Safest Evacuation Path Determination During Flood

Ramandeep

Indian Institute of Technology
Jodhpur, Rajasthan, India

Contact:Gmail
Ramandeepshakya165

***Abstract*—India, a nation prone to a diverse array of natural and man-made disasters, grapples with the daunting task of fortifying its resilience and emergency response mechanisms. Among these adversities, floods stand as a prominent threat, inflicting widespread devastation across the country. This paper delves into the intricacies of flood disasters in India, elucidating their multifaceted impacts on lives, livelihoods, infrastructure, and public health. Drawing upon recent flood statistics and case studies, particularly focusing on the recurrent inundations in Chennai, the paper highlights the urgent need for innovative solutions to mitigate the dire consequences of such calamities. In response, this study proposes a pioneering approach: an ML-powered evacuation management system tailored for urban environments, with Chennai as a primary focus. Leveraging historical data and advanced algorithms, this system aims to predict flood trajectories and delineate optimal evacuation routes, thereby revolutionizing traditional evacuation strategies marred by infrastructural limitations and communication breakdowns. Through seamless integration of diverse data sources and adaptability to varied flood scenarios, this transformative solution not only seeks to safeguard Chennai but also offers a scalable blueprint for flood management in analogous urban centers nationwide. This paper advocates for the adoption of AI-driven technologies as imperative tools in enhancing disaster preparedness and resilience, underscoring the exigency of concerted efforts to safeguard vulnerable communities amidst the specter of rising waters.**

Keywords—India, Flood disasters, Resilience, Emergency response, Chennai, ML-powered evacuation management system, Urban flooding, Predictive algorithms, Disaster Preparedness, Community resilience.

I. Introduction

India's vulnerability to a myriad of natural and anthropogenic disasters poses a formidable challenge to its socio-economic fabric and public safety. Among these adversities, floods emerge as recurrent catastrophes, inflicting substantial human, infrastructural, and economic losses. The perils of flooding in India encompass a broad spectrum of ramifications, from loss of life and displacement to economic upheaval and public health crises. Against this backdrop, this paper elucidates the gravity of flood disasters in the Indian context, with a particular focus on the urban landscape of Chennai. The city's tumultuous history with floods, punctuated by catastrophic events in recent years, underscores the urgent imperative for innovative solutions to bolster its resilience. Central to this endeavor is the proposition of an ML-powered evacuation management system, meticulously crafted to navigate the intricate urban topography of Chennai and beyond. By harnessing the predictive prowess of machine learning algorithms and real-time data streams, this system endeavors to revolutionize conventional evacuation paradigms, offering residents and emergency services invaluable insights into safe and efficient escape routes. Moreover, this study advocates for the scalability of such solutions, positing Chennai as a microcosm of broader global challenges in flood management.

II. Background

India grapples with a diverse array of natural and man-made disasters, including floods, cyclones, earthquakes, droughts, landslides, and industrial accidents. These calamities pose significant challenges to the nation's resilience and emergency response capabilities. The below image shows the history of disasters have been faces by India.

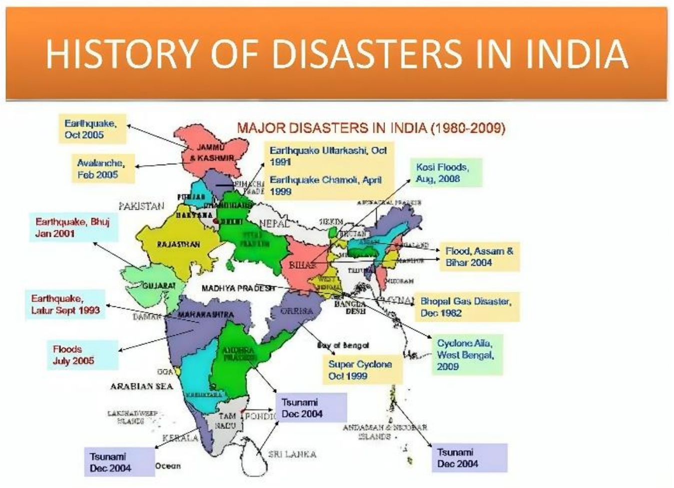


Fig 1. History of disasters in India.

