Resilient Routes: Leveraging Deep Learning to Identify Critical Transportation Links for Humanitarian Response

by

Daniel Aguirre

Submitted to the Department of Systems and Computing Engineering
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS AND COMPUTING ENGINEERING

at the

UNIVERSIDAD DE LOS ANDES

June 2025

Authored by: Daniel Aguirre
Department of Systems and Computing Engineering
May 19, 2025

Certified by: Andrea Herrera Suescún
Associate Professor of Systems and Computing Engineering, Thesis Supervisor

Certified by: Nicolás Cardozo
Associate Professor of Systems and Computing Engineering, Thesis Supervisor

THESIS COMMITTEE

Thesis Supervisors

Andrea Herrera
Universidad de los Andes

Nicolás Cardozo
Universidad de los Andes

Thesis Readers

Ruben Manrique
Universidad de los Andes

Luis Macea
Pontificia Universidad Javeriana de Cali

Resilient Routes: Leveraging Deep Learning to Identify Critical Transportation Links for Humanitarian Response

by

Daniel Aguirre

Submitted to the Department of Systems and Computing Engineering
on May 19, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS AND COMPUTING ENGINEERING

ABSTRACT

This research presents a novel application of graph neural networks (GNN) for identifying critical links in transportation networks during disaster response operations. We simulated diverse transportation networks with varying topologies, weight patterns, and source-terminal configurations to construct a comprehensive training dataset. For each network, we calculated critical link scores using three established vulnerability indexes—Cantillo's index, global efficiency, and number of independent paths. Traditional approaches are computationally intensive, relying heavily on network size, supply/demand nodes, and disruption scenarios, making them impractical for large networks. Our GNN approach addresses this limitation by learning vulnerability patterns directly from network structures, eliminating the need for exhaustive scenario evaluation during inference. To validate the proposed approach, we applied it to both simulated and real-world transportation networks with diverse topological characteristics. We assess performance by sequentially removing links according to their criticality rankings and measuring the Global Importance Index (GII), which quantifies unsatisfied demand and importance post-disruption. This sequential disruption analysis allows direct comparison between GNN predictions and traditional vulnerability indexes. Comparative timing analyses demonstrate that our model identifies critical links with comparable accuracy but at a fraction of the computational cost, even under varying disruption scenarios. This work offers a scalable methodology for conducting vulnerability assessments on previously unmanageable network scales, advancing the planning of humanitarian aid distribution during natural disasters.

Thesis supervisor: Andrea Herrera Suescún
Title: Associate Professor of Systems and Computing Engineering

Thesis supervisor: Nicolás Cardozo
Title: Associate Professor of Systems and Computing Engineering

Acknowledgments

I would like to express my gratitude to my advisors for their guidance and support throughout this research process. Their feedback were valuable contributions to the development of this thesis. I am also thankful to Universidad de los Andes for providing the necessary resources and infrastructure that made this work possible. Access to quality research facilities, databases, and technical equipment was instrumental in conducting experiments and gathering data for this study.

Contents

Chapter 1

Introduction

Transportation networks are fundamental to modern society. Facilitating activities such as human encounters, economic exchange, distribution of goods, among others. Consequently, any interruption in the flow of these activities can have devastating effects on the economy and social welfare of communities (Jafino, Kwakkel, and Verbraeck 2020). Therefore, evaluating the vulnerability of transportation networks is essential to detect critical components, mitigate potential risks, and prevent adverse societal impacts during disruptions. Currently, there is no agreement on an exact definition of vulnerability, as it depends on the factors taken into account for its evaluation. Nevertheless, most researchers accept that it is a measure of the network's susceptibility to adverse events, e.g., disasters (Cantillo, Macea, and Jaller 2018). In particular, for the purpose of this research, a transportation network is vulnerable if the loss or degradation of a small number of routes significantly decreases accessibility to affected communities (M. A. P. Taylor and D'Este 2005).

In this context, identifying critical links that deteriorate network performance becomes essential for vulnerability assessment and disaster preparedness. Specifically, in humanitarian aid distribution, vulnerability models structure transportation networks with distinct supply nodes (distribution points) and demand nodes (communities requiring assistance). Various approaches have been proposed to determine links with the greater impact on network connectivity and functionality (Mattsson and Jenelius 2015). However, traditional approaches face significant computational challenges. As network complexity increases—through larger network sizes, additional supply/demand nodes, or more disruption scenarios—the computational burden grows substantially. This escalating computational cost renders comprehensive vulnerability analysis prohibitively expensive for large-scale networks, limiting its practical application in real-world disaster planning (Holguín-Veras et al. 2013).

Despite the acknowledged importance of transportation network vulnerability assessment, efficient and scalable evaluation methodologies remain scarce. Many existing studies focus primarily on technical and logistical aspects without addressing the computational limitations that prevent their application to complex networks (Faturechi and Elise Miller-Hooks 2014). This gap between theoretical vulnerability assessment and practical implementation stems from both the complexity of modeling multiple relevant variables and the intensive computational resources required. Consequently, many current evaluations either fail to identify all critical links effectively or do so at a computational cost that undermines their utility in disaster preparedness and response planning. This highlights the pressing need for more efficient approaches to vulnerability assessment in transportation network.

Recent advances in artificial intelligence, particularly in the field of deep learning, have created new promising alternatives to traditional methods for addressing complex network analysis problems by potentially learning patterns that can identify critical infrastructure components (Zhou, J. Wang, and Yang 2019). The application of such techniques to transportation vulnerability assessment, especially in humanitarian contexts, represents an emerging research direction with significant potential for improving disaster preparedness and response capabilities.

1.1 Problem statement

The vulnerability assessment of transportation networks for humanitarian aid distribution faces a critical challenge: traditional methods for identifying critical links become computationally prohibitive as problem complexity increases. This computational burden escalates along three key dimensions:

First, as the network size grows (more nodes and links), the number of potential disruption scenarios increases exponentially, dramatically extending computation time (Sullivan et al. 2010). Second, an increase in supply nodes (distribution centers) and demand nodes (affected communities) multiplies the number of origin-destination pairs that must be evaluated for each scenario, further compounding computational requirements (Jenelius, Petersen, and Mattsson 2006). Third, considering multiple disruption scenarios (e.g., single-link failures, cascading failures, or region-based disruptions) necessitates repeating the entire analysis for each scenario, creating an additional multiplicative effect on computational costs (Mattsson and Jenelius 2015).

The combined effect of these three factors creates a computational barrier that prevents comprehensive vulnerability analysis of precisely those networks that would benefit most from it—large-scale infrastructure systems with multiple supply/demand points serving extensive populations. In humanitarian response planning, where timely decisions can significantly impact human welfare, this computational barrier presents a serious operational constraint. The humanitarian context further complicates analysis by introducing specialized considerations beyond traffic flow, such as deprivation costs and accessibility to essential services (Holguín-Veras et al. 2013). While advanced computational techniques have shown promise in learning complex patterns from network structures in various domains, their application to humanitarian transportation vulnerability assessment remains largely unexplored. A notable research gap exists in developing a computationally efficient approach that can: (1) accurately identify critical links in large-scale networks, (2) incorporate humanitarian-specific considerations, and (3) provide results comparable to established vulnerability metrics but at a fraction of the computational cost.

This research addresses this gap by developing and validating a novel methodology for critical link identification in transportation networks, with particular emphasis on humanitarian response applications. The proposed approach aims to overcome the computational limitations of traditional methods while maintaining or improving the accuracy of critical link identification, thereby enabling vulnerability assessments at previously infeasible network scales.

1.2 Objectives

1.2.1 General objective

To propose and evaluate advanced vulnerability modeling techniques for transportation networks in disaster scenarios, emphasizing their applicability and effectiveness in humanitarian aid distribution.

1.2.2 Specific objectives

1. Develop novel vulnerability modeling approaches for humanitarian aid distribution based on deep learning (graph neural networks).

2. Design and implement a test suite (graph topologies, graph sizes, and distribution of service and aid points) and robust metrics for algorithm comparison.

3. Compare the proposed GNN-based methodology against established vulnerability indices using the defined test suite and evaluation metrics.

The remainder of this thesis is organized into five chapters: Chapter 2 explores the theoretical foundations of transportation network vulnerability and GNN, establishing the conceptual framework for our research. Chapter 3 presents our technical approach, detailing the traditional vulnerability assessment methods and introducing our GNN-based solution for critical link identification. Chapter 4 outlines the evaluation methodology, describing our performance metrics, sequential disruption approach, and experimental design. Chapter 5 analyzes our experimental findings, presenting performance comparisons across diverse network types and demonstrating computational advantages. Chapter 6 concludes with a summary of contributions, practical implications, and future research directions. Technical specifications for network generation and implementation details are provided in the appendices to ensure reproducibility.

Chapter 2

Related work

This chapter outlines the key theoretical concepts underpinning our research, establishing the context necessary to understand and evaluate our contributions to the field.

