SMART URBAN NAVIGATION: A CRIME DATA-DRIVEN ROUTE OPTIMIZATION SYSTEM FOR SAFER TRAVEL IN BENGALURU

Project Reference No.: *48S_BE_6281*

College : P.E.S. University, Bengaluru
Branch : Department Of Computer Science And Engineering
Guide(S) : Dr. Sudeepa Roy Dey
Student(S): Ms. Chandra Prabha
Mr. Sampann Raheja
Ms. Sanampudi Sai Sukruthi
Mr. Raghavendra Gajanan Ambekar

Keywords:

Safe urban navigation, Danger index, K-means clustering, A* algorithm, Crime data analysis, Personalized routing.

Introduction/Background:

In an ever-changing world, rapid urbanization brings both significant advantages and downsides to booming development. In crowded cities, most commuters depend on roads for daily travel. Due to rising crime, crowded roads, crashes, and other threats, safety remains a big concern. This is especially true for women, children, late-night travelers, and people who don't know certain areas.

Navigation apps like Google Maps and Apple Maps are popular and widely used. However, they focus on finding the fastest route based on traffic and overlook safety as an important factor. Commuters often travel through dark streets, high-crime zones, or isolated areas. They might not see the potential risks around them. Conventional navigation tools lack crime-aware routing. This highlights the need for a new approach—one that prioritizes both efficient travel and enhanced safety.

We propose a solution that looks at the challenges of current navigation systems. It explores how safety-focused routing can change urban mobility. We also look at how crime-aware navigation affects the real world.

Objectives:

- Enhance Urban Navigation Safety: Develop a navigation system that prioritizes safety by incorporating crime and accident data into route optimization.
- Integrate Multiple Data Sources: Leverage historical crime records, accident data, and crowd-sensed information to provide safer route recommendations, effectively addressing safety concerns and enhancing urban navigation.
- Personalized Route Recommendations: Provide customized route preferences based on time sensitivity and safety concerns, offering users a personalized, flexible, and secure navigation experience based on their preferences.

Methodology:

To design a safety-focused navigation system, we gathered an extensive dataset with a primary focus on Bangalore, given its notably high rates of crime and traffic accidents. Data was sourced from officially backed Karnataka State Police records and other crime and accident datasets from Kaggle. This data from across multiple sources was then cleaned using Python and SQL to ensure uniformity. Outliers were removed using the IQR method.

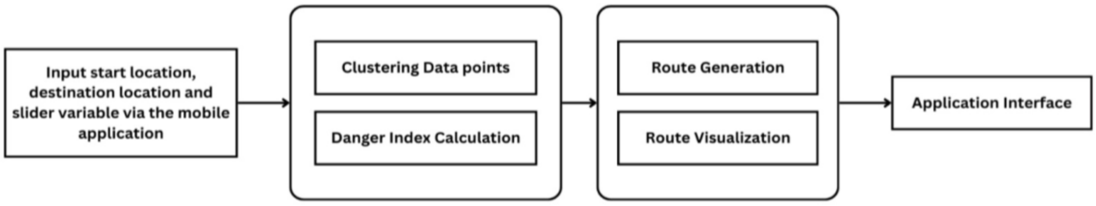
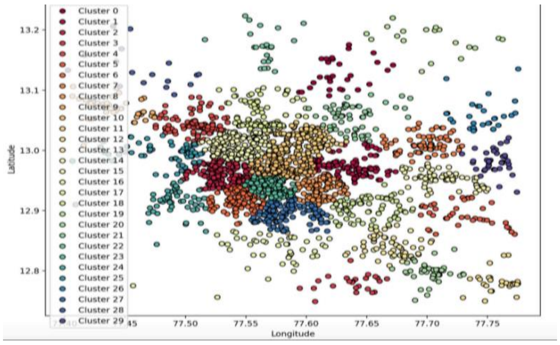


Fig. 1 Proposed Framework

A. Clustering Algorithms

Crime points were grouped into 30 spatial clusters using K-Means clustering to identify hotspots. K-Means was chosen over DBScan for its stability and scalability. These clusters enabled better crime pattern analysis.