A. *Understanding flood disaster in India:*

Floods are one of the most common natural disasters in India. They can be caused by a variety of factors, including heavy monsoon rains, melting snow in the Himalayas, and cyclones. Some of the most devastating floods in Indian history include the 1954 Bihar floods, the 2004 Indian Ocean tsunami, and the 2013 Uttarakhand floods.

B. *Recent Flood Statistics in India:*

According to the National Disaster Management Authority (NDMA), India faces a constant struggle against floods. On average, every year floods inundate over 75 lakh hectares of land, displacing communities and tragically claiming around 1,600 lives. The year 2023 witnessed a series of devastating floods across the country.

- **Himalayan Floods (October):** Heavy rains triggered flash floods and landslides in Sikkim, leaving at least 14 dead and impacting over 22,000 people.
- **North India Floods (June-July):** Monsoon rains unleashed their fury, causing floods in several North Indian states, leading to widespread damage and forced displacement.
- **Chennai Floods (December):** Cyclone Michaung dumped heavy rain, triggering floods in Chennai and surrounding coastal areas of Tamil Nadu.

These are just a few examples. Floods are a recurring nightmare for many regions in India, including Assam, Gujarat, Kerala, Maharashtra, and many more. There are several reasons for the flood and one of them is heavy rainfall fig 2. Shows the statistics of rainfall in India that significantly contributed to the flood.

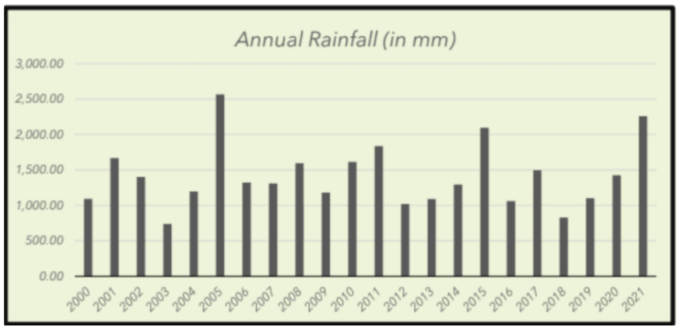
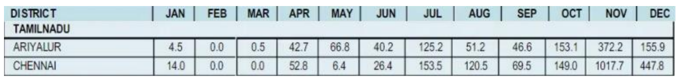


Fig 2. Annual rainfall of India since year 2000.

C. Vulnerability of Chennai Floods:

Chennai, a densely populated city in India, stands as a stark testament to this disaster. Its history with floods is marked by events in 2013, 2015, 2017, 2020, 2021, and 2023. The 2015 deluge, with a staggering 490 mm of rainfall in a single day, stands out as a grim reminder of the devastating consequences these disasters can bring. While subsequent years saw varying intensities, the consistent pattern of disruption underscores Chennai's vulnerability. The impact of these floods extends beyond immediate water damage, encompassing loss of life, with the highest toll in 2015 at around 300 deaths. Economic losses, estimated in billions of rupees, add another layer of devastation. Each major flood event has displaced thousands, leaving temporary shelter a recurring necessity. Roads, bridges, buildings, and utilities sustain significant damage.



DISTRICT	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
TAMILNADU												
ARIYALUR	4.5	0.0	0.5	42.7	66.8	40.2	125.2	51.2	46.6	153.1	372.2	155.9
CHENNAI	14.0	0.0	0.0	52.8	6.4	26.4	153.5	120.5	69.5	149.0	1017.7	447.8

Fig 3. Rainfall statistics of Chennai of 2015.

Fig 3 shows that most of the rainfall happened in October, November, and December, which contributed to the flood in 2015.

D. *Dataset Description:*

We have used the datasets mentioned in the reference as drive links. The main dataset chennai_flood_inundation_inches.csv has many columns but the three important columns are longitude, latitude, and flood intensity at location. But it does not have the rainfall data. I manually added the rainfall data from other datasets.