2.1 Vulnerability Analysis of Transport Networks

As mentioned by (M. A. Taylor 2017), transportation network vulnerability analysis aims to provide tools that assist transport planners and decision-makers in identifying potential weak points in their networks—critical locations and infrastructures—and in developing plans to protect these points or establish viable alternatives in case of disruptions. Four general approaches are primarily identified:

1. Inventory-based risk assessment: Consists of conducting an inventory of the system's infrastructure assets and rating them according to their level of risk to failures or disruptions, considering factors such as traffic and environmental conditions.

2. Topological assessment: Analyzes the structure and connectivity of the network, identifying the most critical nodes and links for flow through the network, as well as locations whose failure would have the most severe effects on the overall operation of the network.

3. Service operability assessment: Focuses on the impacts that failures and disruptions at different locations have on network operations, thus allowing the identification of critical points from the perspective of transport and traffic.

4. Accessibility-based assessment: Similar to operability assessment, but with a greater emphasis on socioeconomic impacts beyond the direct scope of network traffic.

2.1.1 Human Aid Distribution Emphasis

In humanitarian operations, transportation network vulnerability assessment differs fundamentally from conventional approaches by focusing specifically on supply-demand relationships rather than overall network performance. While traditional analyses often evaluate entire network connectivity, humanitarian contexts prioritize the critical paths between distribution centers (supply nodes) and affected communities (demand nodes) (Holguín-Veras et al. 2013). This specialized perspective narrows the analysis to links that directly impact aid delivery capabilities, recognizing that maintaining accessibility between these specific node pairs is paramount for effective disaster response (Cantillo, Macea, and Jaller 2018). This supply-demand emphasis provides a more targeted approach to identifying critical transportation links in humanitarian operations.

Contemporary research frameworks now incorporate deprivation costs as a critical factor in network vulnerability assessment, recognizing that delays in aid delivery create cascading effects that extend beyond immediate transportation disruptions (Ghahremani-Nahr, Nozari, and Szmelter-Jarosz 2024). This evolution reflects a shift toward data-driven approaches that leverage standardized humanitarian databases and real-time information sharing across global humanitarian hubs . Furthermore, current methodologies emphasize sustainability enablers and operational resilience factors that strengthen humanitarian supply chain networks against multidimensional disruptions, moving beyond static vulnerability assessments to dynamic frameworks that can adapt to evolving crisis conditions (Kashav and Garg 2024). This comprehensive approach recognizes that humanitarian transportation networks must balance efficiency optimization with robust contingency planning to maintain aid distribution capabilities under various disruption scenarios

2.1.2 Detecting Critical Links

Using the definition of vulnerability presented above and the context of humanitarian aid distribution:

A link in a transportation network is critical if the loss of the link significantly deteriorates the network's accessibility to humanitarian aid demand vertices, as measured by a standard accessibility index.

This definition underpins methodologies for identifying critical links in humanitarian transportation networks. The conventional approach involves a network scan, where links are systematically removed one by one, and reassessing network performance after each removal (Sullivan et al. 2010). Typically, an accessibility index (such as global efficiency or Cantillo's index presented in chapter 3) is calculated for the intact network and recalculated after each link removal. The difference in accessibility — representing the degradation in network performance — determines each link's criticality, with those causing the greatest decrease considered most critical.

This approach can be extended to evaluate multiple simultaneous disruptions, reflecting more realistic disaster scenarios where several network components might fail concurrently. However, such analyses dramatically increase in computational requirements (Mattsson and Jenelius 2015). For a network with m links, evaluating all possible single-link disruptions requires m calculations, but analyzing all possible combinations of k simultaneous link failures requires $\binom{m}{k}$ calculations—a number that grows exponentially with network size.

The computational challenge becomes even more substantial when calculating accessibility indices themselves, which typically involve evaluating paths between all supply and demand node pairs. For a network with s supply nodes and d demand nodes, this requires analyzing $s \times d$ paths for each disruption scenario. When these calculations must be repeated for hundreds or thousands of potential disruption scenarios, the computational burden becomes prohibitive for large-scale real-world networks (Jenelius, Petersen, and Mattsson 2006), limiting the feasibility of comprehensive vulnerability assessments in humanitarian planning.

2.1.3 Accessibility Indices

Accessibility indices are critical metrics in vulnerability assessments, quantifying the effectiveness of network connectivity in reaching destinations. In humanitarian contexts, these indices evaluate how effectively aid supplies can reach affected populations.

Three prominent indices appear in the vulnerability assessment literature. Cantillo's index (Cantillo, Macea, and Jaller 2018) uniquely incorporates both logistics costs and deprivation costs, recognizing that delays in aid delivery exponentially increase human suffering over time. Global efficiency (Latora and Marchiori 2001) measures overall network connectivity through the inverse of the harmonic mean of shortest path lengths, providing insight into how directly supply and demand points can connect. The number of independent paths (Ip and D. Wang 2011) quantifies network redundancy by counting alternative routes between node pairs, indicating resilience to partial network failures.

As mentioned in the previous section, these indices typically identify critical links by comparing network performance before and after link removals. Links that cause the most severe accessibility degradation are considered most critical. However, this comparative approach contributes significantly to the computational challenges discussed previously, as each potential disruption scenario requires recalculating network-wide measures (Jafino, Kwakkel, and Verbraeck 2020).

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) extend deep learning capabilities to graph-structured data by capturing both node features and topological relationships. Unlike traditional neural networks, GNNs operate directly on graphs, enabling them to leverage spatial dependencies inherent in networked systems.

GNNs learn node representations through message passing, where each node aggregates information from its neighbors to form an embedding that captures local graph structure. These embeddings can then be used for various prediction tasks, including node classification, link prediction, and graph classification (Wu et al. 2021).

Recent applications of GNNs in transportation research include traffic forecasting (Jiang and Luo 2022), public transit analysis (**Li2022**), and infrastructure resilience assessment (Munikoti, Das, and Natarajan 2022). In particular, GNNs have shown promise for analyzing complex transportation networks by capturing the interdependencies between network components.

The advantage of GNNs lies in their ability to learn vulnerability patterns directly from network structure, eliminating the need to evaluate numerous disruption scenarios after training. While traditional methods require recomputing accessibility indices for each possible disruption, a trained GNN can identify critical links in a single forward pass, regardless of how many disruption scenarios were considered during training. This computational efficiency makes GNNs particularly valuable for transportation vulnerability assessment in humanitarian contexts, where timely decision-making is crucial.

Chapter 3

Vulnerability Assessment Methods for Humanitarian Transport Networks

3.1 Network Representation for Transportation Systems in Disaster Response

Transportation networks in disaster response scenarios are modeled as weighted directed graphs $G = (V, E, A)$, where V represents the set of nodes (locations), E represents the set of edges (transportation links), and $A \in \mathbb{R}^{|V| \times |V|}$, the adjacency matrix, which encodes weights typically representing travel time or distance. In the context of humanitarian aid distribution, two key subsets of nodes are identified:

- **Source nodes ($S \subset V$)**: Represent supply points.
- **Terminal nodes ($T \subset V$)**: Represent affected communities.

This graph-based representation captures the essential supply-demand relationships in humanitarian operations. Vulnerability assessment thus focuses on identifying links that, when disrupted, significantly impair accessibility between these source-terminal nodes.

3.2 Traditional Vulnerability Assessment Framework

Current approaches identify critical links through a systematic process of evaluating network performance degradation under various disruption scenarios.

Disruption scenarios

Given a transport network $G = (V, E, A)$, a disruption scenario $D \subseteq E$ represents a subset of edges that are simultaneously disrupted [1].

In vulnerability network analysis, we typically set a threshold level K representing the maximum number of edges that can be simultaneously disrupted in any scenario, reflecting practical constraints on the extent of network damage that might reasonably occur.

Subsequently, the set of all possible disruption scenarios is

$$\mathcal{D} = \{D \mid D \subseteq E, 1 \leq |D| \leq K\} \tag{3.1}$$

Where K is the maximum number of simultaneously disrupted edges considered in the analysis. The size of $\mathcal{D}$ grows combinatorially with the number of edges $|E|$ in the network and the maximum disruption size K:

$$|\mathcal{D}| = \sum_{k=1}^{K} \binom{|E|}{k} \tag{3.2}$$

For single-link disruption analysis ($K = 1$), $|\mathcal{D}| = |E|$, making exhaustive evaluation feasible for moderately sized networks. However, for multi-link disruptions ($K > 1$), the problem becomes computationally intractable as the network size increases (Jenelius, Petersen, and Mattsson 2006). We explore the size of K for this research in section 3.2.3.

[1] In network analysis, a set of edges being disrupted usually means their weights are altered. For the purpose of this research, and following the standard approach, a set of edges being disrupted means its removal from the network.

Accessibility function

An accessibility function $\mathcal{A} : \mathcal{G} \times \mathcal{P}(V) \times \mathcal{P}(V) \to \mathbb{R}$ maps a graph $G \in \mathcal{G}$ and two sets of nodes $S, T \subset V$ to a real number $\mathcal{A}(G, S, T)$ that quantifies the level of accessibility between S and T in graph G.