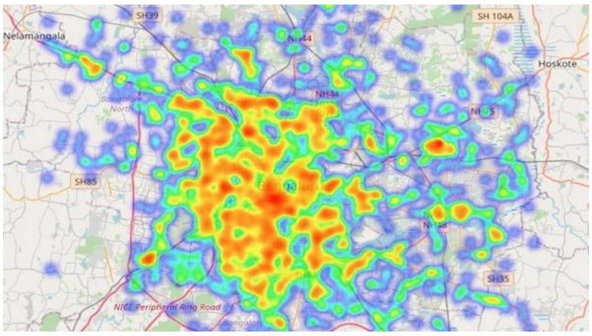


Fig 2. K-means Clustering

Fig. 3 The map highlights high-crime areas (hotspots).

B. Visualization

Folium was used to create heatmap overlays to visualize crime density across Bangalore. In *Fig. 3*, high-crime areas appear in shades of red, while safer zones are shown in green, aiding in crime hotspot identification and spatial crime analysis

C. Danger Index

Danger index is the key metric we used to evaluate the level of crime within a cluster. It is calculated as a weighted sum of crime counts and severity (based on Indian Penal Code penalties).

$$D_i = \sum_{j=1}^{n} (C_{ij} \times S_j)$$

• Di is the Danger Index for cluster i, n is the total number of crime types.

• Sj is the severity score for crime type j, Cij is the count of crime type j in cluster i.

To ensure comparability, we normalized these scores using min-max normalization:

$$D_i' = \frac{D_i - D_{min}}{D_{max} - D_{min}}$$

D. Pathfinding Algorithms

The results of our project are significantly influenced by the pathfinding algorithm used. A graph was constructed using centroids of clusters as nodes, and danger-weighted edges were used for routing.

- Dijkstra's Algorithm is a greedy method that explores all nodes to find the shortest paths, making it resource-heavy. It uses relaxation to update distances:

$$distance[v] = min(distance[v], distance[u] + weight(u, v))$$

- A* algorithm integrates distance and danger through a heuristic function and uses a priority queue for efficiency, which helps avoid high-risk clusters.

$$f(n) = g(n) + h(n)$$

  where:

  g(n) = cost from the start node to the current node.

  h(n) = danger-weighted estimate from current to target.

- GNN learns safety patterns from geographic and crime data using two Graph Convolutional Layers (GCN):
  - Layer 1 - aggregated crime-weighted neighbor data.
  - Layer 2- outputs a danger score for each node.

  During routing, GNN combines physical distance with danger scores to update paths and minimizes MSE loss to predict safer paths dynamically.

E. Mobile Application Features

The mobile app, built using Flutter and developed in Android Studio, integrates the Google Maps API for core navigation functionality. Backend services are powered by Supabase for authentication and data handling, with Flutter_bloc managing app state. Ngrok was used to expose the backend server publicly, enabling mobile access to locally hosted models and APIs

To enhance usability, the app includes:

- Slider: Allows users to adjust route preferences between safety and travel time (0-100 scale).
- Report Page: Enables users to anonymously report incidents, supporting crowd-sourced data.
- Recent Routes: Displays searched routes from the past two weeks.

Results & Conclusions:

Metric	A*	GNN	Dijkstra
Total Physical Distance (km)	9.46	10.39	13.03
Total Danger Index	0.32	0.46	0.43
Avg Danger per Cluster	0.11	0.08	0.11
Clusters Traversed	3	6	4
Execution Time (seconds)	20,595.05	29,468.72	31,339.53
Peak Memory Usage (MB)	447.71	2,339.14	6,940.63

We noted our observations across multiple tests and reached the following conclusions:

The A* algorithm, using a heuristic-driven approach, achieves optimal performance. By combining the actual cost of reaching a node with an estimated cost to reach the goal, A* dynamically balances exploration and exploitation. In selecting the optimal route by preferring safer clusters early in the search, the algorithm reduces cluster transitions, allowing it to prioritize paths that minimize physical distance and cumulative danger. The algorithm's heuristic ensures that it reduces execution time and memory usage by evaluating fewer nodes as compared to Dijkstra.