- E. *APIs used:*
  - *Google Maps API:*
  - *OpenWeatherMap API:*
  - *OpenStreetMap API:*
  - *Openmeteo API*
  - *OpenElevation API*
  - *Metiomatics API*

III. Methodology

These challenges make a sophisticated evacuation management system an urgent priority. Our project proposes a transformative solution: an ML-powered system designed specifically for Chennai's urban landscape. This system harnesses the power of historical data and advanced route optimization algorithms to predict flood paths and identify not just the quickest, but the safest evacuation routes. The project aims to integrate APIs such as Google Maps for traffic insights and OpenWeatherMap for up-to-date weather conditions into the ML models for predictions.

A. Solution Flowchart:

Fig. 4 shows the flowchart of our solution. Which consists of 4 processing steps and two user steps (input and output). Each processing step uses datasets and APIs to train and fetch data. Now we will show the step-by-step process.

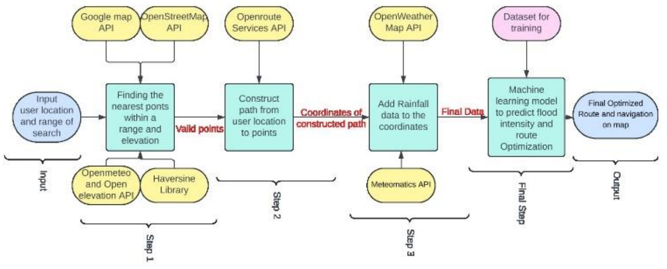

Google map API
OpenStreetMap API
Openroute Services API
OpenWeather Map API
Dataset for training
Input user location and range of search
Finding the nearest ponts within a range and elevation
Valid points
Construct path from user location to points
Coordinates of constructed path
Add Rainfall data to the coordinates
Final Data
Machine learning model to predict flood intensity and route Optimization
Final Optimized Route and navigation on map
Openmeteo and Open elevation API
Haversine Library
Meteomatics API
Input
Step 1
Step 2
Step 3
Final Step
Output

Fig. 4 flowchart of the solution.

1) *User input:* As the first step of the solution, the user will specify their current location and indicate the approximate radius of a circular area they intend to go. Additionally, the user will provide the desired elevation within this specified radius.

2) *Finding the nearest location:* Based on the user input we find the nearest location coordinates by calculating the range of search using the current location, OpenStreetMap API, and Google map API then filter out the points using the haversine algorithm, and then again we filter out the coordinates based on the elevation input using Openmeteo API, and Open Elevation

API. Fig 5,6,7 shows the coordinates found for the location: *13.0313524, 80.2171428(Chennai)*, radius for a circular search: 2 *kilometers*, 1 *kilometer*, having an elevation of 10 meters, 5 meters.

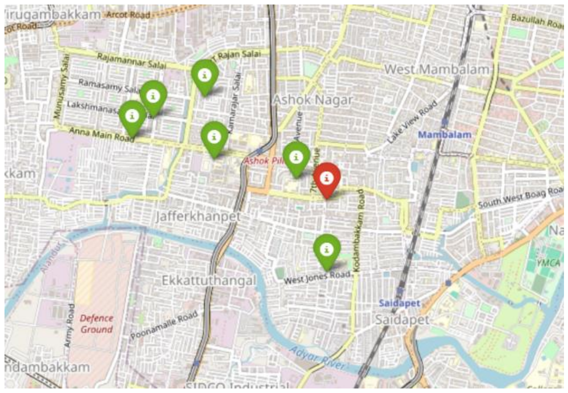


Fig 5. The nearest points around the user for 2 kilometers and have 10 meters of relative elevation.