Criticality score

Given a graph G, a set of source nodes S, and terminal nodes T, we can measure the impact of a disruption scenario D using an accessibility function $\mathcal{A}$. The impact is quantified as:

$$\Delta = \mathcal{A}(G, S, T) - \mathcal{A}(G \setminus D, S, T)$$

Where $G \setminus D$ represents the network after removing the edges in disruption scenario D. A higher value of Δ indicates a more severe impact on network performance, and thus a more critical disruption scenario.

For single-link disruptions (where D contains only one edge), the change in accessibility Δ directly measures that edge's criticality. For multi-link disruptions (where D contains multiple edges), we distribute the total impact Δ among all edges in D to determine their individual criticality scores. The complete procedure for computing these criticality scores is detailed in Algorithm 3.2.1.

3.2.1 Criticality Score Computation Algorithm

This algorithm demonstrates the inherent computational challenge of traditional vulnerability assessment. For a network with $|E|$ edges, evaluating single-link disruptions requires $|E|$ separate calculations of the accessibility index. For multi-link disruption scenarios, the computational burden grows exponentially, as exposed in the previous section.

The calculation of the accessibility index itself adds another layer of computational complexity, as it typically involves evaluating paths between all source and terminal node pairs. For networks with $|S|$ source nodes and $|T|$ terminal nodes, this requires analyzing $|S| \times |T|$ paths for each disruption scenario.

Algorithm 1: Criticality Score Computation

```
Data: Directed graph $G$, set of disruptions $\mathcal{D}$, accessibility index function $\mathcal{A}$
Result: Dictionary of edge criticality scores
Extract source and terminal nodes ($S, T$ respectively) from $G$;
Compute initial performance $P_0 = \mathcal{A}(G, S, T)$;
Initialize empty dictionary link_performance;
foreach disruption $D$ in $\mathcal{D}$ do
    Compute new performance $P_D = \mathcal{A}(G \setminus D, S, T)$;
    Compute $\Delta = P_0 - P_D$;
    foreach edge in disruption do
        Increment link_performance[edge] by $\Delta$ divided by the size of disruption;
    Restore edges to $G$;
return link_performance;
```

3.2.2 Accessibility Indices

Global Efficiency

Global Efficiency (GE) measures the efficiency of information or resource flow across a network through the inverse of the harmonic mean of shortest path lengths:

$$GE = \frac{1}{|V|(|V|-1)} \sum_{i \neq j} \frac{1}{d_{ij}}$$

Where:

- $|V|$ is the number of nodes in the network
- d_{ij} is the shortest distance between nodes i and j

For humanitarian applications, we adapt this measure to focus specifically on source-terminal pairs:

$$\mathcal{A}_{GE}(G, S, T) = \frac{1}{|S||T|} \sum_{s \in S} \sum_{t \in T} \frac{1}{d_{st}} \tag{3.3}$$

This adapted measure quantifies how directly source nodes (distribution centers) can connect to terminal nodes (affected communities).

Number of Independent Paths

The Number of Independent Paths (IP) is a network measure that quantifies redundancy by counting the number of paths between nodes that do not share any common edges. In the original formulation by (Ip and D. Wang 2011), this measure is defined for the entire network as:

$$IP = \sum_{i=1}^{|V|} \frac{1}{|V|-1} \sum_{j=1}^{|V|} k(i,j)$$

Where:

- $|V|$ is the number of nodes in the network
- $k(i,j)$ is the number of independent paths between nodes i and j

For humanitarian applications, we adapt this measure to focus specifically on source-terminal pairs:

$$\mathcal{A}_{IP}(G,S,T) = \sum_{s\in S} \frac{1}{|T|} \sum_{t\in T} k(s,t) \tag{3.4}$$

This metric is particularly valuable for humanitarian operations as it captures the availability of alternative routes when primary paths are disrupted, indicating the network's resilience to partial failures (Haritha and Anjaneyulu 2024). A higher number of independent paths between source and terminal nodes suggests greater redundancy in the network, which is critical for ensuring continuous aid delivery during disasters.

Cantillo's Index

Cantillo's Index uniquely incorporates both logistics costs and deprivation costs, capturing human suffering and making it particularly relevant for humanitarian response operations. The accessibility function, inspired by Cantillo, Macea, and Jaller 2018, is computed as:

$$\mathcal{A}_{CI}(G,S,T) = \ln\left(\sum_{p\in P} e^{-\mu V_p}\right) \tag{3.5}$$

Where:

- P is the set of al reasonable paths.[2]
- $V_p = \Omega_p + \gamma_p$ is the utility of the reasonable path p with $\Omega_p = \sum_{a \in p} C_a$ as the social cost and γ_p is the deprivation cost calculated with a deprivation cost function[3] based on travel time.

3.2.3 Generation of Disruption Scenarios

Given the computational challenges of evaluating all possible disruption scenarios $\mathcal{D}$, we employ a strategic sampling approach. Following empirical studies of transportation network resilience (Haritha and Anjaneyulu 2024), we focus on disruption scenarios that affect at most 40% of the network edges, as historical data suggests that simultaneous disruption of more than 40% of transportation links is extremely rare even in severe disaster scenarios.

We set our tolerance level $K = \lfloor 0.4 \times |E| \rfloor$ and, therefore, from 3.1,

$$\mathcal{D}_{K=40\%} = \{D \mid D \subseteq E, 1 \leq |D| \leq \lfloor 0.4 \times |E| \rfloor\}$$

Now, we define our set of disruption scenarios $\mathcal{D}'$ as:

$$\mathcal{D}' = \begin{cases} \mathcal{D}_{K=40\%}, & \text{if } |\mathcal{D}_{K=40\%}| \leq 10,000 \\ \mathcal{D}_{\text{sample}}, & \text{otherwise} \end{cases} \tag{3.6}$$

Where $\mathcal{D}_{\text{sample}}$ is a sample of 10,000 disruption scenarios. This sample is constructed by first exhaustively including all disruptions of size $k = 1, 2, \ldots, k_{\max}$ until adding one more is not viable, i.e., $\mathcal{D}_{k_{\max}} = \{D \mid D \subseteq E, 1 \leq |D| \leq k_{max}\}$ such that $10,000 - \binom{|E|}{k_{\max}+1} < |\mathcal{D}_{k_{\max}}| < 10,000$. Then randomly sampling from the remaining higher-order disruptions until reaching a total of 10,000 scenarios, with maximum disruption size capped at $k = 40\%|E|$.

This approach ensures comprehensive coverage of lower-order disruptions, which typically have higher probability of occurrence in real-world scenarios (Mattsson and Jenelius 2015), while still maintaining computational feasibility. Even with this sampling approach, generating and evaluating up to 10,000 disruption scenarios remains computationally intensive for large networks,

[2] The set of reasonable paths between sources S and terminals T are obtained with Dial's algorithm. Cantillo, Macea, and Jaller 2018

[3] Based on Cantillo, Macea, and Jaller 2018, the deprivation cost function is $\gamma(t) = 0.0216255(t^2) + 0.052425t + 0.8272$ where t is time

as each scenario requires recalculating the accessibility index between all source and terminal node pairs.

3.3 Graph Neural Network Approach

Given a graph representation $G = (V, E, A)$ as defined in subsection 3.1, GNNs require an additional important element: the node feature matrix $X \in \mathbb{R}^{|V|\times d}$, where $|V|$ is the number of nodes and d is the dimension of the feature vectors. Each row X_i corresponds to a d-dimensional feature vector of node v_i, capturing its attributes in the transportation network.

3.3.1 Model Architecture

We implement a modular architecture consisting of three components specifically designed for detecting link criticality: (1) a node embedding module that processes and transforms graph features, (2) a link embedding layer that converts node-level information to edge representations, and (3) an MLP module that produces the final criticality scores. This structure allows us to effectively capture both local connectivity patterns and wider network characteristics that influence link criticality. The architecture is illustrated in Figure 3.1

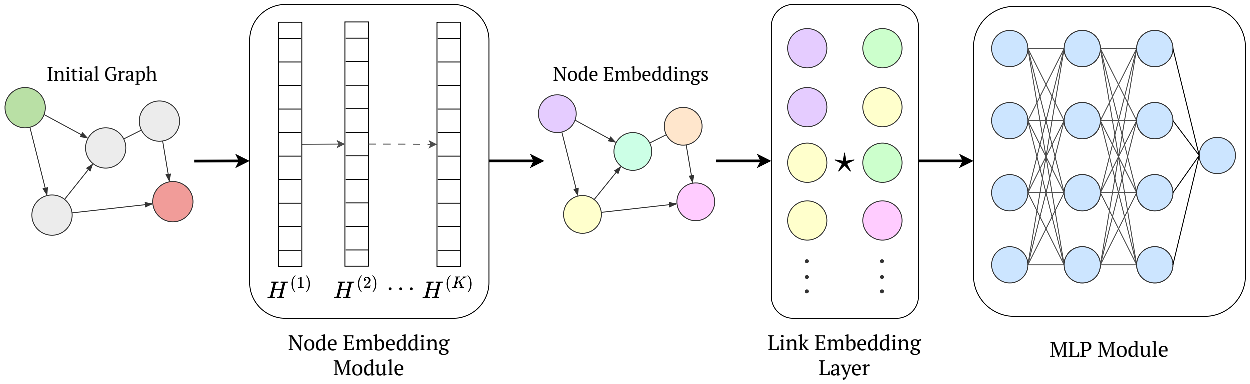


Figure 3.1: GNN Architecture

Node Embedding Module

The node embedding module transforms the initial node feature matrix into learned representations by aggregating information from neighboring nodes. For a given layer $\ell > 0$, in a high-level concept, the operations for a node v are:

$$H_v^{(\ell)} = \gamma_\Theta^\ell\left(H_v^{(\ell-1)}, \bigoplus_{u\in\mathcal{N}(v)} \phi_\Theta^\ell\left(H_v^{(\ell-1)}, H_u^{(\ell-1)}, e_{u,v}\right)\right)$$

Where:

- $H^{(0)}$ is the initial node features X
- $\mathcal{N}(v)$ is the set of neighbors of node v. In a directed graph, $\mathcal{N}(v) = \{u : (u, v) \in E\}$
- $\bigoplus$ denotes a differentiable, permutation invariant function, e.g., sum, mean, min, max.
- γ_Θ^ℓ takes the embedding of a node v in $\ell - 1$ and operates with the aggregated embedding.
- ϕ_Θ is the operator between the embedding node and its neighbor.

