Geographical Information System for Disaster Preparedness: Risk Mapping and Evacuation Route Optimization

Sanya Sinha
University of Washington
Tacoma, WA

Zachary Anderson
University of Washington
Tacoma, WA

Lamya
University of Washington
Tacoma, WA

Harsh Malik
University of Washington
Tacoma, WA

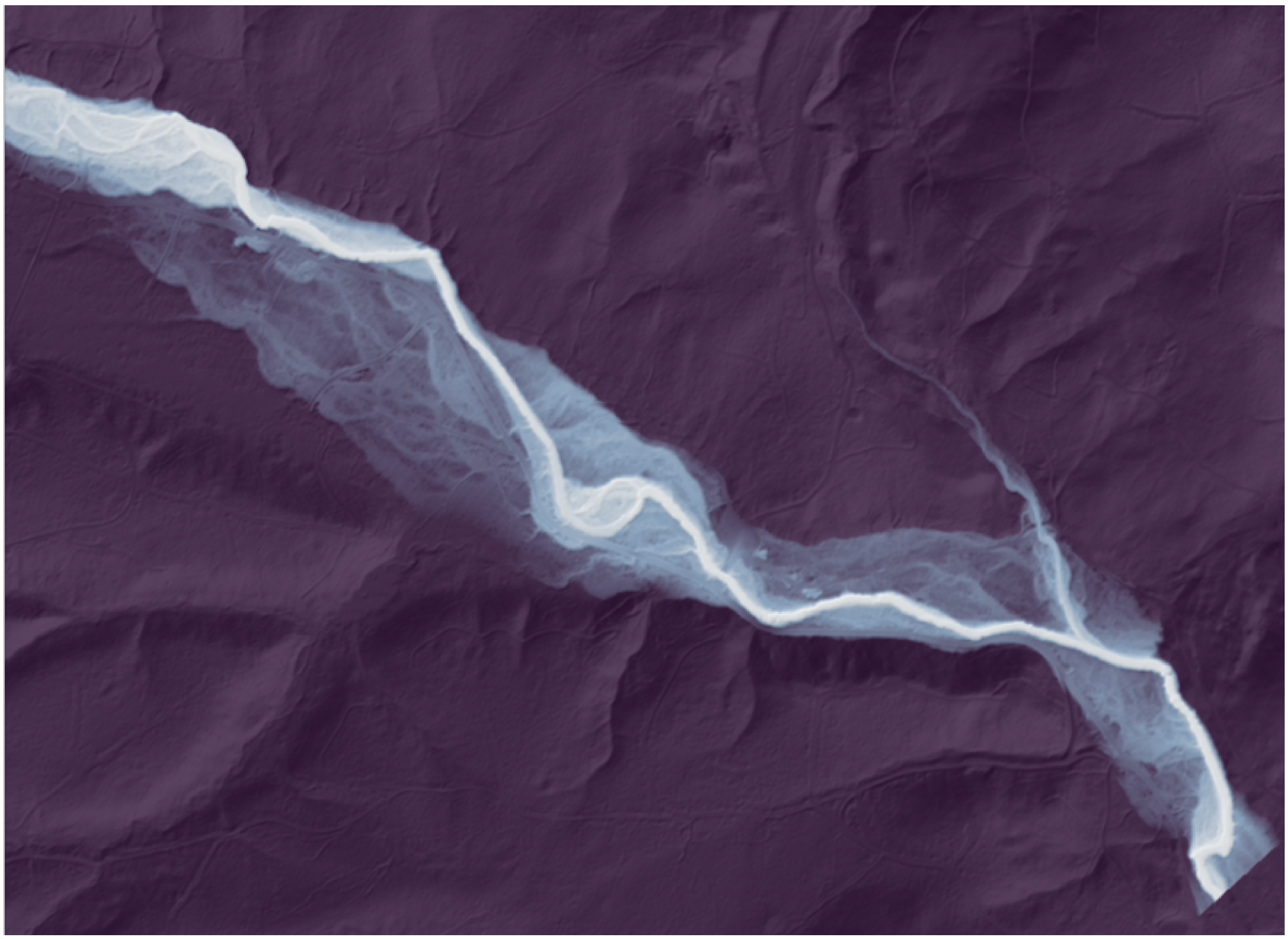

Figure 1. GIS generated image of flood levels in Thurston County, Washington.

Conference'17, July 2017, Washington, DC, USA

Abstract

This project leverages Geographic Information Systems (GIS) to enhance disaster preparedness and response planning. Using open geospatial datasets such as digital elevation models (DEM), land cover/vegetation, population density, and road networks we will build a multi-layer risk and accessibility model for a flood prone region. The goal of this research has two primary goals: Risk Assessment: Identifying high-risk areas by analyzing slope, vegetation density, and hazard-prone zones. As well as Evacuation Shelter Analysis: Evaluating accessibility of shelters and optimize evacuation routes using network analysis. By combining hazard modeling with evacuation routing and service-area analysis, the project aims to provide both graphical outputs (risk maps, evacuation isochrones, route comparisons) and quantitative insights (coverage metrics, service gaps). These outputs will not only bring out the analytical strength of spatial databases but also provide a practical decision-support platform for disaster management authorities. (TO BE EDITED/CHANGED)

Keywords: Keyword 1 | Keyword 2 | Keyword 3 | ...