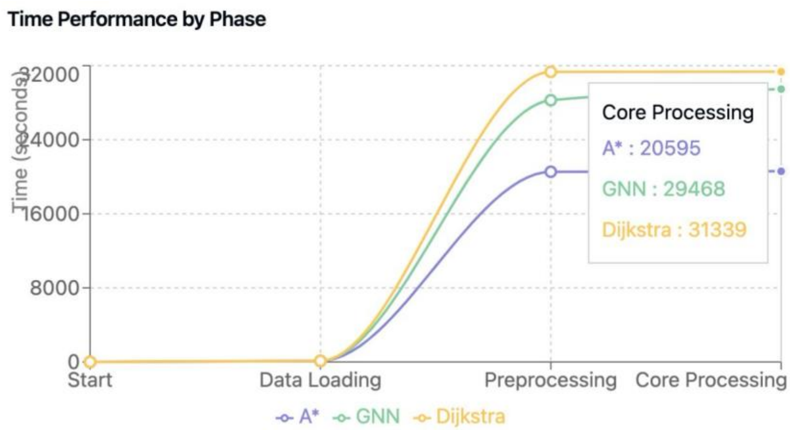


Fig. 4 Time performance comparison

Fig. 4 depicts the time performance of the three pathfinding algorithms. We can see that A* significantly outperforms the others in computational efficiency, making it a preferable choice for safe route optimization in our system. In conclusion, Dijkstra's method is unsuitable for safety-critical navigation systems as it sacrifices computational efficiency to provide longer, somewhat safer routes. Although GNN shows potential for future improvements, it requires significant refinement to compete with A*, and due to its drawbacks, Dijkstra is inappropriate in situations where efficiency and safety are crucial. The most successful was A*, which provided a balance between resource usage, efficiency, and safety.

In conclusion, by leveraging spatial crime data, our approach systematically penalizes high-risk regions in route computation, reducing exposure to unsafe areas. Our system utilizes crime statistics with a data-agnostic approach, employing K-Means clustering for crime pattern analysis and the A* algorithm for optimized pathfinding. Crime density functions are incorporated into the cost metric, dynamically influencing route selection based on severity and historical trends

Project Outcome & Industry Relevance:

Our project has strong practical relevance, especially in the context of rising crime rates in cities like Bangalore. Most existing navigation systems prioritize shortest or fastest routes without factoring in safety. Our app fills this critical gap by actively avoiding unsafe areas, helping users— particularly women, children, and night-time commuters, make safer travel decisions.

The app functions like standard navigation tools but incorporates real crime data using k-means clustering to identify crime hotspots and use these clusters to generate safer routes. In terms of real-world application, it can be used as a working mobile app to navigate through the city safely, the beta version of which has already been developed by us.

From a research perspective, the project contributes a unique approach to the field by modifying the traditional A* algorithm. We've introduced a custom cost function:

tentative g_score = g_score[current node] + distance(current node, neighbor) + danger index[neighbor]

This formula accounts not only for distance but also for the safety of the path, leading to more practical and risk-aware route suggestions.

Working Model vs. Simulation/Study:

The project involved the development of a fully functional, end-to-end working model. We implemented a safety routing system using k-means clustering for hotspot identification and a custom A* algorithm tailored for safe route computation. In addition to the algorithmic model, we also developed a mobile application that integrates these components to provide a complete, real-world solution. Hence, this was not merely a simulation or theoretical study, but a practical implementation with a working physical model in the form of a mobile application.

Project Outcomes and Learnings:

One of the key outcomes of our project is to create awareness and educate people about rising crime trends and the importance of making informed decisions while navigating cities. A system like this can contribute to a greater sense of safety, especially for vulnerable groups.

In addition, we developed a fully functional safety-focused navigation app that helps users avoid crime-prone areas by suggesting safer routes.

Through the process, we learned how to apply machine learning and custom algorithms to real-world safety concerns. We also gained insights into user-centric design and how incorporating safety data into routing logic can significantly enhance the everyday travel experience.

Future Scope:

The results achieved above show significant scope for developing solutions in this area. Future work can include integrating real-time crime reports with sentiment analysis of social media posts and news feeds to detect emerging threats dynamically. This will allow the system to identify sudden spikes in crime-related incidents or public safety concerns, improving route adaptability. Beyond personal navigation, such systems have broader applications in law enforcement and emergency response. Crime-weighted routing can assist law enforcement in ensuring targeted surveillance in high-risk areas. Such advancements will enable more responsive and intelligence-driven navigation, significantly improving urban safety.