Convolutional Layer

For the GCN implementation, the node-wise update rule for a node v at layer ℓ is defined as:

$$H_v^{(\ell)} = \sigma\left(W^{(\ell)} \sum_{u\in\mathcal{N}(v)\cup\{v\}} \frac{1}{\sqrt{\tilde{d}_v\tilde{d}_u}} H_u^{(\ell-1)}\right) \tag{3.7}$$

Where:

- $H_v^{(\ell)}$ is the feature vector of node v at layer ℓ
- $\tilde{d}_v = d_v + 1$ represents the degree of node v plus one (accounting for self-loops)
- $W^{(\ell)} \in \mathbb{R}^{d_\ell\times d_{\ell-1}}$ is the trainable weight matrix for layer ℓ
- d_ℓ is the dimension of the node embeddings in layer ℓ.
- σ is a non-linear activation function (ReLU in our implementation)

Attention Layer

For the GAT implementation, each node v aggregates information from its neighborhood using learned attention weights:

$$H_v^{(\ell)} = \sigma\left(\sum_{u\in\mathcal{N}(v)\cup\{v\}} \alpha_{vu}^{(\ell)} W^{(\ell)} H_u^{(\ell-1)}\right) \tag{3.8}$$

The attention coefficient $\alpha_{vu}^{(\ell)}$ between nodes v and u is computed as:

$$\alpha_{vu}^{(\ell)} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{a}^{(\ell)T}\left[W^{(\ell)}H_v^{(\ell-1)} \| W^{(\ell)}H_u^{(\ell-1)}\right]\right)\right)}{\sum_{w\in\mathcal{N}(v)\cup\{v\}}\exp\left(\text{LeakyReLU}\left(\vec{a}^{(\ell)T}\left[W^{(\ell)}H_v^{(\ell-1)} \| W^{(\ell)}H_w^{(\ell-1)}\right]\right)\right)} \tag{3.9}$$

The key components include:

- $H_v^{(\ell)}$ is the feature vector of node v at layer ℓ
- $W^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ transforms the input features to dimension d_ℓ
- $\vec{a}^{(\ell)} \in \mathbb{R}^{2d_\ell}$ is the attention vector that learns the importance of node pairs
- $\|$ represents concatenation of vectors
- We implement multi-head attention with K independent attention mechanisms:

$$H_v^{(\ell)} = \|_{k=1}^{K} \sigma\left(\sum_{u\in\mathcal{N}(v)\cup\{v\}} \alpha_{vu}^{(\ell,k)} W^{(\ell,k)} H_u^{(\ell-1)}\right) \tag{3.10}$$

This implementation allows the model to learn which neighboring nodes are most important for determining a node's representation, enabling a more precise capture of the structural patterns that influence link criticality.

In summary, for this module, we'll be analyzing:

1. Layers type: [GCN, GAT]
2. Embedding size: [64, 128, 256]

3. Number of layers: [1, 2, 3, 4, 5, 6]

4. Other specific configuration for layers, such as number of heads for GAT, second version of GAT, etc.

Link Embedding Layer

The link embedding layer transforms node embeddings into link representations. For each edge $(i, j) \in E$, we compute a link embedding as:

$$H_{ij} = H_i \star H_j \tag{3.11}$$

Where $\star$ represents the Hadamard (element-wise) product of the node embeddings H_i and H_j. This operation captures the pairwise interactions between connected nodes.

Additionally, we incorporate direct edge features (such as travel time, capacity, and edge betweenness) through a skip connection from the input layer, concatenating these features with the computed link embedding:

$$H_{ij}^{\text{enhanced}} = [H_{ij} \| e_{ij}] \tag{3.12}$$

Where e_{ij} represents the original edge features and $[\cdot\|\cdot]$ denotes concatenation. This enhanced link embedding combines both the learned representations of the endpoint nodes and the explicit edge attributes, providing a rich representation for criticality prediction.

In summary, for this layer we analyze:

1. Operator $\star$ (see figure 3.1): [concat (||), mean, Hadamard product ($\odot$)]

Criticality Prediction MLP Module

The final component is a multi-layer perceptron (MLP) that maps link embeddings to criticality scores. The MLP consists of several fully connected layers with ReLU activations, followed by a final layer that outputs a single scalar value for each link:

$$y_{ij} = \text{MLP}(H_{ij}^{\text{enhanced}}) \tag{3.13}$$

The architecture of the MLP block is as follows:

- Input layer: Dimension of the enhanced link embedding
- Hidden layers: Two layers with 64 and 32 neurons respectively, each followed by ReLU activation and dropout (rate=0.2) for regularization
- Output layer: Single neuron outputting the criticality score. Depending on the loss function, it applies a sigmoid or none function.

This final block transforms the high-dimensional link embeddings into scalar criticality scores that can be used to rank the links based on their importance to the transportation network. The higher the score, the more critical the link is predicted to be.

3.3.2 Feature Engineering for Critical Link Prediction

For our node feature representation, we utilize a minimal yet effective approach based on the functional role of each node in the transportation network vulnerability analysis. Specifically, we categorize nodes into three distinct types: (1) source/supply node, (2) terminal/demand node and (3) regular node. We encode this categorical information using one-hot encoding, resulting in a 3-dimensional feature vector for each node:

$$X_v = \begin{cases} [1, 0, 0] & \text{if } v \text{ is a source (supply) node} \\ [0, 1, 0] & \text{if } v \text{ is a terminal (demand) node} \\ [0, 0, 1] & \text{if } v \text{ is a regular node} \end{cases} \tag{3.14}$$

This representation, while simple, provides essential topological context about each node's function within the vulnerability assessment framework. In transport vulnerability analysis, distinguishing between supply and demand nodes is crucial as disruptions to links connecting these nodes often have disproportionate impacts on network performance. Our approach is motivated by research showing that even minimal node features can be effective when combined with the powerful structural learning capabilities of GNNs, which capture higher-order connectivity patterns during message passing operations.

For the target variable, we normalize the criticality scores of edges using min-max normalization to ensure all values fall within the range [0,1]:

$$\text{criticality}_{\text{normalized}}(e) = \frac{\text{criticality}(e) - \min_{\hat{e} \in E}(\text{criticality}(\hat{e}))}{\max_{\hat{e} \in E}(\text{criticality}(\hat{e})) - \min_{\hat{e} \in E}(\text{criticality}(\hat{e}))} \tag{3.15}$$

This normalization facilitates more stable training and allows for consistent interpretation of predicted criticality scores across different network instances.

3.3.3 Loss Function Design for Criticality Ranking

Since our ultimate objective is to correctly rank transport links according to their criticality rather than precisely predict absolute criticality values, we investigate both pointwise and pairwise learning approaches for our loss function design.

Pointwise Approach

For the pointwise approach, we employ Mean Squared Error (MSE) as our loss function:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{|E|} \sum_{(i,j) \in E} (y_{ij} - \hat{y}_{ij})^2 \tag{3.16}$$

where y_{ij} is the true normalized criticality score for link (i, j) and $\hat{y}_{ij}$ is the predicted score. This formulation treats link criticality prediction as a standard regression problem, directly minimizing the squared difference between predicted and actual criticality values. While MSE effectively captures the magnitude of prediction errors, it lacks awareness of the relative ordering between links. This limitation is significant in vulnerability assessment contexts, where correctly identifying which links are more critical than others (the ranking) is often more important than predicting the exact criticality values themselves.

Pairwise Approach

For the pairwise approach, we explicitly optimize for correct relative ordering of link pairs. Given a pair of links (i, j) and (k, l) with true criticality scores y_{ij} and y_{kl} where $y_{ij} > y_{kl}$, we want to ensure that $\hat{y}_{ij} > \hat{y}_{kl}$.

