



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 15, Issue 3, March 2026**

# **Goe Spatial Based Nearest Place Finder Using Machine Learning**

**Mr.V.Pavan Kumar<sup>1</sup>, Tenali Meghana Sai<sup>2</sup>,Mamilla Hemanth Kumar<sup>3</sup>, Lavdiya Pavan Nayak <sup>4</sup>,  
Doddi Srinadh Reddy<sup>5</sup>**

Assistant Professor, Dept. of CSE-Data Science, ACE Engineering College, Hyderabad, Telangana, India<sup>1</sup>

IV B. Tech Students, Department of CSE (Data Science), ACE Engineering College, Hyderabad, Telangana, India<sup>2,3,4,5</sup>

**ABSTRACT:** The increasing demand for location-based services has driven the need for accurate and customizable geo-spatial search systems. Traditional applications such as Google Maps work well in urban regions but often lack flexibility in rural or offline environments where custom datasets are required. This project aims to address these limitations by allowing users to search for nearby places using manually provided or device-based latitude and longitude coordinates. The system supports village-level and district-level datasets, making it suitable for smart-city applications, emergency support systems, and personalized search tools.

The proposed system integrates Python as the backend engine to process geo-spatial data and Leaflet.js as the frontend tool to visualize locations on an interactive map. The Haversine formula is applied to calculate the distance between the user's location and various points of interest stored in the dataset. The system then sorts and displays the nearest locations both on the map and in a tabular format. By supporting custom datasets and offline functionality, the system ensures reliable performance even in limited connectivity environments, addressing the need for accessible and lightweight geo-spatial solutions.

**KEYWORDS:** Geo-spatial Search, Haversine Formula, Leaflet.js, Nearest Location, Custom Datasets, Web Mapping.

## **I. INTRODUCTION**

The foundation of this geo-spatial project lies in its ability to acquire, process, and manage diverse location-based datasets. The system supports multiple data sources, including government records, publicly available datasets from platforms like Kaggle, and manually created custom CSV files. These datasets typically contain essential attributes such as place names, latitude and longitude coordinates, categories (e.g., hospitals, schools, banks), and additional metadata like addresses or contact details. By accommodating both rural village-level data and broader district or urban datasets, the system ensures flexibility and relevance across different geographical contexts. Before the data can be utilized for intelligent search and analysis, it undergoes a comprehensive preprocessing and cleaning phase. Raw geo-spatial data often contains inconsistencies such as missing coordinates, incorrect entries, duplicate records, or formatting errors. The system applies validation techniques to correct or discard invalid entries, normalizes coordinate formats, and structures the data for efficient querying. This step is critical to ensure that subsequent distance calculations and clustering algorithms operate on high-quality input, directly impacting the accuracy and reliability of search results.

To handle large volumes of spatial information efficiently, the project employs advanced spatial indexing techniques such as KD-Tree and R-Tree. These data structures organize points in multi-dimensional space, enabling rapid nearest-neighbour searches even when working with thousands of location records. Additionally, the Haversine formula is applied to compute accurate great-circle distances between geographic coordinates. This combination of indexing and mathematical computation ensures that the system can deliver fast response times while maintaining high precision in distance measurements.

For persistent storage and advanced querying, the backend integrates a Post GIS-enabled PostgreSQL database, which provides robust support for spatial data types and operations. This allows the system to execute complex geo-spatial queries, filter results by categories, and manage dynamic updates to datasets. Furthermore, the system

implements local caching mechanisms that store frequently accessed data and precomputed clusters on the user's device. This offline capability ensures that users in remote or low-connectivity areas can still perform reliable nearest-place searches without relying on continuous internet access.

## II. LITERATURE SURVEY

- **R. Agarwal et al. (2023)**

The authors explored machine-learning-assisted geo-spatial clustering techniques for urban planning. Their study showed that K-Means and DBSCAN algorithms significantly improved geographic segmentation accuracy. However, their system lacked real-time user-level search and focused primarily on large-scale city datasets rather than village-level mapping. This highlights the need for a more localized and user-interactive approach.

