**Part 1: Network Analysis Deep Dive**

**Traffic Accident Network Analysis**

**1. Stakeholder Network Needs**

Understanding accident patterns through network visualization is crucial for multiple stakeholders:

A screenshot of a web page

AI-generated content may be incorrect.

* **How do stakeholders interact with network data?**
  + City planners and engineers need **geospatial network maps** to prioritize road changes.
  + Law enforcement requires **real-time network insights** to deploy patrols effectively.
  + Policy makers need **cause-based network analysis** to address systemic traffic risks.
  + The public benefits from **interactive visualizations** for route safety awareness.
* **How do networks help users gain insights?**
  + **Identifies accident-prone intersections** (hubs with high crash frequency).
  + **Links accident causes across locations** (understanding why crashes occur).
  + **Helps predict future accident risks** (by detecting patterns in historical data).

**2. Data Assessment**

**Network Representation**

The **traffic accident network** is best represented as a **graph**:

* **Nodes:** Accident locations (e.g., intersections, highways).
* **Edges:** Connections between locations based on shared accident causes.

**Node Attributes**

* **Reported\_Location** (Intersection or road name)
* **Latitude & Longitude** (Geospatial data)
* **Collision Type** (Single vs. multi-car crashes)
* **Injury Type** (Severity of accidents)

**Edge Attributes**

* **Connection type:** Accidents linked by shared causes (e.g., failure to yield, speeding).
* **Weight:** Number of accidents involving the same cause.
* **Directed/Undirected:** Likely **undirected**, as crashes are bidirectional events.

**Data Format**

* CSV dataset, processed into a **graph structure** using Python’s **NetworkX**.

**Network Metrics for Insights**

1. **Degree Centrality** – Identifies the most accident-prone locations.
2. **Betweenness Centrality** – Finds intersections that act as high-risk traffic points.
3. **Edge Weights** – Shows how frequently certain causes lead to crashes.
4. **Community Detection** – Groups accident types by similarity.

**3. Initial Design Exploration**

To analyze accident trends, we developed **two network visualization approaches**:

**1. Location-Based Traffic Network**

* **Nodes:** Accident locations (intersections, roadways).
* **Edges:** Connections between locations where accidents have similar causes.
* **Visual Representation:**
  + **Node size:** Proportional to accident frequency.
  + **Edge thickness:** Based on the number of linked accidents.
  + **Color coding:** Severity of accidents (fatal vs. minor injuries).
* **Stakeholder Use:** Helps planners & law enforcement target high-risk intersections.

A blue circle with white background

AI-generated content may be incorrect.

Here is the **first network visualization**, representing **traffic accident locations and their causes**. The nodes represent **accident locations**, while the edges indicate **accidents connected by similar causes**.

**Summary of Insights**

* + Highlights **accident-prone intersections**.
  + Useful for **urban planners** and **law enforcement** to identify high-risk areas.

**2. Cause-Based Accident Network**

* **Nodes:** Accident causes (e.g., "Failure to Yield", "Speeding").
* **Edges:** Connections between causes that frequently appear together at accidents.
* **Visual Representation:**
  + **Node size:** Frequency of cause occurrence.
  + **Edge weight:** How often two causes co-occur.
  + **Cluster analysis:** Groups common causes together.
* **Stakeholder Use:** Helps policymakers **develop safety regulations** by understanding recurring crash causes.

A close-up of a computer screen

AI-generated content may be incorrect.

**What This Visualization Shows**

1. **Nodes (Blue Circles):**
   * **Each node represents a unique accident cause (e.g., "Failure to Yield", "Speeding").**
   * **Larger nodes indicate causes that appear frequently in the dataset.**
2. **Edges (Gray Lines):**
   * **Edges connect accident causes that co-occur at the same accident location.**
   * **Thicker edges mean that the two causes frequently appear together.**
3. **Labeled High-Frequency Causes:**
   * **Only causes with high occurrence (more than 50 times) are labeled.**
   * **This avoids clutter and focuses on major contributing factors.**

* **Summary of Insights**
  + Shows relationships between common **causes of accidents**.
  + Useful for **safety officials** and **policy makers** to **target prevention strategies**.