We investigate multiple pairwise loss functions:

1. **Logistic loss**:
$$\mathcal{L}_{\text{log}} = \sum_{(i,j),(k,l)\in P} \log(1 + \exp(-(\hat{y}_{ij} - \hat{y}_{kl}))) \tag{3.17}$$

2. **Exponential loss**:
$$\mathcal{L}_{\text{exp}} = \sum_{(i,j),(k,l)\in P} \exp(-(\hat{y}_{ij} - \hat{y}_{kl})) \tag{3.18}$$

3. **Square root exponential loss**:
$$\mathcal{L}_{\text{sqrt-exp}} = \sum_{(i,j),(k,l)\in P} \sqrt{\exp(-(\hat{y}_{ij} - \hat{y}_{kl}))} \tag{3.19}$$

where P is the set of all link pairs $(i, j), (k, l)$ such that $y_{ij} > y_{kl}$.

The pairwise losses are specifically designed to penalize incorrect ordering of link pairs, which directly optimizes for the ranking task in vulnerability assessment. The logistic loss provides a smooth approximation to the 0-1 loss for incorrect rankings, the exponential loss places higher penalties on pairs with large ranking errors, while the square root exponential loss moderates the penalty for extreme cases to improve training stability.

In our experiments, we compare these different loss functions to determine which best supports the accurate identification of critical links in vulnerable transportation networks.

3.3.4 Multi-stage Hyperparameter Optimization

Efficiently optimizing neural network hyperparameters requires a strategic approach that balances exploration breadth with computational efficiency. We implemented a three-stage hyperparameter optimization framework using Optuna, Akiba et al. 2019, a state-of-the-art hyperparameter optimization library that provides sophisticated algorithms for efficiently navigating complex parameter spaces. Optuna's Bayesian optimization capabilities allow us to progressively narrow the search space while increasing training duration for each trial. This methodical approach allows us to identify optimal configurations without exhausting computational resources. Our hyperparameter optimization explores the following parameters:

- Node Embedding size: [32, 64, 128, 256]
- Message passing layers: [1-5]
- MLP layers: [1-5]
- Edge embedding operator: ["concat", "hadamard", "mean"]
- Dropout: [0.1-0.5]
- Batch size: [16, 32, 64]
- Message passing mechanism: ["GCN", "GAT"]
- GAT version: [v1, v2] (conditional)
- GAT attention heads: [1, 2, 4, 8] (conditional)
- Loss function: ["MSE", "RANK"]
- Loss transformation: ["sqrt_exp", "exponential"]
- Learning rate: [10^{-5}-10^{-2}]

Exploration Stage

In the initial exploration stage, we cast a wide net across the entire hyperparameter space to identify promising regions. We define wide boundaries for all hyperparameters. Each configuration is trained for only 10-15 epochs, sufficient to detect performance trends without excessive computation. We evaluate 200 trials to adequately sample the search space, with aggressive pruning of underperforming trials to conserve resources. This stage provides a computationally efficient map of the hyperparameter landscape, revealing which regions merit deeper investigation.

Refinement Stage

The refinement stage focuses computational resources on the most promising areas identified during exploration. We restrict hyperparameters to the vicinity of top-performing configurations from the previous stage, with each trial running for 30 epochs to allow more reliable performance assessment. We conduct 20-30 trials within this restricted parameter space and analyze which hyperparameters most significantly impact model performance to prioritize their optimization. This stage enables fine-grained differentiation between competing configurations and identifies subtle interaction effects between parameters that were not apparent during the initial exploration.

Final Stage

The final stage thoroughly evaluates the top candidates to select the optimal configuration. We select 3-5 top-performing configurations from the refinement stage and subject each to complete training with 100 epochs.

Chapter 4

Evaluation Framework

4.1 Evaluation Metrics

To evaluate the effectiveness of our GNN approach for identifying critical links in transportation networks, we take into account four groups of metrics: Global Importance Index (GII), which combines travel cost increases and unsatisfied demand; Ranking Correlation Metrics using Kendall's tau-b coefficient to assess alignment with traditional vulnerability indices; Ranking Quality Metrics including MAP@K and NDCG@K to evaluate the identification of truly critical links; and Computational Efficiency Metrics measuring scalability across network size, disruption scenarios, and source-terminal configurations.

4.1.1 Global Importance Index (GII)

The Global Importance Index (GII) is employed as the primary performance metric for evaluating critical link identification. As defined by Haritha and Anjaneyulu (2024), GII combines both conventional travel cost increases and the humanitarian-specific concept of unsatisfied demand. In the context of humanitarian aid distribution, this metric captures the dual impact of network disruptions: delayed delivery of critical supplies and complete isolation of communities, potentially leading to severe deprivation.

$$GII = \frac{\sum_{s \in S} \sum_{t \in T} x_{st}(c_{st}^d - c_{st}^u) + \alpha \times \sum_{s \in S} \sum_{t \in T} u_{st}^d}{\sum_{s \in S} \sum_{t \in T} x_{st}} \tag{4.1}$$

The numerator's first term measures increased travel costs between source nodes (S) and terminal nodes (T), weighted by the demand between them (x_{st}). Here, c_{st}^u represents travel time under normal conditions, while c_{st}^d represents travel time after disruption. The second term addresses scenarios where connections become completely severed. In such scenarios, u_{st}^d represents the unsatisfied demand between source s and terminal t, defined as:

$$u_{st}^d = \begin{cases} x_{st}, & \text{if } c_{st}^d = \infty \\ 0, & \text{otherwise} \end{cases}$$

4.1.2 Ranking Correlation Metrics

To assess the alignment between our GNN predictions and traditional vulnerability indices, we employ Kendall's tau-b coefficient (τ_b). This non-parametric measure evaluates the monotonic relationship between two rankings while accounting for ties, a common occurrence in criticality rankings when multiple links have identical impacts on network performance. Kendall's tau-b is calculated as:

$$\tau_b = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \tag{4.2}$$

Let $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ be the set of n links such that a tuple (x_i, y_i) represents the criticality scores of two different indices x and y for link i. Then,

- n_c is the number of concordant pairs representing link pairs that both rankings order in the same direction, i.e., pairs with $0 < i < j < n$ where $x_i < x_j$ and $y_i < y_j$ or $x_i > x_j$ and $y_i > y_j$
- n_d is the number of discordant pairs where rankings disagree on ordering, i.e., pairs with $0 < i < j < n$ where $x_i < x_j$ and $y_i > y_j$ or $x_i > x_j$ and $y_i < y_j$
- $n_0 = \frac{n(n-1)}{2}$ represents the total number of possible pairs where n is the total number of links

- $n_1 = \sum_i \frac{t_i(t_i-1)}{2}$ accounts for ties in the first ranking, with t_i being the number of tied values in the ith group of ties and $n_2 = \sum_j \frac{u_j(u_j-1)}{2}$ similarly adjusts for ties in the second ranking, with u_j representing the number of tied values in the jth group of ties.

The inclusion of these tie-adjustment factors is essential for transportation network analysis, where multiple links often have identical criticality scores.

4.1.3 Ranking Quality Metrics

In humanitarian logistics, identifying the most critical links is often more important than accurately ranking all links. As noted by Jafino et al. (2020), decision-makers typically focus on protecting or monitoring a limited subset of the most vulnerable infrastructure components. Therefore, we employ quality metrics focused on the top-k critical links:

MAP@K

To evaluate our model's performance in pinpointing the most impactful disruptions, we adapted a metric from information retrieval, the Mean Average Precision at K (MAP@K):

$$\text{MAP@}K = \frac{1}{|N|} \sum_{G \in N} \text{AP@}K_G \tag{4.3}$$

where N is the set of all test networks, K represents a percentage of the total links (e.g., $K = 5\%$), and AP@K_G is the Average Precision for network G. Since networks vary in size, K_G represents the actual number of links in the top $K\%$ for network G (e.g., $K_G = 5$ links in a 100-link network, $K_G = 10$ links in a 200-link network). The Average Precision is calculated as:

$$\text{AP@}K_G = \frac{1}{|\mathcal{T}_K|} \sum_{k=1}^{K_G} \text{Precision@}k \times \mathbb{1}_{\{\text{e}^{(k)} \in \mathcal{T}_K\}}$$

with $\mathbb{1}_{\{\text{e}^{(k)} \in \mathcal{T}_K\}}$ being an indicator function that equals 1 if the link e ranked at k_{th} position (in the predicted ranking) is among the relevant links in top-K, $\mathcal{T}_K$, and 0 otherwise.

$$\text{Precision@}k = \frac{|\mathcal{T}_k|}{k}$$

This metric quantifies the proportion of truly critical links captured by our model's predictions. It measures the accuracy of our model in identifying the links that would have the greatest impact on humanitarian operations if disrupted.