- **M. Khan & Al-Turjman (2022)**

Their work examined mobile GPS-based applications for nearest hospital and emergency service identification. While the results were promising, the approach relied heavily on live internet connectivity and cloud APIs, making it unsuitable for rural or offline environments. Their system did not include spatial indexing or ML-based recommendation, which are essential for intelligent offline functionality.

- **P. Srinivasan et al. (2021)**

This study presented a Post GIS-based geo-spatial database design capable of performing advanced spatial queries. Although it achieved high performance for spatial data retrieval, it lacked machine learning integration and could not intelligently cluster or recommend locations based on user behaviour. This gap emphasizes the value of combining spatial databases with ML techniques.

- **OpenStreetMap Research Initiative (2020)**

The initiative provided insights into open-source mapping and crowd-sourced geo-data collection. While OSM offers large datasets, its coverage is inconsistent in rural India, making it insufficient for localized nearest-place search without custom data mining and preprocessing. This reinforces the importance of supporting custom and offline datasets in geo-spatial systems.

- **Google Local Search Dataset Analysis (2019)**

Google's search engine was evaluated for location accuracy and personalization. Results indicated that while Google Maps provides robust general navigation, its performance declined for remote or newly developed regions. It also does not support offline ML-based recommendation or customized databases, underscoring the need for alternative solutions tailored to local requirements.

## III. EXISTING SYSTEM

Existing systems include popular navigation applications and basic location-based service tools such as Google Maps, Bing Maps, and Apple Maps. These platforms provide general navigation and nearest-place search functionalities. However, they remain heavily dependent on internet connectivity and global datasets. Their accuracy is limited, especially in rural or newly mapped areas. They also lack ML-based personalization and offline clustering capabilities. GPS-based mobile applications use device coordinates to locate nearby facilities. These apps often rely on third-party APIs with limited customization options. They provide no data mining or machine learning analysis for intelligent recommendations. Database-driven spatial search tools utilize SQL queries with geospatial extensions. While suitable for small datasets, they lack scalability for larger applications. They do not support clustering, ranking, or intelligent recommendation features. OSM-based regional mapping tools depend heavily on crowd-sourced data. Their coverage varies with location and remains incomplete in rural contexts, while also lacking ML integration and offline functionality.

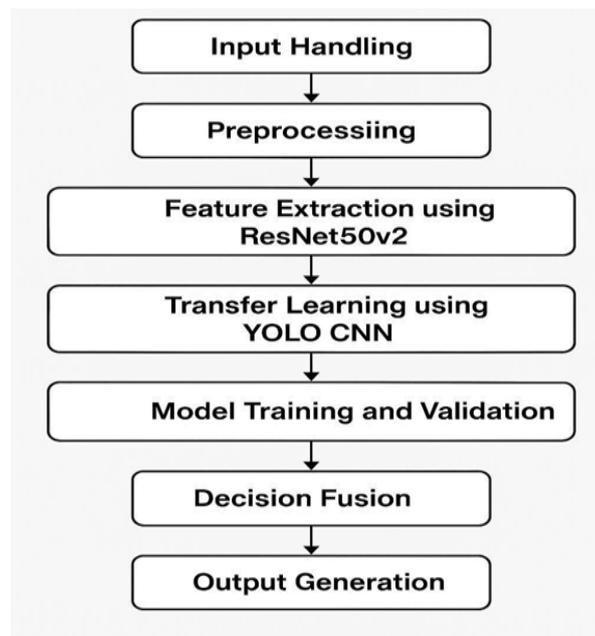
## IV. PROPOSED SYSTEM

The proposed Geo-Spatial Based Nearest Place Finder Using Machine Learning and Data Mining addresses the limitations of existing systems through an intelligent and integrated approach. The system utilizes K-Means clustering to group and analyze geographic locations, enabling efficient organization of spatial data. It applies the Haversine formula for accurate distance computation combined with spatial indexing structures such as KD-Tree and R-Tree for fast nearest-neighbour search. ML-based ranking algorithms and user preference learning are incorporated to deliver personalized search results tailored to individual needs. The backend integrates Post GIS with Python Flask, providing

scalable geo-spatial data storage and processing capabilities. Results are displayed through an interactive map visualization using Leaflet.js, offering users an intuitive and engaging experience. A key feature of the proposed system is its ability to allow users to upload and use local datasets, significantly improving accuracy in rural and underserved areas. Continuous data mining techniques are employed to analyze user behaviour and search patterns, enhancing recommendations and overall search efficiency over time.