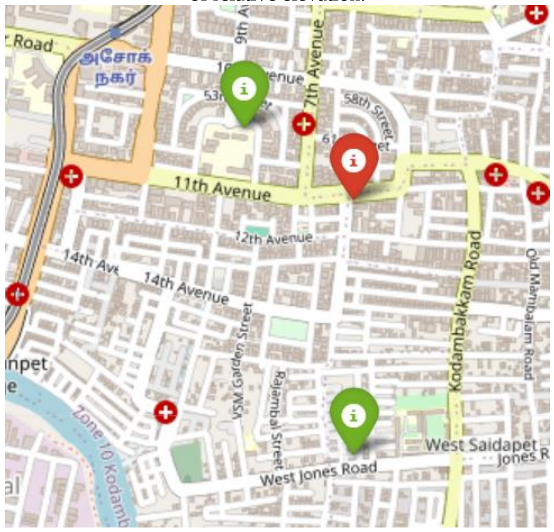


Fig. 6. The nearest points around the user for 1 kilometer and have 10 meters of relative elevation.

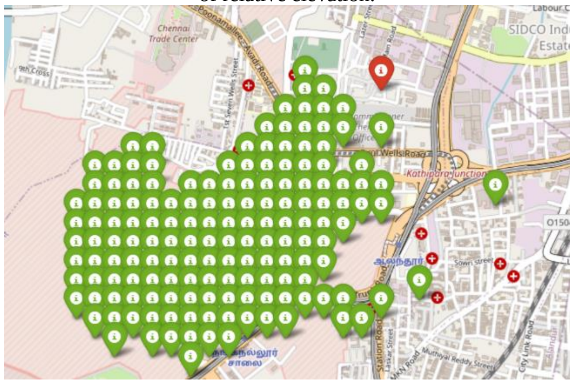


Fig. 7. The nearest points around the user for 2 kilometers and have 5 meters of relative elevation for the hilly region (13.011596, 80.200567).

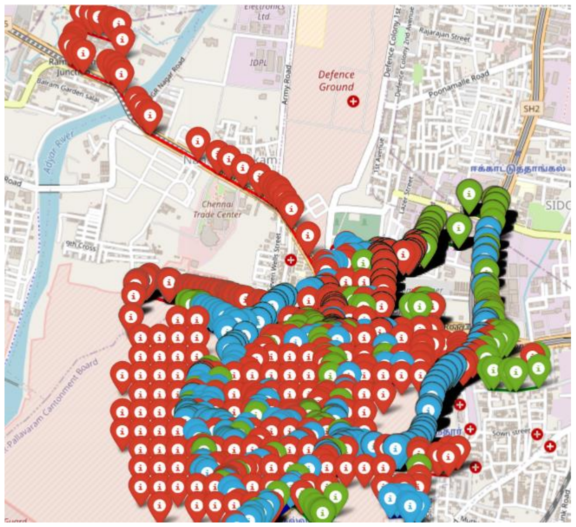


Fig. 8 paths from the user location (shown in Fig. 7) to all other coordinates.

3) Constructing Path: In this critical step, our objective is to construct the path leading from the user's location to the coordinates identified in the preceding phase. To achieve this, we leverage the capabilities of Openrouteservices API, which efficiently processes origin and destination coordinates, returning the sequential path coordinates. Upon obtaining the path coordinates, we meticulously store them in the all-path coordinate array for subsequent analysis and utilization. To optimize computational efficiency for further processing, we apply a filtering mechanism to retain only those coordinates within a 10-meter distance threshold. This strategic filtration helps mitigate computational costs without compromising the integrity of the subsequent analyses.

Figure 8 visually represents the resulting paths constructed from the user's location to the various coordinates identified in Figure 7. It's important to note that due to the limited color palette, the paths may not be distinctly visible. However, the significance of this step lies in laying down the groundwork for further refinement and analysis in our quest to determine the safest route amidst potential flood hazards.

4) Creating Final Dataset For Training: In the subsequent stage of our project, we augment the location data with rainfall information to construct the final dataset for training our machine learning models. Integrating rainfall data into our dataset is crucial for accurately predicting flood risk at each coordinate along the paths. Traditionally, time series models such as LSTM (Long Short-Term Memory) have been employed to predict rainfall at individual coordinates. However, these models often require extensive historical data and are computationally expensive due to their reliance on factors like temperature, humidity, etc.