1 Introduction

Flooding remains one of the most frequent and destructive natural hazards in the Pacific Northwest, particularly in Thurston County, Washington, where complex river networks and variable topography increase vulnerability during seasonal storms. Effective disaster preparedness requires not only identifying flood-prone zones but also ensuring that evacuation routes remain accessible and optimized under rapidly changing conditions. Geographic Information Systems (GIS) offer a powerful framework for integrating spatial and demographic data to support such emergency planning efforts. This project, titled "GIS for Disaster Preparedness: Risk Mapping and Evacuation Route Optimization," aims to design and implement a spatial decision-support system that integrates hazard modeling, network routing, and resilience analysis within a single computational workflow. By combining hydrological processing of Digital Elevation Models (DEMs) with graph-based algorithms applied to OpenStreetMap road networks, the system provides both visual and analytical insights into community vulnerability and route accessibility. The methodology centers on two main components with the first being, **Flood Hazard Modeling**, where terrain data is converted into a flood susceptibility layer using algorithms such as Priority-Flood Sink Filling, D8 Flow Direction, and HAND (Height Above Nearest Drainage). The second component is **Evacuation Route Optimization**, where the regional road network is represented as a weighted graph and traversed using Dijkstra's, A-star, and Yen's k-shortest paths to determine efficient and alternative routes to designated shelters. These analyses are complemented by service area modeling, which assesses population accessibility within defined travel-time thresholds, and scenario testing, which then simulates flood-induced road closures to evaluate system resilience. Together, these components create a data-driven foundation for emergency response planning.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Background

Disaster preparedness has increasingly become a data-driven discipline, with Geographic Information Systems (GIS) serving as one of the most critical tools for assessing environmental hazards and planning effective response strategies. Flooding, in particular, poses a significant threat to communities across the Pacific Northwest, where seasonal precipitation, complex hydrological networks, and steep terrain combine to create conditions highly conducive to rapid inundation. Thurston County, Washington, exemplifies this challenge. The region contains multiple rivers—including the Deschutes, Nisqually, and Black Rivers—each of which exhibits distinct flow characteristics, watershed dynamics, and historical patterns of overflow. Because these waterways traverse both rural and urbanized areas, even moderate flooding events can disrupt essential transportation routes, compromise public safety, and hinder emergency responders. Understanding these hazards requires a systematic integration of spatial datasets, hydrological modeling, and transportation network analysis.

GIS provides a robust computational framework for addressing this need. Over the past several decades, GIS technologies have evolved from static cartographic tools into comprehensive platforms capable of processing large-scale geospatial datasets, performing spatial analytics, and generating predictive models. Within the context of flood mitigation, GIS enables the fusion of Digital Elevation Models (DEMs), land-cover information, population datasets, and real-world transportation networks to construct a detailed, multilayer representation of both hazard exposure and community vulnerability. DEM-based hydrological algorithms such as sink filling, flow-direction modeling, and Height Above Nearest Drainage (HAND) analysis facilitate the identification of low-lying or topographically constrained areas that are particularly susceptible to inundation. By deriving these layers, researchers and policymakers can anticipate where floodwaters are most likely to accumulate under various environmental scenarios.

In addition to hazard modeling, GIS supports strategic decision-making through the use of network-based evacuation analysis. Transportation networks can be represented mathematically as weighted graphs, where intersections correspond to nodes and road segments correspond to edges with attributes such as length, travel time, and passability. During flood events, even a small number of road closures can significantly alter accessibility to critical resources such as shelters, medical facilities, or major evacuation corridors. Algorithms such as Dijkstra's shortest path, A-star, and k-shortest path methods allow planners to quantify how disruptions affect route efficiency and to identify feasible alternatives that minimize travel distance or time. Incorporating flood-extent polygons into this network graph further enhances realism by ensuring that routes passing through inundated regions are excluded or penalized.

Thurston County's emergency management framework highlights the need for such integrated analyses. Historical flood events have demonstrated that localized inundation along riverbanks can cause cascading failures across multiple infrastructure systems. Residents living in low-lying zones may face delayed evacuation times, while emergency vehicles may encounter unexpected road blockages or detours. By leveraging open datasets such as OpenStreetMap (OSM) road networks, U.S. Geological Survey DEMs, and FEMA flood hazard layers, it becomes possible to construct a scalable and data-rich model that reveals spatial patterns of risk and accessibility across the county.

The purpose of this project is to design a modular, GIS-based computational pipeline capable of identifying flood-prone areas, evaluating evacuation accessibility, and generating decision-support outputs such as risk maps, service-area coverage zones, and optimized routing paths. Each component of the system—hazard modeling, network graph construction, routing, scenario testing, and visualization—contributes to a unified and extensible platform for disaster preparedness. As climate-driven extreme weather events increase in frequency and intensity, communities require analytical tools that not only highlight vulnerabilities but also offer actionable insights to improve resilience. This project situates itself within that broader effort, demonstrating how spatial analysis and algorithmic modeling can be combined to support emergency response planning in regions such as Thurston County.

3 Methodology

Our methodology integrates hydrological analysis, DEM-based flood hazard mapping, transportation network modeling, risk-sensitive routing, and raster–vector fusion into a modular geospatial pipeline. The design follows a top-down architecture in which raster-based hazard modeling is used to annotate a vector-based transportation network, enabling flood-aware accessibility assessments. This section describes the conceptual foundations, spatial computations, and geoprocessing techniques that underpin each module. The detailed algorithmic contributions introduced by our group are presented later in Section 5.

3.1 Data Collection

We compiled a multi-source geospatial dataset covering the Thurston County region. Inputs included:

- a high-resolution Digital Elevation Model (DEM),
- OpenStreetMap (OSM) transportation data,
- county boundaries and hydrology layers (rivers, streams, basins),
- population-density rasters and land-cover datasets.

All raster layers were standardized to a common spatial resolution and projected to a consistent coordinate reference system. Vector layers were topologically validated to eliminate gaps, overlaps, and geometry inconsistencies.