The proposed system offers several distinct advantages over existing solutions. It achieves high accuracy through ML-based clustering and geodesic distance computation methods. Fast search performance is ensured with the implementation of KD-Tree and R-Tree spatial indexing structures. Personalized recommendations are generated through continuous data mining of user preferences and historical search patterns. The system supports wide dataset compatibility, functioning effectively for villages, towns, and cities alike. Its scalable architecture makes it suitable for future integration with smart city applications and larger geographic regions. Finally, the interactive and user-friendly map interface provides seamless navigation and visualization of nearest places, making the system accessible and practical for real-world use.

## V. METHODOLOGY

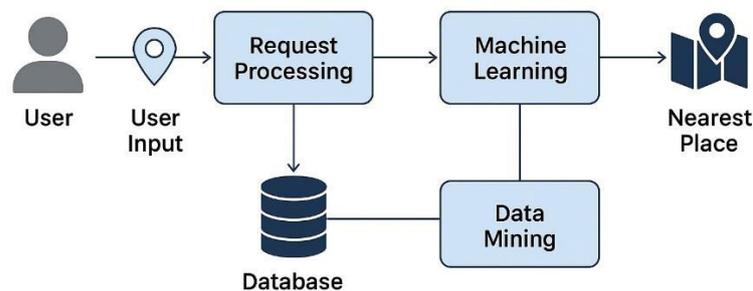


The proposed system begins with comprehensive data collection from diverse sources including government open-data portals, OpenStreetMap, local administrative authorities, and CSV or GeoJSON files containing coordinates, place names, and categories. Each dataset includes essential attributes such as latitude, longitude, place name, category, ratings, and accessibility levels. Once collected, the data undergoes rigorous preprocessing where duplicates, incomplete entries, and incorrect coordinates are identified and removed. The raw data is then converted into structured formats suitable for machine learning and spatial processing. Locations are categorized into groups such as hospitals, banks, schools, ATMs, and police stations for organized search functionality. Coordinates are normalized to prepare them for subsequent clustering and indexing operations. To enable fast nearest-neighbour searches, the system employs R-Tree for bounding-box based queries and KD-Tree for high-performance search. These indexing techniques significantly reduce computation time by organizing spatial data in multi-dimensional structures. For accurate distance measurement, the system implements both the Haversine formula and geodesic distance computation methods. These mathematical approaches ensure high precision in calculating real-world distances between coordinates. The combination of spatial indexing and accurate distance computation forms the foundation for efficient location retrieval across both urban and rural datasets.

The system leverages K-Means clustering to geographically group nearby locations, enabling faster searches within localized clusters rather than scanning the complete dataset. This clustering approach not only improves search efficiency but also aids in visual grouping when displaying results on interactive maps. Following clustering, an ML-based ranking algorithm is applied to sort locations based on calculated distance, user preferences, and historical search patterns. Ratings and accessibility levels are also factored into the ranking process to ensure personalized and relevant results for each user. A distinctive feature of the system is its offline caching mechanism, which stores frequently accessed datasets and map tiles locally on the user's device. This ensures that nearest-place search functionality remains available even without internet connectivity in remote areas. Continuous data mining techniques analyze user search history and frequently visited categories to identify location popularity trends over time. These mined data patterns are used to enhance recommendation accuracy and personalize future search results. Finally, all results are visualized on an interactive digital map using Leaflet.js with category-specific icons and colour codes. Each location displays detailed information including distance, name, category, and ranking for comprehensive understanding. The complete methodology ensures an intelligent, accurate, and user-friendly geo-spatial search experience for all users.