To circumvent these limitations, I've adopted a proactive approach by leveraging multiple APIs like Open Weather API, OpenmeteoAPI, and meteomatics API. These APIs utilize optimized techniques, incorporating satellite data and various environmental factors to predict rainfall accurately over 5 days. Using a skewed factor of 0.7, the predicted rainfall is then distributed across all coordinates along the paths. This accounts for the spatial distribution of rainfall, considering that rainfall typically covers a certain area rather than being localized to individual points. The resulting rainfall data is seamlessly integrated into the dataset obtained from the previous steps. This enriched dataset will serve as input for our machine-learning models, empowering them to predict rainfall intensity at the user's location with greater accuracy. By leveraging this proactive approach, we aim to enhance the reliability and efficiency of our flood risk prediction system, ensuring robustness in the face of potential inundation events.

.

5) Machine learning models: In this step, we will be using machine learning models to predict flood intensity. we will use basic regressors and progress towards advanced Boosting models. The choice of models is pivotal in ensuring accurate predictions and robust performance metrics. We begin with foundational regression models such as Linear Regression and Decision Tree Regression. These models serve as the building blocks, offering simplicity and interpretability. Despite their simplicity, they provide valuable insights into the relationships between predictors and flood intensity.

Moving forward, we explore more sophisticated algorithms including Random Forest Regression and Gradient Boosting Regression. These ensemble techniques harness the collective intelligence of multiple decision trees, enabling them to capture complex patterns and interactions within the data. Their ability to handle non-linear relationships and mitigate overfitting makes them invaluable assets in our predictive modeling arsenal.

To evaluate the performance of these models, we employ a range of metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2) score. These metrics offer comprehensive insights into the accuracy, precision, and goodness-of-fit of our models. By systematically comparing the performance of each model against these metrics, we gain a nuanced understanding of their strengths and weaknesses.

a) Linear Regressor: Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data. The Fig. 9 shows how the linear regressor works. I Applied Applied Grid search to the linear regressor to find the best parameters for prediction. The best parameters across ALL searched params: {'learning_rate': 0.01, 'max_depth': 8, 'n_estimators': 100, 'subsample': 0.1} Then I used these Best parameter for prediction. And achieved :

Mean Absolute Error (MAE): 5.082, Mean Squared Error (MSE): 79.5595, and Root Mean Squared Error (RMSE): 8.9196. These are not the desired result.

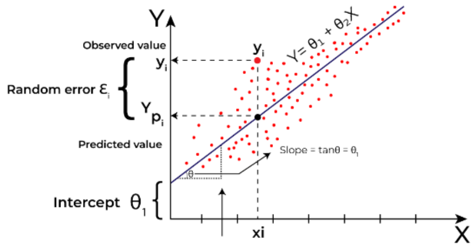


Fig 9. Linear regressor

b) Random Forest Regressor: To improve the above accuracy I used the Random Forest Regressor. Random Forest Regression is a machine learning ensemble technique adept at handling both regression and classification tasks. It achieves this through the utilization of numerous decision trees. Instead of relying solely on individual decision trees, Random Forest combines the outputs of multiple trees to arrive at the final prediction. This technique employs multiple decision trees as its base learning models. Through random sampling of rows and features from the dataset, separate sample datasets are formed for each model. This process ensures diversity among the trees and enhances the robustness of the overall model. Fig. 10 shows how the random forest works.

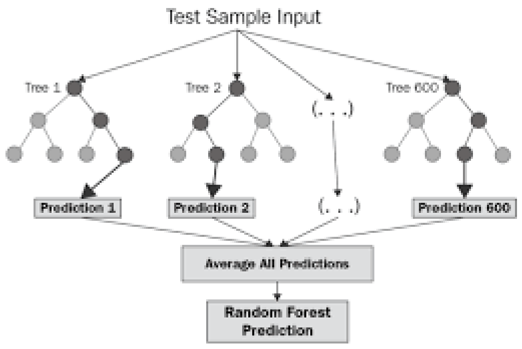


Fig.10 Random Forest.

