front-end, we start by collecting the source and destination addresses from users through a user-friendly interface designed with HTML and CSS. Once these addresses are inputted, they are sent to the backend for processing.



In the backend, these addresses are transformed into geographic coordinates using **OSMnx geocode**. We then deploy our backend algorithms to determine a route. This process involves returning a list of nodal points representing the safest and shortest path, as well as identifying the location of hotspots.

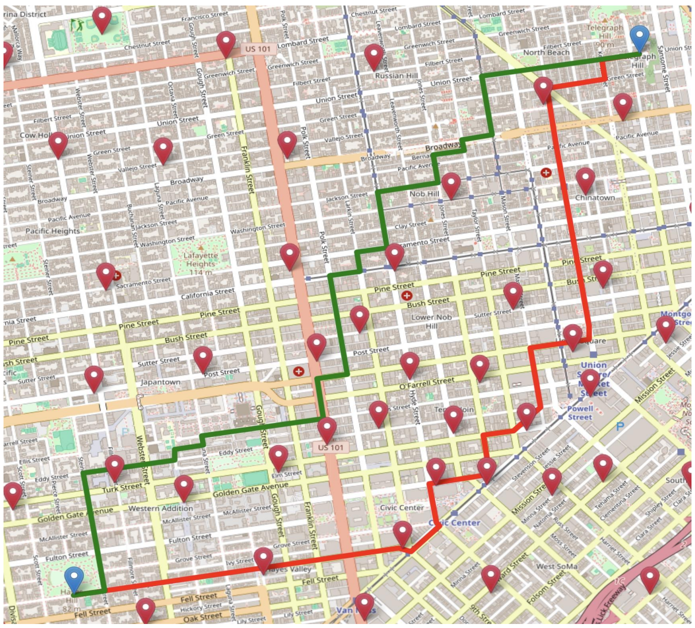
For visual representation, we use Leaflet to create an interactive map of San Francisco. This map is centered on the specified coordinates and displays both the recommended route and the identified crime hotspots. The Flask framework plays a crucial role in seamlessly integrating the front-end and back-end operations, enabling the dynamic display of nodal points and hotspots on the map.

The initial step in our backend processing involves loading a pre-processed graph of San Francisco, created using OSMnx, and used **200 most dense crime hotspots** found using **KMeans clustering**. Once the source and destination addresses are converted into coordinates, we determine the nearest nodes on the San Francisco graph. We utilized OSMnx to remove nodes with High Crime Density from the graph. After excluding nodes in high-crime areas (red hotspots), **NetworkX** is utilized to find the optimal path that prioritizes safety and efficiency. This chosen path, along with the graph and crime hotspots, is then sent back to the front end for display.

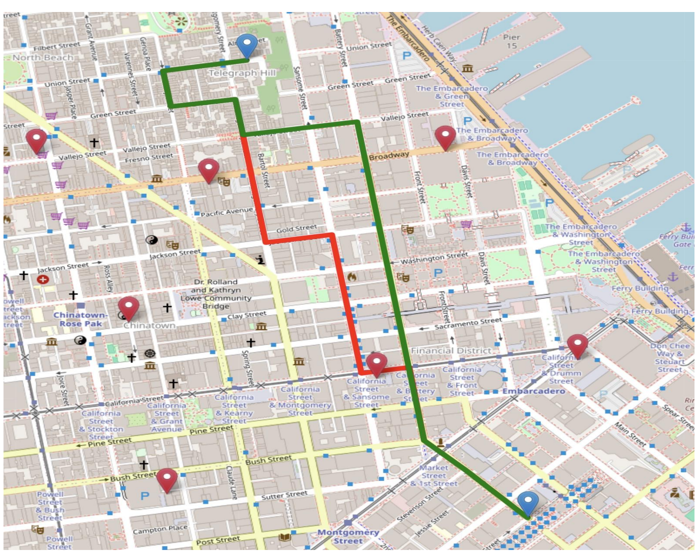
4.2 Results and Experimental Study

The following figures show the hazard free path between source and destination by avoiding certain crime spots which are marked as red.