## V. SYSTEM ARCHITECTURE

### Geo-Spatial Based Nearest Place Finder Using Machine Learning



The proposed system follows a modular architecture that begins with the Client interface, where users enter a location or select a category such as hospital, ATM, or bank as input. This request is sent to the Request Processing component, which validates the coordinates and category before communicating with the database to fetch the required geo-spatial records. The Database serves as the central repository storing all geo-spatial datasets including coordinates, categories, place names, ratings, and accessibility levels, providing necessary data to both Request Processing and the Data Mining module. The Data Mining Module then extracts relevant places from the database based on the specified category, performing essential cleaning operations such as filtering irrelevant entries, removing duplicates, and organizing data into structured formats suitable for analysis.

The refined data is passed to the Machine Learning Module, which forms the core intelligence of the system by performing clustering, ranking, and nearest-neighbour analysis. It computes distances using accurate formulas, determines relevance scores, and prioritizes places based on multiple factors including user preferences and historical patterns. This module identifies the most suitable nearest locations by applying K-Means clustering and ML-based ranking algorithms. Finally, the processed results are sent to the Final Output stage, where the ML module returns the nearest place details and the user sees the results visually displayed using map markers, icons, and category-specific labels on the interactive interface. This end-to-end architecture ensures seamless data flow from user input to intelligent processing and final visualization.