The performance metric of random forest is:

- *Mean Squared Error: 0.6620958333333332*
- *R-squared: 0.9053199543441062*

From the above result, it can be observed that the Random Forest has performed better than the linear regressor.

c) Gradient Boosting Regressor(GBR): Gradient boosting is One powerful boosting approach that combines many weak learners into strong ones is gradient boosting. Every succeeding model in the ensemble is trained using gradient descent to minimize the loss function of the previous model, such as mean squared error or cross-entropy. The algorithm computes the gradient of the loss function with respect to the current ensemble's predictions at each iteration. A fresh weak model is then trained to reduce this gradient. After that, the predictions made by this new model are added to the ensemble, and the process continues until a predetermined stopping threshold is met. I Applied Applied Grid search to the linear regressor to find the best parameters for prediction and calculated the performance using these parameters. Fig. 11 shows the visualization of GBR.

- The best score across ALL searched params: 0.8811252026423153.
- The best parameters across ALL searched params: {'learning_rate': 0.03, 'max_depth': 6, 'n_estimators':1500, 'subsample': 0.5}
- Mean Absolute Error (MAE): 0.1954243177817295
- Mean Squared Error (MSE): 0.2843280860168371
- Root Mean Squared Error (RMSE): 0.533
- R-squared (R^2): 0.9653820592098921

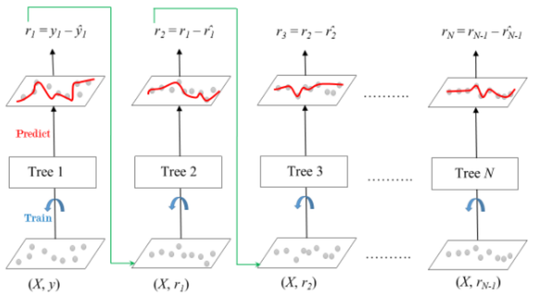


Fig. 11 visulaization of working GBR.

We can see from the above performance the GBR has performed better than the models. It has the minimum MAE, RMSE, and MSE. And best R-squared score. We will use this model for prediction.

d) Performance Overplot: After analyzing the best model. Prediction results are now visualized in this step.

i) *Depth vs Actual and predicted intensity:*

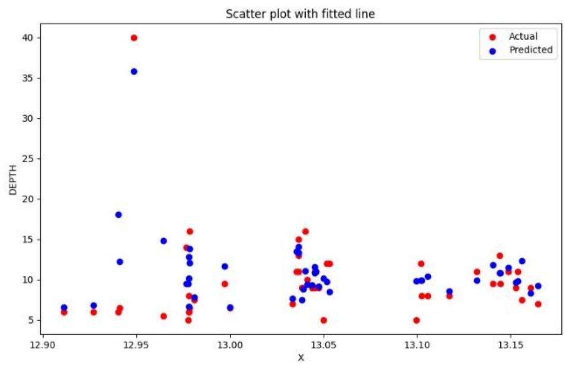


Fig. 12 Scatter plot of Depth vs actual and predicted intensity of flood.

Figure 12 represents the scatterplot of depth vs the actual and predicted values. From the figure we can observe that the machine learning models have performed well. We can see the predicted values are approximately around the actual intensity. We can also see that there are some outliers still the predicted values are around it.

ii) *Actual vs Predicted intensities:*

We have fitted a line to visualize both the predicted and actual intensity. We can observe from Figure 13, that that the predicted depth is generally higher than the actual depth. This is because the majority of the data points fall below the diagonal line, which represents a perfect fit where the predicted and actual depths would be the same. There seems to be a bias towards overprediction. The fitted line slants above the perfect fit diagonal, indicate a tendency of the model to predict higher depths than what was actually measured. The data points are somewhat scattered around the fitted line, suggesting some variability in the accuracy of the predictions. There are data points above and below the fitted line, which means some predictions were overestimates and others were underestimates. The overall model is doing a decent job.

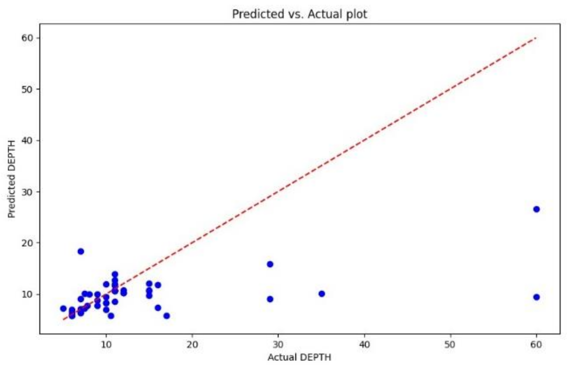


Fig. 13 Scatterplot with fitted line for Actual and predicted intensity.