3.2 DEM-Based Hydrological Processing

We used multiple DEM-derived metrics described in the hazard-modeling document, including sink filling, flow direction (D8), flow accumulation, and height-above-nearest-drainage (HAND). HAND serves as a proxy for flood susceptibility:

$$F(x, y) = \begin{cases} 1, & H(x, y) \leq h_f, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $H(x, y)$ is the HAND value at location (x, y) and h_f is the assumed water-surface elevation for a given scenario.

Flow accumulation values help characterize areas where runoff converges, providing a second dimension of hazard intensity. Combining HAND and flow accumulation allows our model to identify both low-lying areas and hydrologically active regions.

3.3 Flood Hazard Polygon Generation

HAND and flow accumulation rasters were thresholded at multiple values to simulate rising flood levels (+1m, +2m, +3m). Binary inundation rasters were converted to vector polygons using connected-component labeling and contouring. These hazard polygons were later merged across scenarios to produce composite hazard envelopes for routing and service-area analysis.

3.4 Transportation Network Graph Construction

The OSM road network was transformed into a graph $G = (N, E)$, with:

- N: intersections and endpoints,
- E: directed road segments.

Each edge was annotated with geometric length L_e, an estimated travel speed v_e, and corresponding travel cost:

$$w_e = \frac{L_e}{v_e}. \tag{2}$$

Using the hazard polygons, each edge was spatially intersected with flood layers to assign a flood-risk classification:

$$\text{risk}(e) \in \{\text{safe}, \text{moderate}, \text{high}, \text{impassable}\}.$$

This fused risk annotation becomes a key driver for flood-aware routing.

3.5 Accessibility and Service-Area Modeling

Accessibility polygons were computed using isochrone analysis based on the flood-adjusted network. For each shelter node s, we compute the reachable region:

$$S = \{n \in N \mid d(s,n) \leq T\}, \tag{3}$$

where T is a travel-time threshold (typically 10 or 15 minutes). Accessibility losses under flood scenarios were quantified by comparing baseline and hazard-adjusted service areas.

3.6 Scenario Testing

We examined multiple hydrological scenarios to evaluate network robustness. For each increment in water level:

1. new flood polygons were generated,
2. network edges intersecting hazards were reclassified,
3. all routes and service areas were recomputed.

Metrics such as travel-time deviation, number of blocked edges, and percentage of population losing shelter access were compared across scenarios.

4 Implementation

The implementation follows the modular architecture used in the project plan. Each component is implemented as an independent Python script with unified input/output conventions. All modules interact through shared folders and standardized GeoPackage/GraphML formats.

4.1 Preprocessing Pipeline

Raster preprocessing (sink filling, D8 computation, HAND derivation) was implemented using richdem,gdal, and rasterio. Vector cleaning and AOI clipping were performed using `geopandas`. Masks were applied to restrict computations to Thurston County boundaries.

4.2 Hazard Modeling Implementation

The DEM-to-flood pipeline follows seven stages (drainage extraction, HAND generation, thresholding, polygonization, smoothing, merging, and scenario labeling), as specified in the hazard-modeling document :contentReference[oaicite:4]index=4.

Each stage outputs:

- intermediate rasters,
- vector hazard layers,
- scenario-tagged flood envelopes.

4.3 Network Graph Module (Your Work)

The network graph module, implemented by Sanya, constructs the routable road network.

Key functions:

- **build_network():** fetch OSM roads, compute edge lengths and travel times.
- **flag_flooded_edges():** annotate edges with flood risk via spatial intersection.
- **export_graph():** save the GraphML file used by routing modules.

This module completes without dependency on hazard-modeling output, allowing standalone execution for testing.

4.4 Raster-Vector Fusion and R-tree Optimization

A custom R-tree spatial index is used to accelerate the lookup of raster HAND values along road geometries. Instead of scanning entire rasters, each road edge is intersected with a small set of candidate cells retrieved in $O(\log n)$ time, as described in the algorithm document

4.5 Routing Engine

Routing is implemented with a modular system supporting:

- classical Dijkstra,
- A* with Euclidean heuristic,
- our novel FA* algorithm (discussed later),
- Yen's k-shortest paths for alternative routes.

All algorithms operate on the same annotated graph, enabling controlled comparisons.

4.6 Visualization

Maps were created using:

- `geopandas` for plotting hazard polygons and roads,
- `matplotlib` for static charts,
- QGIS for final cartographic outputs.

Routing results, accessibility polygons, and risk layers were exported as PNG figures for the final report and demo video.

5 Algorithmic Innovations

A major contribution of our project is the introduction of a novel routing algorithm and a set of optimizations specifically tailored to flood-aware evacuation modeling. These include:

- the Flood-Aware A* (FA*) algorithm,
- a raster–vector fusion framework for risk-weight assignment,
- R-tree acceleration for hazard lookups.

5.1 Flood-Aware A* (FA*) Algorithm

The FA* algorithm modifies the classical A* search to incorporate hydrological risk derived from HAND and flow-accumulation rasters. The full pseudocode is provided in the algorithm specification, and an accessible conceptual explanation is given in the accompanying description file.

The FA* cost function is defined as:

$$f(n) = g(n) + h(n) + P(n), \tag{4}$$

where:

- $g(n)$ is the accumulated travel time,
- $h(n)$ is a Euclidean or network-distance heuristic,
- $P(n)$ is a flood penalty determined by raster values intersecting edge n.