VI. RESULTS

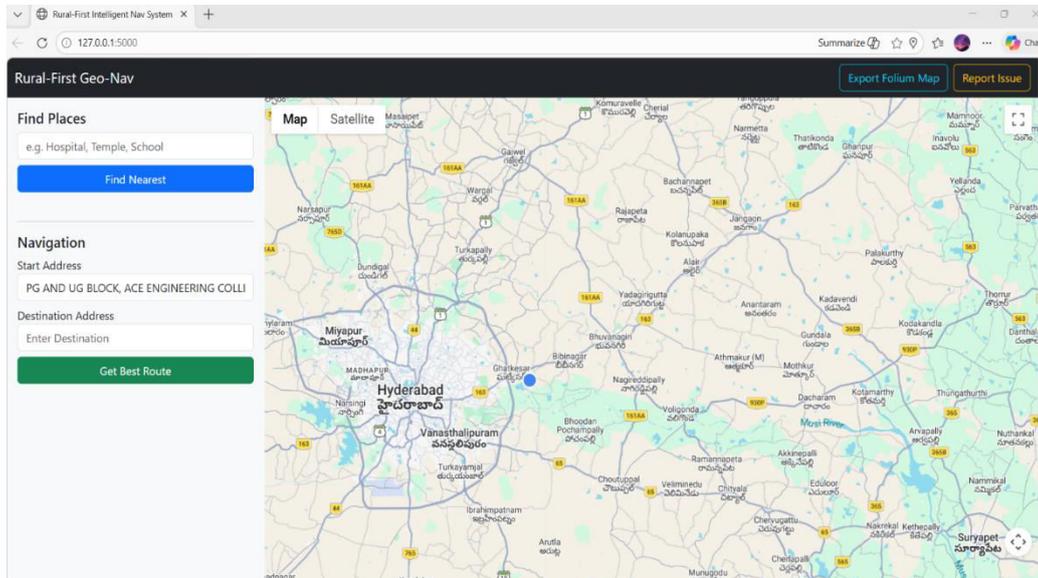


Fig. 6.1 Home Page

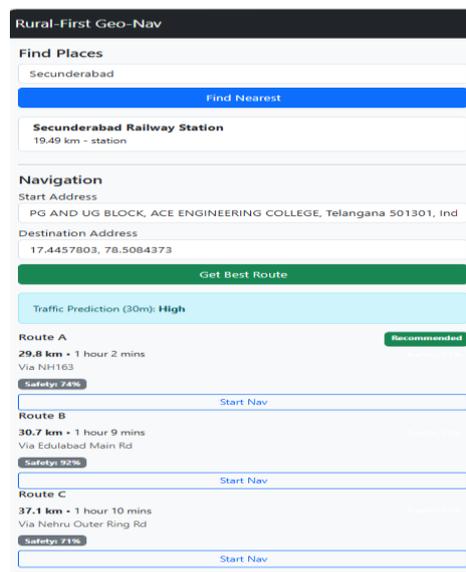


Fig. 6.2 Search Destination

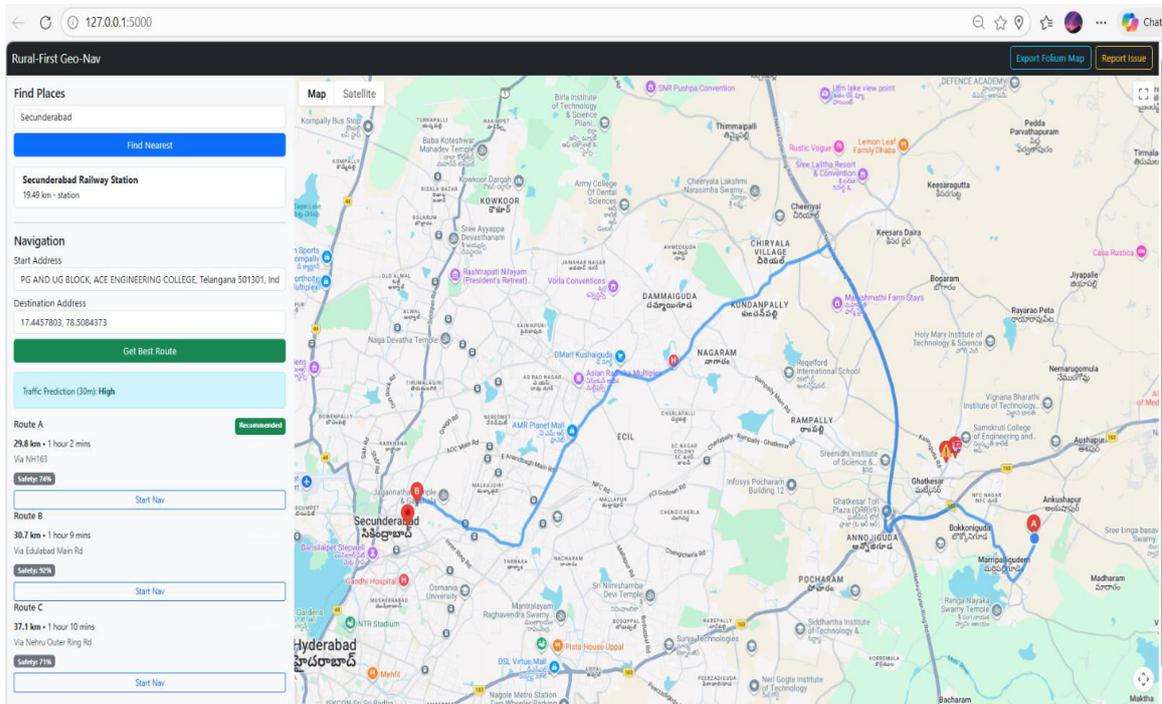


Fig. 6.3 Map Overview

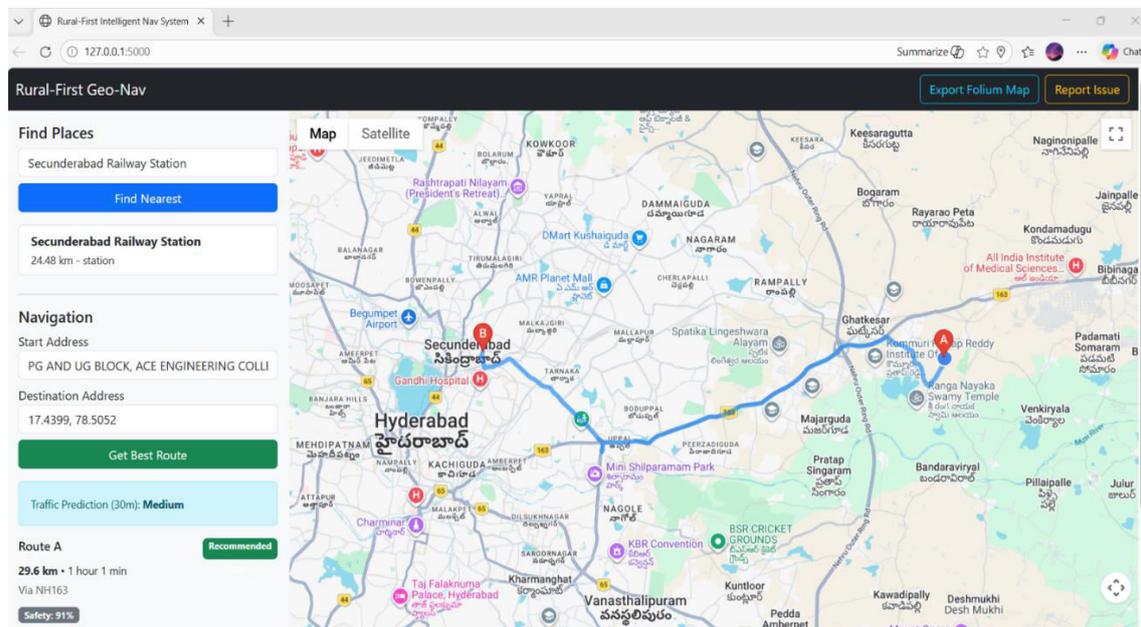


Fig. 6.4 Finding Nearest Route

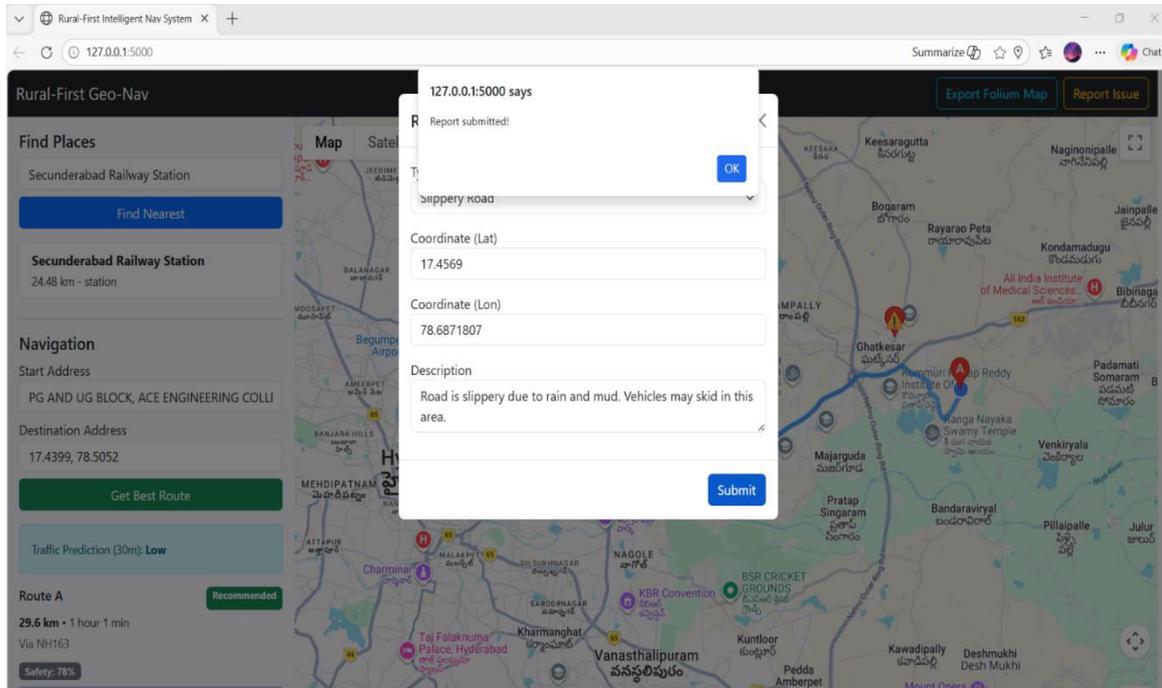


Fig. 6.5 Report Issue

## VII. CONCLUSION

The Geo-Spatial Based Nearest Place Finder developed in this project successfully demonstrates how machine learning and spatial data processing can be combined to provide accurate, fast, and user-friendly location-based services. By integrating geo-spatial datasets, Haversine distance computation, spatial indexing structures like KD-Tree and R-Tree, and clustering techniques, the system is able to efficiently identify the nearest essential places based on the user's current location. The interactive map visualization further enhances the user experience by providing clear and intuitive access to important places such as hospitals, schools, banks, and ATMs.

Overall, the system addresses the limitations found in existing map-based applications, especially in rural and semi-urban areas where data accuracy and offline availability are major concerns. With its modular architecture, offline support, and personalized ranking features, the application offers a reliable tool for both everyday users and local administrative planning. Future enhancements such as real-time traffic integration, route optimization, voice-based search, and dynamic dataset updates can further improve performance and usability, making the system more robust and widely applicable.