6) Optimization for choosing the Route: Now, as we move forward from predicting the intensity at each coordinate of all paths, our focus shifts to determining the optimal or safest path. To achieve this, we utilize the intensity values along with the coordinates of each path. A dedicated function has been devised for calculating the maximum intensity of a path. This step is crucial as it helps identify potential flood risks along each path. The rationale behind calculating the maximum intensity lies in ensuring that even if some coordinates along a path exhibit lower intensities, the presence of a single high-intensity coordinate could still pose a significant flood risk. By evaluating the maximum intensity along each path, we gain insights into the potential severity of flooding. Subsequently, we proceed to identify the minimum intensity among the maximum intensities of all paths. This allows us to pinpoint the path with the least overall flood risk. We meticulously store the index corresponding to this minimum intensity path for subsequent retrieval.

Finally, leveraging this optimized approach, we return the safest path among all the paths constructed in the preceding steps. This crucial stage serves to streamline our efforts, ensuring that we navigate towards the safest route amidst potential flood hazards.

7) Final step: After optimization, we will get the optimized path, and now will show the optimized path from the user location.

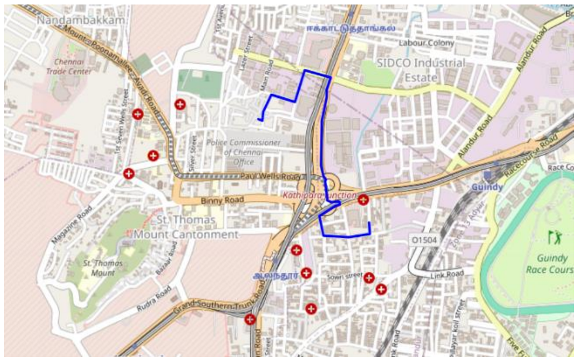


Fig. 14 safest route for Figure 7,8.

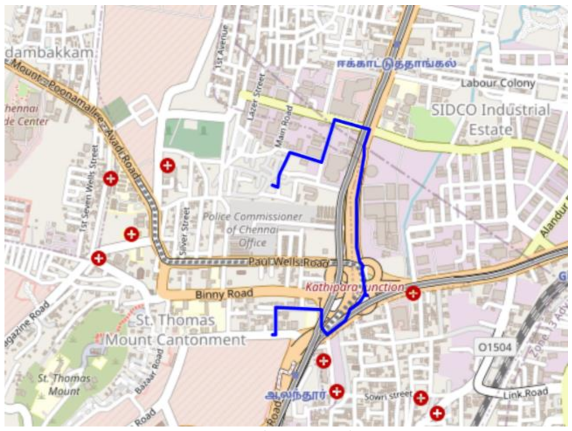


Fig 15 Another safest route for Figure 7,8.

Figure 14,15 shows the final optimized route from the user location shown in Figure 7,8. Both the optimized paths are almost the same, there is only a slight variation which is due to the slight variation in the intensity. Both these results are obtained using the rainfall data of 11 April 2024. Now I will use the actual rainfall of 2015 which was during the flood.

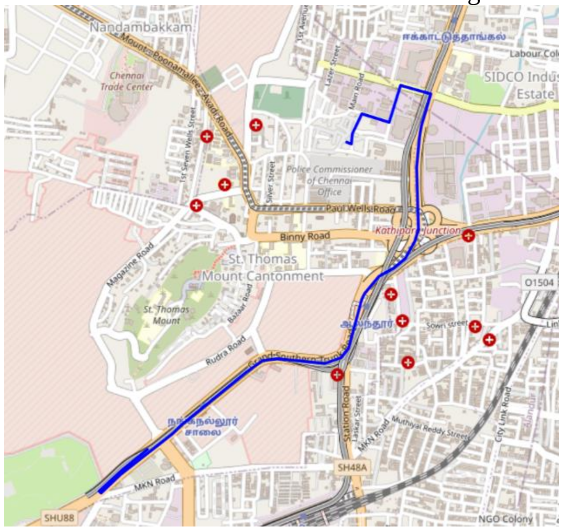


Fig.16 Another safest route for Figure 7,8.