NDCG@K

In addition to precision, we employ an adapted Normalized Discounted Cumulative Gain (NDCG@K) to evaluate the quality of our ranked list of critical links for multiple networks:

$$\text{NDCG@}K = \frac{1}{|N|} \sum_{G \in N} \frac{\text{DCG@}K_G}{\text{IDCG@}K_G} \tag{4.4}$$

where N is the set of all test networks, K represents a percentage of total links (e.g., $K = 5\%$), and K_G represents the corresponding number of links in network G as defined previously. DCG@K_G is the Discounted Cumulative Gain at K calculated as:

$$\text{DCG@}K_G = \sum_{k=1}^{K_G} \frac{s(k)}{\log_2(k+1)} \tag{4.5}$$

with $s(k)$ being the real score of the link at position k and IDCG@K_G is the Ideal Discounted Cumulative Gain, which is the DCG@K_G of the perfectly ordered list.

4.1.4 Computation Efficiency Metrics

Assessing computational efficiency is paramount when identifying critical links in large-scale transportation networks, especially in time-sensitive humanitarian contexts. We measure execution time across three key dimensions:

- **Network Size Scalability**: We measure how execution time T scales with increasing network size, defined by the number of nodes $|V|$ and edges $|E|$. Traditional methods exhibit polynomial or even exponential growth in computational time as network size increases (Munikoti, Das, and Natarajan 2022), while our GNN approach aims to achieve near-linear scaling post-training.

- **Disruption Scenario Scalability**: We evaluate how execution time varies with the number of disruption scenarios $|D|$. This is particularly relevant for vulnerability analysis, as conventional methods must re-evaluate network performance for each scenario, whereas our GNN approach aims to generalize across scenarios without requiring re-computation.

- **Source-Terminal Configuration Scalability**: We measure execution time sensitivity to the number of source $|S|$ and terminal $|T|$ nodes. A critical factor in humanitarian contexts where the configuration of aid distribution points (sources) and affected communities (terminals) may vary considerably.

For fair comparison, all timing experiments are conducted on identical hardware evaluate both the online phase (inference time) and the offline phase (training time) for our GNN approach. For traditional methods, we measure the complete execution time required to calculate the criticality rankings.

4.2 Sequential Disruption Methodology

4.2.1 Criticality Based Link Removal Process

Following the methodology established by Haritha and Anjaneyulu (2024), we employ a sequential link removal process to evaluate the impact of critical link identification on network performance. The process involves the following steps:

1. Rank all links according to their criticality scores, derived either from traditional indices or GNN predictions
2. Sequentially remove links in decreasing order of criticality
3. Recalculate network performance metric (GII) after each removal
4. Construct a degradation curve showing how performance decreases as critical links are systematically removed

This approach provides a direct, quantifiable measure of each method's effectiveness in identifying links that, when removed, cause maximal network performance degradation.

4.2.2 Aggregate Performance Curve Analysis

Aggregate performance curves offer a visual framework for comparing the effectiveness of different criticality ranking methods. By analyzing the steepness of the degradation curve until a specified K threshold. For example, in figure 4.1 Method A demonstrates steeper initial degradation, indicating more effective identification of the most critical links compared to Methods B and C. This pattern of performance deterioration provides a quantitative basis for comparing the effectiveness of different criticality ranking approaches.

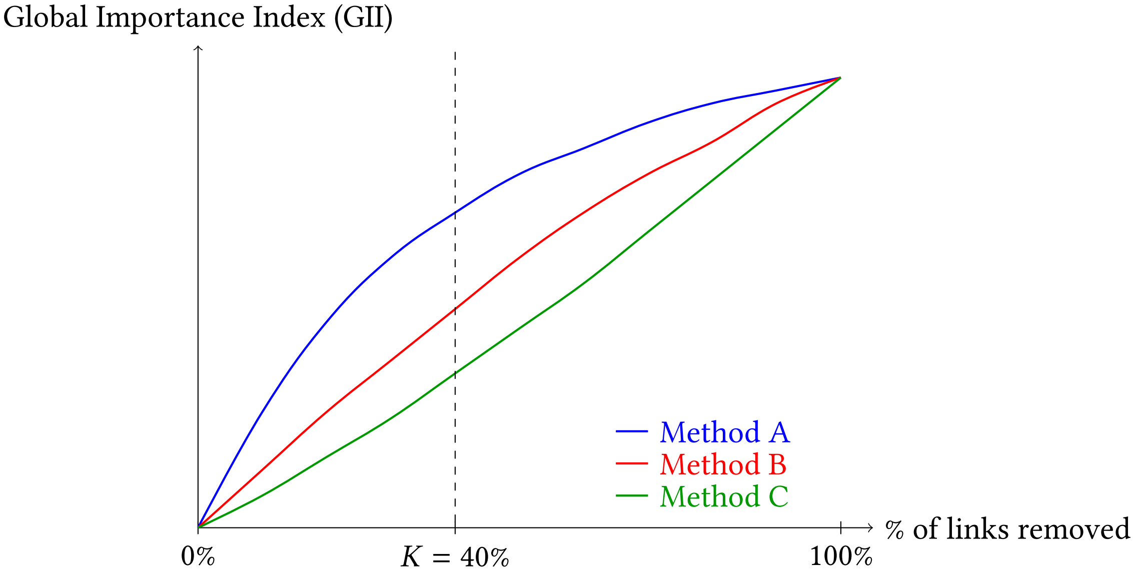


Figure 4.1: Performance degradation curves comparing different critical link identification methods. The x-axis represents the percentage of links removed in order of criticality ranking, while the y-axis shows the Global Importance Index (GII). The vertical dashed line at $K = 40\%$ indicates the maximum disruption threshold considered for analysis, as historical data suggests that simultaneous disruption of more than 40% of transportation links is extremely rare even in severe disaster scenarios (Haritha and Anjaneyulu 2024).

Area Under the Curve

The Area Under the Curve (AUC) provides a quantitative measure for comparing different critical link identification methods, especially when visual inspection alone is insufficient. This metric is particularly important in the following scenarios:

- When curves intersect, AUC offers a comprehensive performance measure across the entire analysis range. For example, in Figure 4.2, Method B exhibits a larger AUC despite Method A performing better initially, indicating better overall performance when considering all potential disruption levels.

- For subtly different curves with similar visual profiles, AUC provides mathematical precision capturing nuanced performance differences that may be overlooked through visual inspection alone.
- When comparing multiple methods (more than two), visual comparison becomes increasingly difficult and subjective. AUC provides a single numerical value that enables consistent and objective ranking of multiple approaches.

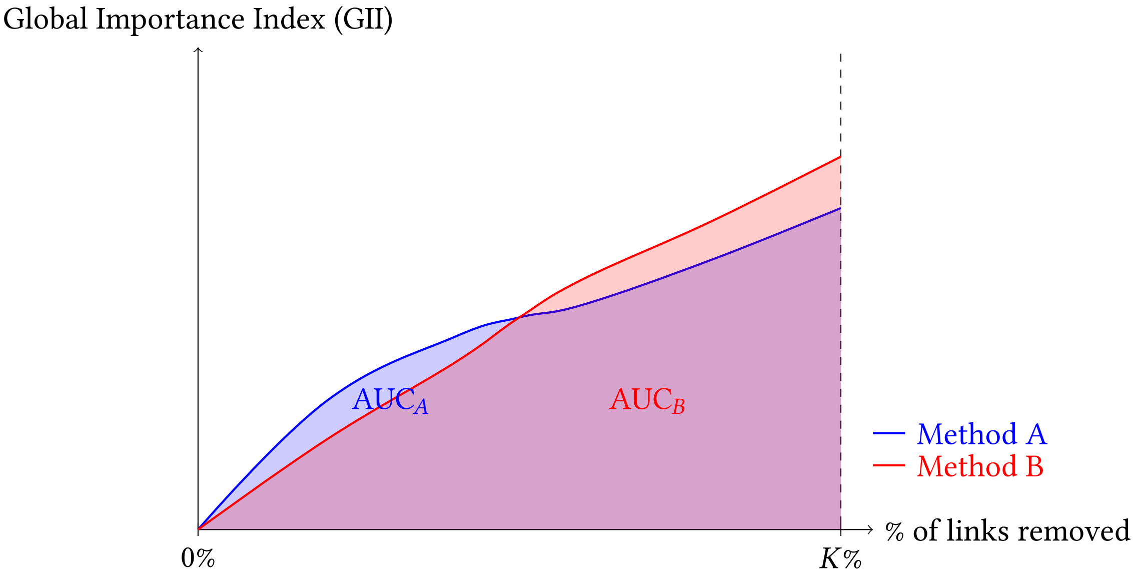


Figure 4.2: Performance degradation curves comparing two link criticality identification methods up to the maximum analysis threshold of K%. The curves intersect at approximately the midpoint of link removal, creating a distinctive crossover effect in performance.