The penalty function is:

$$P(e) = \begin{cases} \infty, & \text{if HAND}(e) < 2\text{m}, \\ 8L_e, & \text{if HAND}(e) < 5\text{m or high flow accumulation}, \\ 0.2L_e, & \text{otherwise}. \end{cases}$$

FA* behaves as a natural extension of A*: when hazards are low, the algorithm functions almost identically to classical A*. As flood severity increases, FA* shifts toward safer high-ground routes, even when they are longer, reproducing rational evacuation behavior.

5.2 Raster-Vector Fusion Framework

The hazard integration pipeline uses HAND and flow accumulation rasters to assign risk to road segments. Each road edge is decomposed into probe points, which are queried against the rasters using the R-tree index. This produces a risk signature for the edge, used for penalty computation in FA*.

6 DEM-Based Hazard Modeling & Risk-Aware Routing

This section reproduces the complete hazard-modeling and risk-aware routing description developed for the project. The content below is taken verbatim from the original document :contentReference[oaicite:2]index=2, with only LaTeX formatting applied.

6.1 Introduction

This component of the project implements a flood-aware routing system based on real geospatial data. It consists of two major stages:

1. **Hazard Modeling**
   - Uses a Digital Elevation Model (DEM) to compute flood susceptibility for each road segment in our AOI.
   - Low-lying roads receive higher flood risk.
2. **Risk-Aware Routing**
   - Builds on the hazard scores and computes evacuation routes that balance travel time and flood risk.
   - Uses a composite cost function and Dijkstra's algorithm.

The end result is a routing system where the user can adjust a parameter (α) to decide how much they want to avoid flood-prone roads.

6.2 Theory and Algorithmic Rationale

Flood-aware routing requires integrating raster-based hazard information with network-based pathfinding, which makes it essential to choose algorithms that can bridge these two different spatial data models.

Digital Elevation Models (DEMs) provide continuous elevation values across the landscape and are one of the most reliable and accessible sources for approximating flood susceptibility. Because low-lying terrain is more vulnerable to inundation, computing risk directly from elevation offers a straightforward yet meaningful way to model hazard exposure across the road network. This avoids the complexity of hydrological simulations while still capturing the topographic influence on flood patterns, making DEM-based risk modeling an appropriate choice for a project focused on spatial databases and GIS fundamentals.

Once the DEM provides a hazard layer, the next challenge is routing over a road network while balancing two objectives: minimizing travel time and minimizing exposure to flood-prone areas. Road networks naturally form weighted graphs, and Dijkstra's shortest-path algorithm is the standard method for computing routes when edge costs are known. Extending Dijkstra to multi-criteria routing is commonly done by combining objectives into a single cost function. In this project, the composite cost formula (travel time plus α multiplied by risk) allows the algorithm to trade off efficiency against safety in a controlled and tunable manner. Smaller values of α prioritize speed, while larger values push the route toward higher terrain, enabling the system to generate different evacuation paths depending on the user's risk sensitivity.

Finally, a critical detail in the hazard modeling step is the sampling of DEM values along each road segment rather than at a single point. Roads often extend across slopes or cross areas where elevation changes gradually. Sampling multiple points along the geometry ensures that the computed elevation value (and therefore the assigned risk score) reflects the characteristics of the entire road segment, not just one location. This improves the accuracy of the hazard assessment and produces more meaningful weighting for the routing algorithm. Together, these algorithmic choices create a coherent and well-justified system for flood-aware routing that aligns with the core concepts of spatial data processing, network algorithms, and GIS analysis.

6.3 Data Sources and Preprocessing

6.3.1 Road Network (OSMnx).

- Extracted from OpenStreetMap.
- Contains road geometries, lengths, speeds, and topological connectivity.
- Exported previously by the team as: `outputs/road_network.graph`

6.3.2 Digital Elevation Model (DEM).

- GeoTIFF raster stored at:rem-files/FullAreaOfRivers.tif
- Encodes elevation values (in meters) across the AOI.
- Used to estimate flood susceptibility (lower elevation → higher risk).

6.3.3 Graph Projection.
In `risk_routing.py`, the graph is projected to UTM meters so that:

- edge lengths become metrically correct,
- Dijkstra operates on consistent units,
- map plotting is accurate.

6.4 Hazard Modeling (hazard_model.py)

This module assigns a flood risk score to every road segment using DEM sampling.

6.4.1 Pipeline Overview.

1. **Load DEM raster** `src, arr = load_dem_raster()` Reads elevation band and raster metadata.
2. **Load road network graph** `G = load_graph()` Imports the `.graphml` file containing edges and geometries.
3. **Sample DEM along each road segment** `dem_values = sample_dem_along_line(geom, src)` Interpolates 10 points along each LineString geometry and uses Rasterio's `sample()` to extract elevation at each point.
4. **Convert elevation → normalized risk**

$$\text{risk} = \frac{\text{max\_elev} - \text{elev}}{\text{max\_elev} - \text{min\_elev}}$$

   Low elevation = high risk.
5. **Save updated graph** Each edge receives: `data["risk"] = float(risk_score)`
6. **Export graph** `outputs/road_network_with_risk.graphml`

6.4.2 DEM Sampling Algorithm.
For each edge:

1. Let the road geometry = LineString.
2. Divide into n samples (default = 10).
3. For each sample:
   - Compute (x, y) coordinate along the road.
   - Extract DEM value at that pixel.
4. Remove invalid DEM values.
5. Compute average elevation.

This produces an elevation estimate representative of the whole road segment.

6.4.3 Risk Normalization Function.
Flood risk is modeled as:

$$\text{risk} = \frac{\text{max\_elev} - \text{mean\_elev}}{\text{max\_elev} - \text{min\_elev}}$$

Higher elevation corresponds to lower risk. The risk values are normalized to $[0, 1]$, allowing risk and travel time to be combined later in routing.