A*:

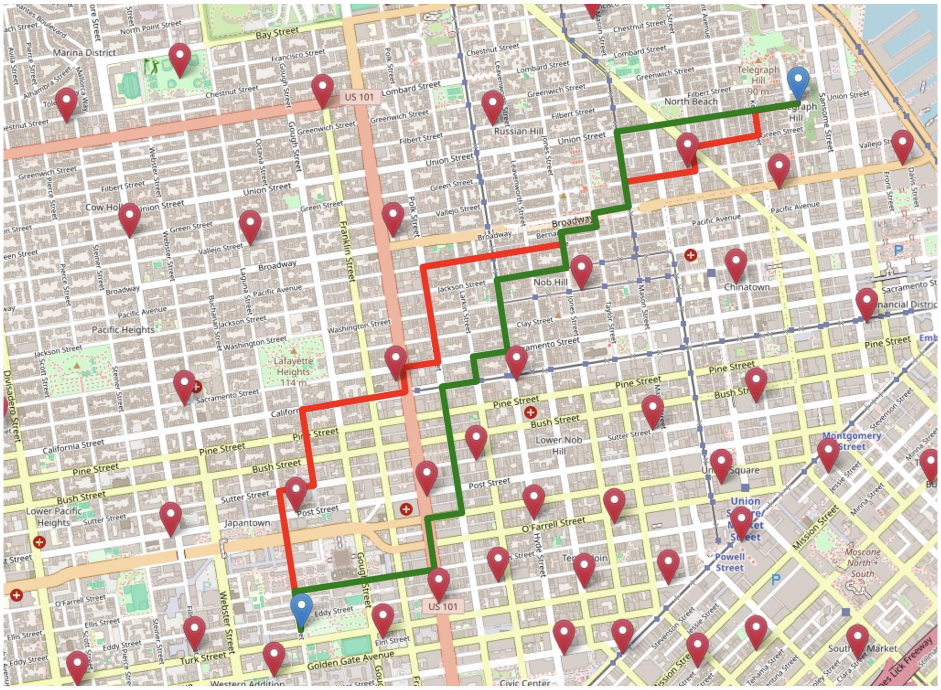


Here we see, both the red and the green paths are calculated using A* algorithm, but the green path is avoiding all the crime hotspots, whereas the red path is going through them.So we get the safest and shortest path using A* algorithm



Similarly in the above map, the dijkistras algorithm is used to get the shortest path.Similar to the A* The red path goes through one of the crime spots whereas the green spot is avoiding it to give the path.

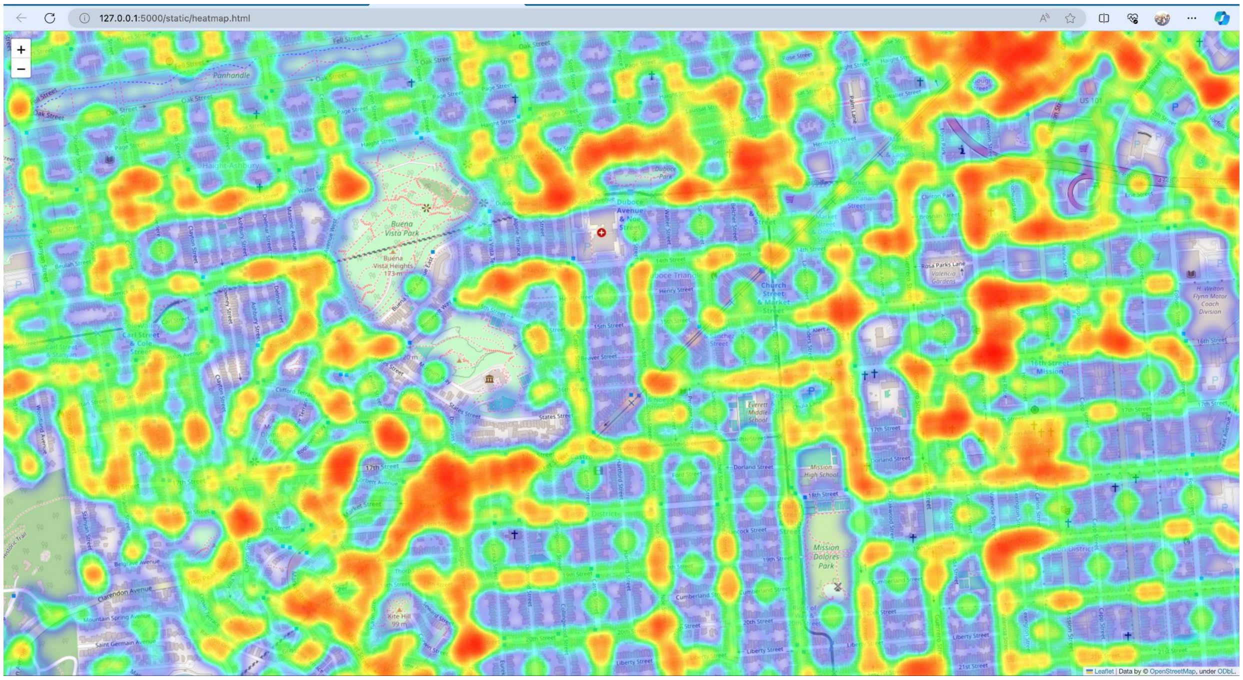
NetworkX:



The above map gives path using the inbuilt networkx path algorithm which is optimized Dijkistra's. As we see again the red and green path follow the same criteria as we have mentioned above.

In addition to plotting safe and efficient routes, our application displays the heat maps of the whole crime data to visualize crime concentration across different locations. We use the Folium library, which is well-suited for generating interactive maps, to accomplish this.

This visualization helps users to understand the frequency and distribution of crimes within the city at a glance. By overlaying these heat maps onto standard maps, we provide a dual-layered perspective: one that guides users along safe paths, and another that educates them about crime density in various areas.



4.3 Future Work

- **Optimization Techniques:** We can enhance performance through parallel processing using PyCuda, enabling the system to efficiently handle multiple factors like Covid-19, crime, and traffic simultaneously. We can incorporate spatial keyword search methods to effectively manage these varied constraints.
- **Nationwide Application:** we can use a system that is capable of calculating routes in numerous cities across the USA by analyzing actual hotspot data. This wide-reaching capability ensures relevance and utility in diverse urban contexts.
- **Accessibility and Open-Source Availability:** We can plan to make this tool widely accessible by open-sourcing it and hosting it on various platforms, including AWS, Android, and iOS. This approach allows for broad, general use and encourages community contributions and improvements.
- **Traffic and Public Transport Planning:** Leveraging AI and big data analytics, this tool processes multiple factors like vehicle density, public transport schedules, and road works to optimize traffic flow and public transport routes in cities nationwide. Hosted on Azure, it's also available for Android and iOS, facilitating easy access for city planners and the public.

CONCLUSION

In today's ever-changing world, where crime rates are steadily rising, our application emerges as an essential tool, specifically designed to enhance personal safety. By prioritizing safety, it offers users a reliable way to navigate between two locations, ensuring peace of mind during their journeys. This innovative application not only considers the fastest routes but also integrates crime data, enabling it to suggest safer alternatives. Its user-friendly interface simplifies route planning, making it accessible to everyone. Ultimately, this application is more than just a navigation tool; it's a companion for safer travel in an increasingly unpredictable world.

References

- Koritsoglou, K.; Tsoumanis, G.; Patras, V.; Fudos, I. Shortest Path Algorithms for Pedestrian Navigation Systems. Information 2022, 13, 269. https://doi.org/10.3390/info13060269
- Klunder, G.A. and Post, H.N., 2006. The shortest path problem on large‐scale real‐road networks. Networks: An International Journal, 48(4), pp.182-194.
- Wei H, Zhang S, He X. Shortest Path Algorithm in Dynamic Restricted Area Based on Unidirectional Road Network Model. Sensors (Basel). 2020 Dec 30;21(1):203. doi: 10.3390/s21010203. PMID: 33396895; PMCID: PMC7795470.
- Li, S., Dragicevic, S., Castro, F.A., Sester, M., Winter, S., Coltekin, A., Pettit, C., Jiang, B., Haworth, J., Stein, A. and Cheng, T., 2016. Geospatial big data handling theory and methods: A review and research challenges. ISPRS journal of Photogrammetry and Remote Sensing, 115, pp.119-133.
- Bates, S., 1987. Spatial and temporal analysis of crime. Research Bulletin, April.
- He Z, Lai R, Wang Z, Liu H, Deng M. Comparative Study of Approaches for Detecting Crime Hotspots with Considering Concentration and Shape Characteristics. Int J Environ Res Public Health. 2022 Nov 2;19(21):14350. doi: 10.3390/ijerph192114350. PMID: 36361227; PMCID: PMC9655231.
- Shariati M, Mesgari T, Kasraee M, Jahangiri-Rad M. Spatiotemporal analysis and hotspots detection of COVID-19 using geographic information system (March and April, 2020). J Environ Health Sci Eng. 2020 Oct 12;18(2):1499-1507. doi: 10.1007/s40201-020-00565-x. PMID: 33072340; PMCID: PMC7550202.