The area under the curve can be calculated as

$$\mathrm{AUC} = \int_0^K \mathrm{GII}(x)\, dx$$

Where x represents the percentage of removed links and GII(x) represents the Global Importance Index at that point. Due to the discrete nature of the data points, as (Haritha and Anjaneyulu 2024) suggests, we can approximate this integral using numerical methods such as the Trapezoidal rule:

$$\text{AUC} \approx \sum_{i=1}^{n} \frac{\text{GII}(x_{i-1}) + \text{GII}(x_i)}{2} \cdot (x_i - x_{i-1}) \tag{4.6}$$

4.3 Experimental Setup

4.3.1 Dataset Overview

Synthetic transport networks

Our critical link analysis dataset comprises a curated list of transportation network topologies informed by research on network vulnerability (Zhang, E. Miller-Hooks, and Denny 2015). The dataset features 11 distinct topology types across 3 size categories (small, medium, large). For each topology-size combination, we generate multiple variants by applying 2-3 parameter configurations with 2-3 random seeds each. These base structures are then enriched with 3-5 different edge weight patterns to simulate various transportation conditions and 3-5 different source-terminal configurations inspired by humanitarian aid distribution challenges, where identifying critical transportation links between supply points and affected populations is essential during disaster response. As emphasized in recent research (Jafino, Kwakkel, and Verbraeck 2020), link criticality rankings change dramatically depending on origin-destination patterns—a link may be critical for delivering aid to one community but redundant for another. This comprehensive approach creates 4350 unique networks that challenge GNN models to identify genuinely critical infrastructure across diverse scenarios, ultimately supporting more effective disaster response planning and humanitarian logistics operations. For specific details of data generation, see appendix A

Real transport networks

To complement our synthetic topologies, we incorporate several real-world transportation networks widely used as benchmarks in vulnerability analysis research:

1. Sioux Falls Network: A simplified representation of Sioux Falls, South Dakota, consisting of 24 nodes and 76 directional links.

2. Eastern Massachusetts Network: A representation of the Boston region road system with 74 nodes and 258 links.

3. Anaheim Network: A larger network with 416 nodes and 914 links, representing a significant urban transportation system.

4. Tokyo Network: A representation of Tokyo's transportation infrastructure comprising 344 nodes and 471 links.

4.3.2 Training and Testing Configuration

Based on current best practices, we implement the following training and test configuration for our dataset of 4,350 network instances:

- We employ a 70/20/10 split ratio, allocating 70% (3,045 networks) for training, 20% (870 networks) for validation, and 10% (435 networks) for testing.

- To ensure representative distributions across all splits, we implement stratified sampling based on network topology type (11 variants) and network size (small, medium, large)

- The ground of truth are the criticality scores computed by the accessibility indices. They are normalized (min-max) for each network since the magnitude of scores depends on network sizes.

Time of generating labels

The generation of ground truth criticality scores using traditional vulnerability indices represents a significant computational investment in our methodology. For each network in our dataset:

- Global Efficiency labels: Average generation time of 10 minutes per network.

- Independent Paths labels: Average generation time of 15 minutes per network.

Total label generation time for the complete dataset of 4350 networks required approximately 700+ hours of computation using parallel processing across 32 CPU cores (CPU Ryzen 7 5800X). This one-time computational cost enables training models that can subsequently perform inference in miliseconds.

Chapter 5

Results

5.1 Environment Execution

All experiments were conducted in a controlled environment to ensure consistent and reproducible results. For training the GNN models, we utilized an NVIDIA RTX 3090 GPU with 24GB of VRAM, which provided the necessary computational capacity for efficient deep learning model optimization. The traditional vulnerability assessment methods were implemented in both sequential and parallel versions, with the parallel versions leveraging an AMD Ryzen 7 processor with 16 cores to distribute computational load across multiple threads. To provide a fair and practical comparison that reflects real-world deployment scenarios in humanitarian contexts, all inference evaluations—for both GNN models and traditional methods—were performed exclusively on CPU hardware. This approach ensures that the computational advantages demonstrated by our GNN methodology represent achievable performance on standard computing equipment that might be available during disaster response operations in the field.

5.2 Best models

Using the three-stage framework and Optuna, mentioned in Section 3.3.4, we search for the best model for the critical scores of global efficiency, GNN_{GE}, and for the number of independent paths, GNN_{IP}. The objective function for Optuna is maximizing the metric $MAP@K$ presented in equation 4.3. The results in this section are shown for the validation dataset.

5.2.1 Exploration Stage

Time Analysis

- GNN_{GE}: 2.8 hour of training on NVIDIA RTX 3090 GPU.
- GNN_{IP}: 2.8 hour of training on NVIDIA RTX 3090 GPU.
- Total GPU hours: 5.6.

Our hyperparameter exploration reveals different optimal architectures depending on the criticality score being predicted.

Global Efficiency

As shown in Table 5.1, the best-performing models for Global Efficiency systematically utilize GAT-based message passing layers, achieving MAP@K scores of up to 0.794. These models favor shallow network architectures (1 MLP layer) coupled with multiple attention heads (8), with the pairwise approach using square root exponential loss function showing superior performance. A compact embedding size (64) combined with concatenation operations in the embedding module proved most effective, indicating that the attention mechanism effectively captures the complex relationships that determine link criticality in transportation networks.

MAP@K	Batch Size	Dropout	Embedding Op	Emb Size	GAT Heads	GAT v2	Loss Fn	Loss Phi	Learning Rate	Msg Passing	Layers
0.7943	**32**	**0.20**	**concat**	**64**	**8**	**True**	**RANK**	**sqrt_exp**	**0.00907**	**GAT**	**1/2**
0.7941	32	0.22	concat	64	4	True	RANK	sqrt_exp	0.00855	GAT	1/2
0.7912	32	0.24	concat	64	2	True	RANK	sqrt_exp	0.00987	GAT	1/2
0.7906	32	0.22	concat	64	2	True	RANK	sqrt_exp	0.00772	GAT	1/2
0.7900	32	0.22	concat	64	2	True	RANK	sqrt_exp	0.00890	GAT	1/2
0.7896	32	0.22	concat	64	2	True	RANK	sqrt_exp	0.00697	GAT	1/2
0.7893	32	0.22	concat	64	2	True	RANK	sqrt_exp	0.00882	GAT	1/2
0.7892	32	0.22	concat	64	2	True	RANK	sqrt_exp	0.00968	GAT	1/2
0.7890	32	0.22	concat	64	2	True	RANK	sqrt_exp	0.00868	GAT	1/2

Table 5.1: Exploration stage for GNN_{GE}. TOP 10 of 200 trials. Each trial was trained for 10 epochs with a configuration. Pruning bad configurations was allowed. The Layers column shows MLP layers/message passing layers. Training executed with generated data from Appendix A.

Number of Independent Paths

Conversely, as shown in Table 5.2, GCN-based architectures demonstrate superior performance for predicting the Number of Independent Paths criticality metric, achieving $MAP@K$ scores of up to 0.862. These models consistently prefer larger embedding dimensions (256) and deeper network architectures (3 MLP layers with 2 message passing layers). The pairwise ranking approach with square root exponential loss function consistently appears in the top configurations, along with concatenation operations in the embedding module. This suggests that the uniform neighborhood aggregation mechanism of GCN effectively captures the structural connectivity patterns that determine path redundancy in transportation networks.

MAP@K	Batch Size	Dropout	Embedding Op	Emb Size	Loss Fn	Loss Phi	Learning Rate	Msg Passing	Layers
0.8620	**64**	**0.20**	**concat**	**256**	**RANK**	**sqrt_exp**	**0.00471**	**GCN**	**3/2**
0.8618	64	0.22	concat	256	RANK	sqrt_exp	0.00485	GCN	2/2
0.8617	64	0.22	concat	256	RANK	sqrt_exp	0.00234	GCN	5/2
0.8611	64	0.22	concat	256	RANK	sqrt_exp	0.00453	GCN	3/2
0.8608	64	0.34	concat	256	RANK	sqrt_exp	0.00263	GCN	1/2
0.8606	64	0.20	concat	256	RANK	sqrt_exp	0.00510	GCN	3/2
0.8604	64	0.36	concat	256	RANK	sqrt_exp	0.00272	GCN	1/2
0.8602	64	0.20	concat	256	RANK	sqrt_exp	0.00869	GCN	3/2
0.8598	64	0.34	concat	256	RANK	sqrt_exp	0.00571	GCN	5/2
0.8594	64	0.22	concat	256	RANK	sqrt_exp	0.00294	GCN	2/2

Table 5.2: Exploration stage for GNN_{IP}. TOP 10 of 200 trials. Each trial was trained for 10 epochs with a configuration. Pruning bad configurations was allowed. The Layers column shows MLP layers/message passing layers. Training executed with generated data from Appendix A.

5.2.2 Refinement Stage

Time Analysis

- GNN_{GE}: 2.4 hour of training on NVIDIA RTX 3090 GPU.
- GNN_{IP}: 2.5 hour of training on NVIDIA RTX 3090 GPU.
- Total GPU hours: 4.9

The refinement stage further optimized promising hyperparameter configurations identified during exploration, revealing important architectural patterns for each criticality metric while improving overall performance.

Global Efficiency

As shown in Table 5.3, the refinement stage for Global Efficiency prediction yielded GAT models with improved MAP@K scores reaching 0.814. These refined models consistently utilize 8 attention heads across all top configurations, demonstrating the critical importance of the attention mechanism for identifying efficiency-based vulnerability. The best-performing model adopts a deeper message passing architecture (1 MLP layer with 3 message passing layers) compared to the exploration phase, suggesting that additional layers of attention improve the model's ability to capture complex network relationships. All top models maintain consistent batch sizes of 32 and operate within a narrow learning rate range (0.00784-0.00978), indicating convergence toward an optimal configuration for efficiency-based criticality detection.