## VIII. FUTURE SCOPE

The proposed geo-spatial based intelligent navigation system provides an efficient solution for identifying the nearest places and recommending routes within localized environments. However, there is significant scope for further improvement to enhance the system's robustness, scalability, and intelligence. In the future, the system can be integrated with real-time GPS tracking and live traffic information obtained from sensors, mobile devices, or third-party APIs. This enhancement would enable dynamic route optimization and provide real-time navigation guidance, thereby improving accuracy and overall user experience.

In addition, advanced machine learning and deep learning models such as neural networks and time-series prediction techniques can be incorporated to improve traffic prediction and congestion analysis. The system can also be expanded by developing a dedicated mobile application for Android and iOS platforms, allowing users to access navigation services conveniently from their smartphones. Features such as multilingual interfaces and voice-based

commands can further enhance usability, especially for users in rural and semi-urban areas who may prefer simple and interactive navigation assistance.

Furthermore, the system can be deployed on cloud platforms to support large-scale data storage, faster processing, and the ability to handle multiple users simultaneously. Integrating map-based visual analytics dashboards can help authorities analyze traffic patterns, frequently visited locations, and accident-prone areas. Future improvements may also include crowdsourced data validation mechanisms and enhanced security measures such as secure authentication and encrypted data transmission to protect user information. These advancements would transform the system into a more intelligent, scalable, and reliable navigation solution suitable for smart city development and emergency management applications.