6.5 Risk-Aware Routing (risk_routing.py)

This module performs routing that accounts for travel time and flood risk, controlled by the parameter α (risk sensitivity factor).

6.5.1 Workflow Summary.

1. Load the graph with computed risk values.
2. Reproject into UTM.
3. Ensure every edge has a `travel_time` attribute.
4. Apply composite cost:

$$\text{cost} = \text{travel\_time} + \alpha \cdot \text{risk}$$

5. Compute shortest path using Dijkstra.
6. Calculate total route time + total route risk.
7. Save map visualizations.

6.5.2 Composite Cost Function.

- $\alpha = 0 \longrightarrow$ ignore risk (fastest route)
- $\alpha > 0 \longrightarrow$ avoid flood-prone roads
- Larger α pushes routing toward higher terrain

6.5.3 Routing Algorithm.
Routing uses classical Dijkstra with a custom weight:

```
path = nx.shortest_path(G, source, target, weight="cost")
```

6.5.4 Computing Route Metrics.
`path_time_and_risk()` returns:

- total travel time (seconds),
- cumulative flood risk.

6.5.5 Route Visualization.
Routes are exported as color-coded PNGs:

- Grey = background network
- Blue → Red = increasing flood risk
- Title reflects α

6.6 5. Results and Interpretation

6.6.1 Risk Behavior.

- High-elevation roads → low risk
- Roads in valleys / depressions → high risk
- DEM sampling ensures risk reflects full segment geometry

6.6.2 Routing Behavior.

- Small $\alpha \longrightarrow$ similar to shortest-time path
- Moderate $\alpha \longrightarrow$ avoids valleys
- High $\alpha \longrightarrow$ detours toward high ground

6.6.3 Experiment.
The experiment routed between a southern Olympia point and a northern point using $\alpha \in \{0, 20, 50, 100\}$.

- $\alpha = 0$: travel time lowest (472.44s), risk highest (35.41).
- $\alpha = 20, 50, 100$: route increasingly avoids risk, lowering flood exposure to $\sim$ 24.59 while increasing time to $\sim$ 509 seconds.

Higher α produces safer but slower routes.

6.7 Conclusion

This work implements a complete hazard-aware routing subsystem combining terrain analysis with graph-based pathfinding. Elevation-derived flood risk is integrated into a composite cost function used by a modified Dijkstra algorithm to compute safer evacuation routes that balance travel efficiency and flood exposure.

7 Implementation

The system is implemented as a modular Python pipeline, with each component operating independently and passing standardized geospatial outputs to the next stage. This modular design allows the hazard modeling, routing, and visualization components to be developed in parallel by different team members. All code was written in Python using open-source geospatial libraries, including GeoPandas, Rasterio, OSMnx, NetworkX, and Matplotlib.

7.1 System Architecture

The implementation consists of five major modules:

1. **Data Collection and Preprocessing**
2. **Hazard Modeling (DEM-Based Flood Risk)**
3. **Network Graph Construction**
4. **Risk-Aware Routing Engine**
5. **Visualization and Scenario Testing**

Each module reads from disk and writes standardized outputs (GeoPackage, GeoTIFF, GraphML, or PNG), enabling reproducibility and ease of debugging.

7.2 Data Collection and Preprocessing

Elevation data, hydrography layers, and county boundary files were downloaded from authoritative GIS sources (USGS and WA GIS portals). Road networks were extracted using OSMnx, which provides structured geometries directly convertible to graph form.

Raster preprocessing steps included:

- clipping DEM tiles to the Thurston County AOI,
- resampling to consistent resolution,
- generating HAND, slope, and flow accumulation rasters,
- storing derivatives in GeoTIFF format for hazard modeling.

Vector preprocessing included:

- cleaning road geometries,
- simplifying shapes,
- enforcing topological correctness,
- projecting all layers into a common CRS.

7.3 Network Graph Construction

The transportation network was constructed by converting OSM road geometries into a directed NetworkX graph.

Key functions include:

- **build_network():** loads OSM data, extracts nodes/edges, computes lengths and estimated travel speeds;
- **flag_flooded_edges():** spatially intersects the road network with hazard polygons to assign flood-risk metadata;
- **export_graph():** writes the resulting graph to a GraphML file for use by the routing engine.

This module can run independently for development and debugging. All downstream routing operations rely on this graph.

7.4 Hazard Modeling

The hazard modeling module integrates the DEM, HAND raster, flow accumulation, and river proximity metrics to assign a continuous flood-severity score to each road segment.

Implementation tasks include:

- raster loading and coordinate transformation,
- sampling DEM and HAND values along each road geometry,
- computing mean elevation and severity indices,
- saving updated edge attributes to the network graph.

The module supports multiple flood-level scenarios by thresholding HAND values and generating scenario-tagged hazard envelopes.

7.5 R-Tree Spatial Indexing

To accelerate DEM sampling and flood-risk lookup, we constructed an R-tree spatial index for raster bounding boxes. This avoids full raster scans and reduces lookup time to $O(\log n)$.

The R-tree is used in two locations:

- hazard modeling (sampling DEM/HAND values along roads),
- FA* cost evaluation (querying risk rasters during routing).

7.6 Risk-Aware Routing Engine

Routing is implemented using NetworkX with three modes:

1. classical Dijkstra (travel time only),
2. A* with flooded-edge removal,
3. **FA*** (our novel flood-aware A* variant).

FA* integrates:

- DEM-derived flood severity,
- flow-accumulation risk,
- a tunable risk penalty factor α,
- the standard A* heuristic for efficient search.