**Design Rationale**

* **Why two different visualizations?**
  + The **location-based** network is useful for **spatial analysis** (where crashes happen).
  + The **cause-based** network focuses on **policy & prevention** (why crashes happen).
* **Why use a network instead of a basic heatmap?**
  + A network **shows relationships** between crash locations and causes.
  + A heatmap **only shows density** without revealing interconnected risks.

**Part 2: AI-Assisted Design Process**

**1. Documenting AI Interactions**

* **AI Model Used:** OpenAI’s **ChatGPT-4**, version **January 2024**.
* **Prompts Used (Verbatim Examples):**
  1. *"How can I visualize accident data as a network using NetworkX?"*
  2. *"What are the best layouts for a cause-based accident network?"*
  3. *"How can I adjust node sizes based on accident frequency in NetworkX?"*
  4. *"How do I filter and zoom into high-accident areas in a traffic network?"*
  5. *"What are some effective network metrics for analyzing crash causes?"*
* **Why These Prompt Structures?**
  1. Used **specific questions** to ensure **clear, actionable responses**.
  2. Structured prompts to **get code suggestions** and **best practices for network visualization**.
  3. Iteratively refined prompts based on initial AI responses.

**2. Implementation Plan**

**Why These Tools/Libraries?**

| **Tool/Library** | **Purpose** |
| --- | --- |
| **Pandas** | Data cleaning and processing of crash dataset (CSV format). |
| **NetworkX** | Building and analyzing the accident cause network. |
| **Matplotlib** | Visualization of the accident network. |
| **Seaborn** | For additional styling and visual clarity. |

**What Interactive Features Might Be Helpful?**

* **Hover tooltips** on nodes to display accident causes dynamically.
* **Clickable nodes** for filtering specific accident causes.
* **Sliders** to adjust threshold for filtering low-frequency causes.

**Data Preparation Steps**

1. **Load Dataset** → Read CSV into Pandas.
2. **Clean Data** → Remove missing values and inconsistencies.
3. **Extract Key Fields** → Use location, accident type, and cause data.
4. **Filter Data** → Select **high-frequency** causes to reduce clutter.
5. **Convert to Graph Structure** → Create **nodes (accident causes) and edges (co-occurrence relationships).**

**Data Analysis & Visualization Tools**

* **Pandas**: Data processing.
* **NetworkX**: Graph construction and network metrics.
* **Matplotlib**: Graph rendering.
* **Seaborn**: Improved color mapping.
* **Gephi (Optional)**: If a **highly interactive** visualization is needed.

**Handling Data Quality Issues**

* **Missing Data**: Removed incomplete records.
* **Duplicate Entries**: Dropped duplicate accident reports.
* **Outliers**: Removed unrealistic latitude/longitude points.
* **Data Consistency**: Standardized accident cause labels (e.g., merging "Failure to Yield" and "Failure to Yield Right of Way").

**3. Evaluation of AI Suggestions**

**Helpful AI Suggestions**

* Suggested filtering low-frequency accident causes to improve clarity.
* Recommended using the Kamada-Kawai layout to avoid the circular shape issue.
* Helped in scaling node sizes based on accident frequency.
* Guided on edge weight calculation to represent co-occurrence intensity.

**Limitations Encountered**

* AI initially suggested using a heatmap, which was not ideal for relationship-based data.
* Overestimated the effectiveness of the spring layout, causing overlapping nodes.
* Did not automatically suggest optimal node label filtering, requiring manual adjustments.

**Modifications & Improvements**

* Used Kamada-Kawai layout instead of the default spring layout to prevent circular clustering.
* Adjusted node size scaling to prevent oversized bubbles.
* Refined edge thickness calculation based on cause frequency instead of uniform thickness.
* Applied selective labeling for only high-frequency accident causes to avoid clutter.

**Visualization Best Practices Missed by AI**

* Color differentiation for cause categories, as AI did not suggest using distinct color groups for severity.
* Readability issues, as the initial AI-generated graphs had overlapping text labels.
* Edge weight optimization, as AI suggested using default values, but manual tweaking led to better balance.
* Dynamic filtering, as AI did not recommend interactive thresholding for zooming into relevant data.