#### REFERENCES

- [1] Wang X. et al. (2013) proposed an approximate **K-Nearest Neighbour spatial clustering algorithm using KD-Tree** to improve the efficiency of searching nearest locations in large spatial datasets. This approach helps in faster geospatial. <https://scholar.google.com/scholar?q=APPROXIMATE+K-NEAREST+NEIGHBOUR+BASED+SPATIAL+CLUSTERING+USING+KD-TREE+Wang+2013>
- [2] Temitope O. et al. (2018) applied **K-Means clustering to geospatial models** for optimizing data collection across large geographic regions. The clustering method helps group nearby spatial data efficiently for improved analysis. <https://scholar.google.com/scholar?q=K-Means+clustering+of+inputs+to+a+geospatial+model+Temitope+2018>
- [3] Nurhayati et al. (2019) designed an application to **find nearby locations using the Haversine formula and Euclidean distance** for accurate geographic distance calculation. This method helps identify the nearest place. <https://scholar.google.com/scholar?q=On+Designing+Application+of+Finding+Nearby+Islamic+Boarding+Schools+using+Haversine+Formula+Nurhayati+2019>
- [4] Rani M. et al. (2021) proposed a **nearest location search system using the K-Nearest Neighbours (KNN) algorithm** to classify spatial data and find nearby places. The system uses distance-based calculations on Rank. <https://scholar.google.com/scholar?q=Nearest+Location+Search+Using+K-Nearest+Neighbours+Algorithm+Rani+2021>



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details