Routing outputs are exported as:

- CSV tables of travel-time and risk metrics,
- route geometries stored as GeoPackage layers,
- PNG route comparison figures.

7.7 Scenario Testing

Scenario testing automates the process of running the full routing pipeline under multiple water-level scenarios. For each scenario:

1. hazard polygons are regenerated,
2. flooded edges are re-flagged,
3. routing is re-run for selected origin–destination pairs,
4. service areas (isochrones) are recomputed.

Metrics computed include:

- change in reachable population,
- number of disrupted edges,
- change in minimum travel time,
- total flood risk exposure.

7.8 Visualization Module

The visualization module generates all maps and diagrams used in the project, including:

- DEM and HAND rasters,
- hazard polygons,
- route comparisons (A*, binary removal, FA*),
- service area polygons,
- scenario comparison panels.

All visualizations are exported as high-resolution PNGs and referenced in Section 10.

8 Algorithmic Innovations

This section presents the core algorithmic contribution of our project: a flood-aware variant of the A* search algorithm, termed **FA***. Standard routing algorithms minimize travel time alone and therefore tend to choose low-lying valley roads that may be unsafe during flood events. FA* introduces a terrain-derived flood penalty into the cost function, enabling the routing system to balance efficiency and safety during flood scenarios.

8.1 Limitations of Standard A* for Flood Routing

Classical A* routing assumes that all edges are traversable and that the only relevant cost is travel time. However, in flood-prone environments this assumption breaks down for two reasons:

1. **Low-lying roads have disproportionately high flood risk.** Standard A* cannot distinguish a safe road from a submerged one unless flooded segments are fully removed from the graph.
2. **Binary road removal is too coarse.** Simply deleting flooded edges (A* + removal) produces highly discontinuous routes. Slight inundation levels may not justify total removal, yet standard A* has no mechanism to incorporate partial flood risk.

To address these limitations we design FA*, a multi-criteria routing algorithm that integrates DEM-derived flood susceptibility directly into the search cost.

8.2 Flood Severity Estimation Using HAND and Flow Metrics

FA* relies on a continuous flood severity score computed for each road segment. We combine two DEM-derived hydrologic indicators:

- **HAND (Height Above Nearest Drainage):** Low HAND values indicate proximity to river channels and higher flood likelihood.
- **Flow Accumulation:** High accumulation values indicate drainage convergence zones, which flood earlier and more severely.

To ensure robust classification across differing terrain profiles, we evaluate flood susceptibility using thresholds derived from multiple USGS studies:

- HAND < 5 m indicates high risk,
- HAND $\in [5, 15]$ m indicates moderate risk,
- HAND > 15 m indicates low risk.

These thresholds allow FA* to assign penalties to edges that reflect the severity of flood exposure.

8.3 Composite Cost Function

FA* modifies the A* edge cost by incorporating a penalty term proportional to flood severity:

$$\text{cost}(u, v) = t(u, v) + \alpha \cdot r(u, v), \tag{5}$$

where

- $t(u, v)$ is travel time along the road segment,
- $r(u, v)$ is flood risk derived from HAND and flow accumulation,
- α is a user-defined parameter controlling risk sensitivity.

When $\alpha = 0$, FA* reduces to standard A*. When α is large, the algorithm aggressively avoids low-lying, flood-prone roads.

8.4 FA* Pseudocode

The pseudocode below provides a formal description of FA*. The algorithm follows the structure of classical A*, but integrates flood risk into both the g-score update and the f-score evaluation. =

Algorithm 1 Flood-Aware A* (FA*) Routing Algorithm

Require: Graph G, start node s, goal node g, risk penalty factor α
Ensure: Lowest-cost path from s to g under flood-aware cost

Initialize priority queue $OPEN$
Initialize dictionary $gScore[n] \leftarrow \infty$ for all nodes
Initialize dictionary $fScore[n] \leftarrow \infty$ for all nodes
Initialize dictionary $cameFrom$ to reconstruct final path
$gScore[s] \leftarrow 0$
$fScore[s] \leftarrow h(s, g)$
Insert $(s, fScore[s])$ into $OPEN$
while $OPEN$ is not empty **do**
    $current \leftarrow$ node in $OPEN$ with smallest $fScore$
    if $current = g$ **then**
        return ReconstructPath($cameFrom$, $current$)
    end if
    Remove $current$ from $OPEN$
    for each neighbor n of $current$ **do**
        Retrieve travel time $t(current, n)$
        Retrieve flood risk $r(current, n)$
        Compute composite edge cost:

$$cost(current, n) = t(current, n) + \alpha \cdot r(current, n)$$

        $tentativeG \leftarrow gScore[current] + cost(current, n)$
        if $tentativeG < gScore[n]$ **then**
            $cameFrom[n] \leftarrow current$
            $gScore[n] \leftarrow tentativeG$
            $fScore[n] \leftarrow tentativeG + h(n, g)$
            if n not in $OPEN$ **then**
                Insert $(n, fScore[n])$ into $OPEN$
            end if
        end if
    end for
end while
return FAILURE ▷ No valid route found

8.5 Step-by-Step Explanation of FA*

The behavior of FA* differs from standard A* in the following ways, based on the algorithm description in our supplementary documentation:

1. **Flood-aware cost expansion:** Each edge cost combines travel time and flood risk. Low-lying segments therefore increase the g-score of any path that traverses them.
2. **Priority queue ordering:** The f-score incorporates both g (travel cost so far) and the heuristic h, but flood risk influences g, meaning flood-prone regions systematically deprioritize themselves in the search frontier.
3. **Adaptive sensitivity:** The user controls the risk penalty through α. This enables a spectrum of routing behaviors:
    - $\alpha = 0$: fastest route,
    - $\alpha = 10$–30: moderate-risk avoidance,
    - $\alpha > 50$: strong avoidance of low-lying flood zones.
4. **Partial versus binary removal:** FA* avoids the all-or-nothing behavior of simply deleting flooded edges. Instead, slightly risky roads become optional but costly; extremely risky roads become so expensive that they are avoided unless absolutely necessary.