Figure 16 shows the optimized path for the location shown in Figure 7,8. For this figure, I used the actual rainfall of 1011mm of Chennai November in 2015.

Figure 17 shows the optimized path for location shown in Figure 6.

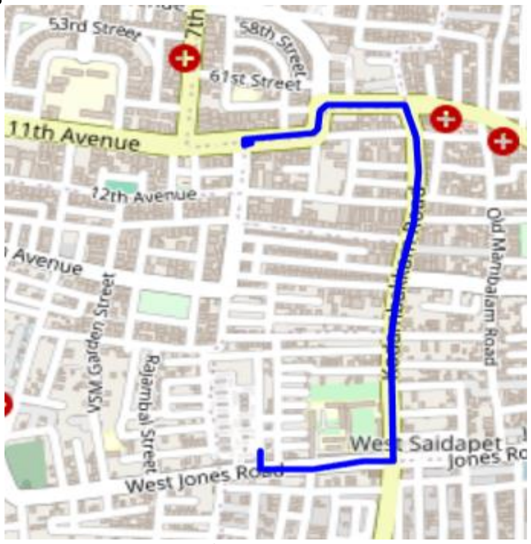


Fig. 17 Optimized path for Figure 6.

IV. RESULT

The results of the study demonstrate the effectiveness of the proposed ML-powered evacuation management system in enhancing flood preparedness and resilience, particularly in urban environments like Chennai. By leveraging historical data and advanced algorithms, the system successfully predicts flood trajectories and identifies optimal evacuation routes, thereby mitigating the risks associated with inundation events. the system lays the groundwork for proactive disaster preparedness measures and reinforces the resilience of at-risk communities.

Overall, the results highlight the transformative potential of ML-driven technologies in addressing complex challenges such as flood disasters, underscoring the importance of continued innovation and collaboration in safeguarding lives and livelihoods in the face of environmental threats.

V. CONCLUSION

In conclusion, this study underscores the critical importance of innovative approaches in addressing the pervasive threat of flood disasters in India, particularly in urban areas like Chennai. By elucidating the multifaceted impacts of floods and the shortcomings of traditional evacuation strategies, the study highlights the urgent need for adaptive and resilient solutions.

The proposed ML-powered evacuation management system emerges as a pioneering solution to this pressing challenge. Through the integration of historical data, advanced algorithms, and real-time information, the system demonstrates its capacity to predict flood trajectories, identify optimal evacuation routes, and enhance disaster preparedness and response efforts.

Furthermore, the scalability and adaptability of the system render it not only applicable to Chennai but also transferrable to other urban centers grappling with similar flood-related challenges across the country.

In essence, this study advocates for the adoption of AI-driven technologies as indispensable tools in enhancing disaster resilience and protecting lives and livelihoods in the face of rising waters. Through collaborative efforts and continued innovation, we can forge a path towards a safer and more resilient future for all.
Although, this system has some limitations

- *Slow execution*: Increasing the range and decreasing the elevation range may result in slow execution.
- *API dependent:* Most of the APIs I used are free, However, the meteomatics is not free or available for a free trial of one month only to get rainfall Data. other weather APIs have rate limits. However, it is better since we are getting the most accurate data from APIs.
- *Unavailability of route*: we won't be able to find some routes since they are not available in Google Maps or OpenStreetMap.

References

1) *https://www.chennaifloodsdss.in/HomePage/DownloadImagesFromAlert/1-%20Bulletin%201_0_Observed%20Rainfall%20-%2005-12-2022?fname=5122022123478301-%20Bulletin%201.0_Observed%20Rainfall%20-%2005-12-2022.pdf*

2) *https://github.com/hydrosenselab/India-Flood-Inventory*