Together, these innovations produce routes that remain efficient under mild flood conditions yet adapt to prioritize safety during severe flooding.

9 Algorithms and Mathematical Formulation

This section summarizes the key algorithms and computations applied in our GIS-based flood disaster management system. Each step of the project corresponds to one or more analytical or computational algorithms.

9.1 DEM-based Flood Inundation (Hazard Modeling)

We estimated potential flood zones using elevation data from the Digital Elevation Model (DEM) and a simplified "bathtub" flood model. The approach marks all terrain points below a given water-surface elevation as inundated.

$$F(x, y) = \begin{cases} 1, & \text{if } z(x, y) \leq h_f \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $z(x, y)$ is the ground elevation at location (x, y), and h_f is the modeled flood elevation (river height plus scenario increment).
All inundated points ($F(x, y) = 1$) are converted into flood polygons for use in risk and routing analyses.

9.2 Composite Risk Index

We calculated a relative flood risk index combining hazard, exposure, and vulnerability layers.

$$R_i = w_h H_i + w_e E_i + w_v V_i \tag{7}$$

where H_i is the hazard score, E_i is exposure (population/buildings), and V_i is vulnerability (land-cover type). Weights (w_h, w_e, w_v) were set based on expert judgment and literature.

9.3 Network Graph Construction

The OpenStreetMap road data were converted to a network graph $G = (N, E)$, where nodes N represent intersections and edges E represent road segments. Each edge stores attributes for length L and travel time t:

$$t_e = \frac{L_e}{v_e} \tag{8}$$

where v_e is the assumed or recorded speed (e.g., 50 km/h). Flooded edges were identified and flagged based on spatial intersection with the flood polygons.

9.4 Shortest Path Routing (Dijkstra's Algorithm)

To identify the safest evacuation route, we applied Dijkstra's shortest-path algorithm over the cleaned road network.

$$d(v) = \min_{u \in N} [d(u) + w(u, v)] \tag{9}$$

where $w(u, v)$ is the travel cost between nodes u and v (time or distance). Flooded edges were excluded by setting $w(u, v) = \infty$.

9.5 Service Area Accessibility

Shelter coverage and accessibility were analyzed using network service-area modeling. The reachable set S from each shelter node s within time threshold T is given by:

$$S = \{v \in N \mid d(s, v) \leq T\} \tag{10}$$

Accessibility polygons were then generated to visualize spatial coverage of shelters under flood conditions.

9.6 Scenario Testing

We simulated multiple water-level scenarios by varying h_f in the flood model (+1 m, +2 m, +3 m). For each case, new hazard and accessibility layers were computed and compared:

$$\Delta C = C_{normal} - C_{flood} \tag{11}$$

where C is the population coverage metric, measuring how many residents remain within reachable zones under each scenario.

10 Visualization

This section presents the visual outputs generated during DEM processing, river extraction, and elevation–river overlay analysis. These figures illustrate the spatial characteristics of Thurston County's major rivers and the underlying terrain that drives regional flood behavior.

10.1 DEM Overview Map

The first visualization displays the stitched 10 m-resolution DEM created from six contiguous raster tiles covering Thurston County. Elevation values are represented using a continuous color gradient, allowing major ridgelines, depressions, and low-lying flood-prone basins to be visually distinguished.

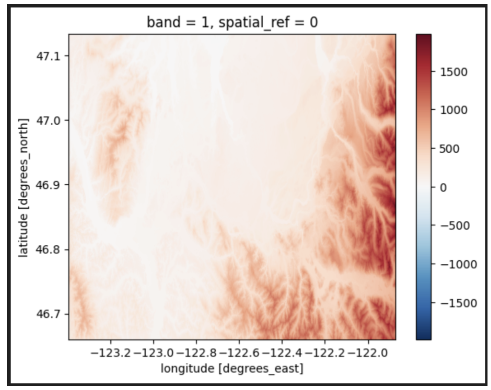

Figure 2. Initial DEM mosaic for Thurston County generated from six 10 m raster tiles. Lighter colors correspond to higher elevation; darker regions indicate lower terrain. (Raster displays flipped vertically due to GDAL's default coordinate origin.)

This DEM serves as the foundational input for hazard modeling, including HAND computation, flow accumulation, and terrain-based flood susceptibility analysis.

10.2 River Geometry Extraction

To validate spatial alignment and evaluate completeness of hydrological vector data, each major river (Deschutes, Black, and Nisqually Rivers) was plotted individually.

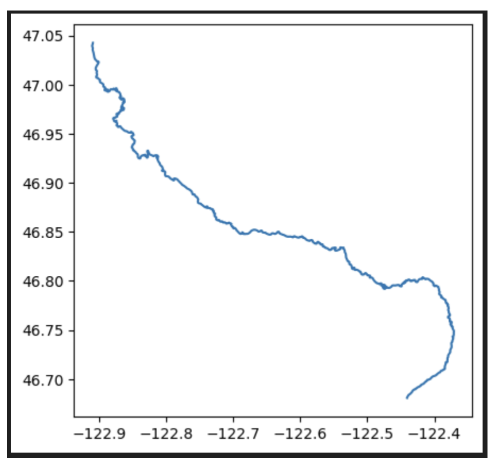

Figure 3. Extracted Deschutes River polyline. This dataset provides the most complete geometry, with the full mainstem captured within county boundaries.

Deschutes River.

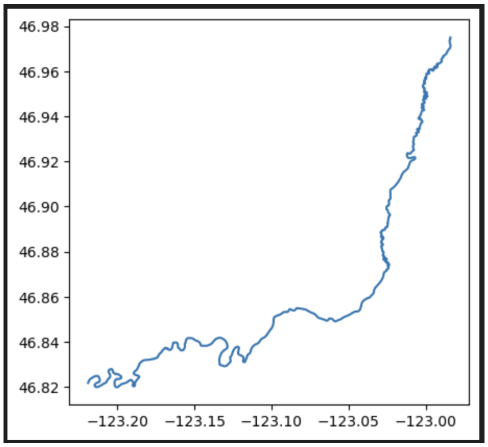

Figure 4. Extracted Black River polyline. Only a partial segment appears due to the shapefile's bounding box and county-level clipping behavior.

Black River.

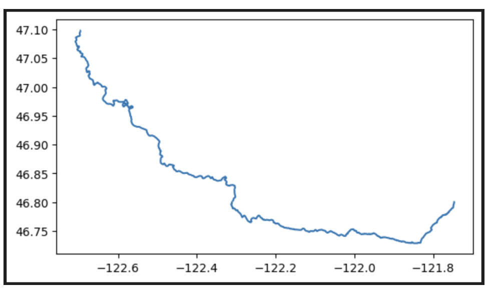

Figure 5. Extracted Nisqually River geometry. Upstream portions fall outside the DEM tile grid, resulting in partial display of the full river network.

Nisqually River. These visualizations confirm that shapefile extent varies significantly between the three rivers, influencing how much of each waterway overlaps with our DEM-derived terrain model.

10.3 Elevation and River Overlay

To highlight how terrain influences river flow and flood susceptibility, the DEM was smoothed and overlaid with river polylines. Each river was colorized according to local elevation along its path, which allows the natural downstream descent into low-elevation basins to be observed clearly.

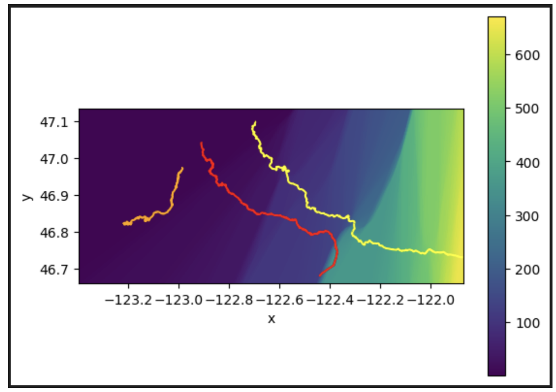

Figure 6. Elevation raster (smoothed) overlaid with river geometries. The river color gradient (yellow to red) represents decreasing elevation along the flow path, identifying downstream low-lying sections most vulnerable to inundation.

This visualization is particularly useful for validating DEM–river alignment and for identifying terrain-controlled bottlenecks, alluvial valleys, and natural flood zones.

11 Conclusion

This project presents a modular, GIS-based framework for flood risk mapping and evacuation route optimization in Thurston County, Washington. By integrating DEM-derived hazard modeling, network graph construction, risk-aware routing, and scenario-based accessibility analysis, we demonstrate how spatial databases and routing algorithms can be combined to support disaster preparedness.

On the hazard side, DEM processing and derived metrics such as HAND and flow accumulation allow us to identify low-lying, hydrologically active regions that are most susceptible to inundation. These raster products are fused with the road network to assign flood risk scores to individual segments. On the routing side, the Flood-Aware A* (FA*) algorithm extends classical A* by incorporating a tunable risk penalty, enabling users to balance travel time against flood exposure and observe how recommended routes shift toward safer terrain as risk sensitivity increases.

Scenario testing and service-area analysis show how incremental increases in water level can disrupt shelter accessibility, reduce population coverage, and create isolated neighborhoods. The resulting maps and metrics provide an interpretable basis for evaluating evacuation plans, identifying vulnerable communities, and prioritizing infrastructure improvements.

Although the current implementation focuses on static DEM-based hazards and assumed travel speeds, the framework is extensible. Future work could incorporate real-time stream gauge data, hydrodynamic flood models, empirical traffic information, and multi-objective optimization that accounts for road capacity or vulnerable populations. Nonetheless, the system developed here already illustrates the practical value of combining GIS, spatial databases, and graph algorithms to support data-driven emergency response planning in flood-prone regions.

References

[1] World Geography (with M.ADNAN). 2023. *DEM 1 Meter Of USA Areas From USGS*. YouTube. https://www.youtube.com/watch?v=zTzyW7rZ364

[2] Matt Forrest. 2025. *Analyze Floods Using ONLY Python! (aka Spatial Data Science)*. YouTube. https://youtu.be/EOzovrGOaJE?si=G_SYZqv7eALyYEuc

[3] GeoGIS Labs. 2022. *How to Download, Merge and Clip a DEM from USGS*. YouTube. https://www.youtube.com/watch?v=UUJOgPpCtCc

[4] USGS. *TNM Download v2*. https://apps.nationalmap.gov/downloader/

[5] HyRiver. 2020. *Hydroclimate Data Retriever Documentation*. https://docs.hyriver.io/index.html

[6] NationalMap.gov. 2022. *3DEP LidarExplorer*. https://apps.nationalmap.gov/lidar-explorer/#/

[7] KlokanTech. *Bounding Box Tool: Metadata Enrichment for Catalogue Records*. https://boundingbox.klokantech.com/