MAP@K	Batch Size	Dropout	GAT Heads	Learning Rate	Layers (MLP/Msg)
0.814	**32**	**0.30**	**8**	**0.00784**	**1/3**
0.809	32	0.20	8	0.00900	2/2
0.806	32	0.20	8	0.00962	2/2
0.806	32	0.22	8	0.00866	1/2
0.805	32	0.22	8	0.00876	2/2
0.805	32	0.20	8	0.00907	2/2
0.805	32	0.20	8	0.00978	2/2
0.804	32	0.28	8	0.00816	3/2
0.804	32	0.20	8	0.00957	2/2
0.803	32	0.20	8	0.00975	2/2

Table 5.3: Refine stage for GNN_{GE}. TOP 10 of 100 trials. The Layers column shows MLP layers/message passing layers. Training executed with generated data from Appendix A.

Number of Independent Paths

As demonstrated in Table 5.4, the refinement stage for Independent Paths prediction produced highly consistent GCN-based models with MAP@K scores up to 0.871. All top-performing configurations show remarkable architectural similarity, using batch size 64, embedding size 256, and a consistent 3 MLP layers with 2 message passing layers. The dropout rate converges tightly around 0.22, with minor variations (0.20-0.28) among top performers. This striking consistency in optimal hyperparameters confirms our hypothesis that connectivity-based criticality metrics benefit significantly from uniform neighborhood aggregation provided by GCN architectures, which effectively capture the structural patterns that determine path redundancy in transportation networks.

MAP@K	Batch Size	Dropout	Embedding Size	Learning Rate	Layers (MLP/Msg)
0.871	**64**	**0.22**	**256**	**0.00550**	**3/2**
0.869	64	0.22	256	0.00550	3/2
0.869	64	0.22	256	0.00547	3/2
0.869	64	0.22	256	0.00540	3/2
0.868	64	0.22	256	0.00546	3/2
0.867	64	0.20	256	0.00628	3/2
0.867	64	0.28	256	0.00316	3/2
0.866	64	0.22	256	0.00534	3/2
0.866	64	0.20	256	0.00533	3/2
0.866	64	0.22	256	0.00546	3/2

Table 5.4: Refine stage for GNN_{IP}. TOP 10 of 100 trials. The Layers column shows MLP layers/message passing layers. Training executed with generated data from Appendix A.

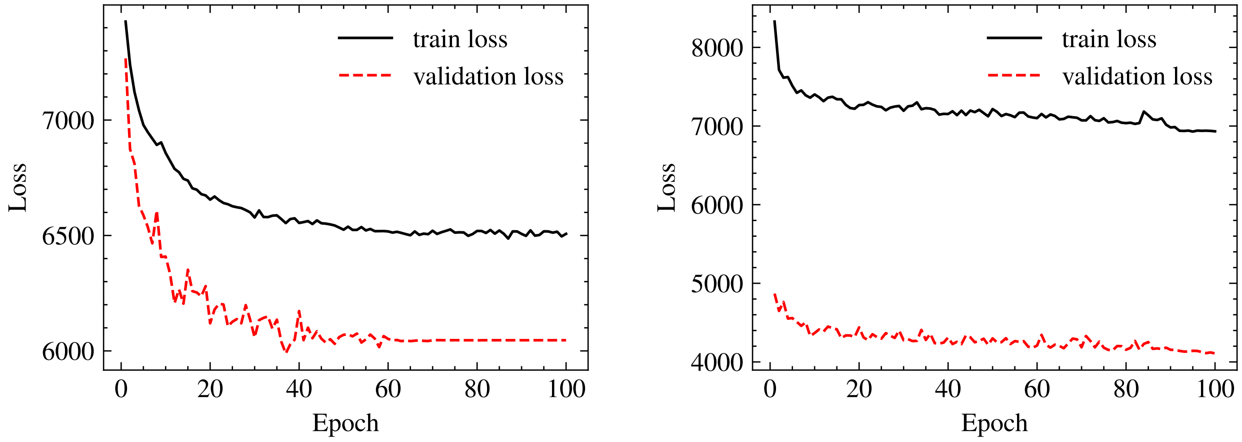
5.2.3 Final Stage

Time Analysis

- GNN_{GE}: 1 hour of training on NVIDIA RTX 3090 GPU.
- GNN_{IP}: 1 hour of training on NVIDIA RTX 3090 GPU.
- Total GPU hours: 2.

For the final stage, we conducted extensive training of 10 models for each criticality score, using the optimal configurations identified in the refinement stage with minor variations. Each model was trained for 100 epochs with early stopping to prevent overfitting. Figure 5.1 illustrates the training and validation loss curves along with the optimal hyperparameter configurations for both models. As evident from the plots, both models demonstrate stable convergence with validation loss consistently decreasing and plateauing without significant signs of overfitting.

The final models achieved MAP@K scores of 0.893 for global efficiency using the GAT architecture and 0.917 for independent paths using the GCN architecture, confirming our architectural divergence hypothesis across the entire optimization process. The significant difference in optimal embedding sizes (64 for GNN_{GE} versus 256 for GNN_{IP}) and network architecture (attention-based for GNN_{GE} versus uniform aggregation for GNN_{IP}) further reinforces the fundamental difference in how these models capture network vulnerability patterns.



(a) Global Efficiency (GNN_{GE}): Dropout=0.28, Embedding Size=64, GAT Heads=8, Message Passing=GAT (v2), MLP/Message Passing Layers=1/3, RANK loss with sqrt_exp transformation, Learning Rate=0.00731.

(b) Independent Paths (GNN_{IP}): Dropout=0.22, Embedding Size=256, Message Passing=GCN, MLP/Message Passing Layers=2/2, RANK loss with sqrt_exp transformation, Learning Rate=0.00893.

Figure 5.1: Train and validation loss during training for best GNN models

5.3 Model Performance Against Ground Truth Criticality Scores

Our evaluation focused on how accurately the developed GNN models could predict link criticality rankings compared to ground truth scores calculated using traditional vulnerability indices. Table 5.5 presents a comprehensive evaluation of model performance using multiple ranking quality metrics.

	Global Efficiency			**Independent Paths**		
	MAP@K	**NDCG@K**	**Kendall tau**	**MAP@K**	**NDCG@K**	**Kendall tau**
GNN_{GE}	**0.90**	**0.91**	**0.31**	0.89	0.86	**0.42**
GNN_{IP}	0.78	0.76	0.22	**0.90**	**0.89**	0.41

Table 5.5: Performance comparison of GNN models on testing dataset. Values (rounded to 2 decimals) represent ranking metrics when predicting each criticality score. Bold values indicate best performance for each metric and criticality score type.

5.3.1 Global Efficiency Prediction Performance

For global efficiency predictions, the specialized GAT-based model (GNN_{GE}) achieved superior performance with MAP@K of 0.90, NDCG@K of 0.91, and Kendall tau of 0.31. This model significantly outperformed the GNN_{IP} model when predicting global efficiency criticality, with improvements of approximately 15% in MAP@K and NDCG@K scores. These results demonstrate the effectiveness of our tailored architectural approach using attention mechanisms for efficiency-based metrics, where the ability to assign varying importance to different neighboring nodes proves crucial for accurate criticality assessment.

5.3.2 Independent Paths Prediction Performance

When predicting independent paths criticality, the GCN-based model (GNN_{IP}) demonstrated excellent performance with MAP@K of 0.90, NDCG@K of 0.89, and Kendall tau of 0.41. The uniform neighborhood aggregation mechanism of GCN appears particularly well-suited for capturing the structural patterns that determine path redundancy in transportation networks. Interestingly, the GNN_{GE} model also performed reasonably well on independent paths prediction (MAP@K of 0.89), indicating some transferability of learned patterns between different vulnerability measures.

5.3.3 Cross-Model Analysis

The Kendall tau values (0.31-0.42) are notable given the complexity of the ranking task and the variability in network topologies. Both models excel at identifying the most critical links even when the complete ranking shows variation, which is valuable for humanitarian applications where prioritizing the most vulnerable components is crucial.

A significant finding is the asymmetric transferability between models: the GAT-based GNN_{GE} generalizes well to independent paths prediction (achieving comparable MAP@K scores to the specialized model), while the GCN-based GNN_{IP} performs poorly on global efficiency prediction. This suggests that attention mechanisms capture broader structural patterns relevant to both metrics, while uniform aggregation in GCN is more specifically suited to connectivity-based measures.

5.4 Global Importance Index Analysis

To assess real-world effectiveness of our GNN models, we analyzed how sequential removal of links identified as critical affects network performance through the Global Importance Index (GII) as outlined in Section 4.2.2. Figure 5.2 displays the normalized GII curves for both GNN models alongside traditional vulnerability indices across multiple disruption scenarios, with Area Under the Curve (AUC) values provided for quantitative comparison.

5.4.1 Independent Paths Model Performance

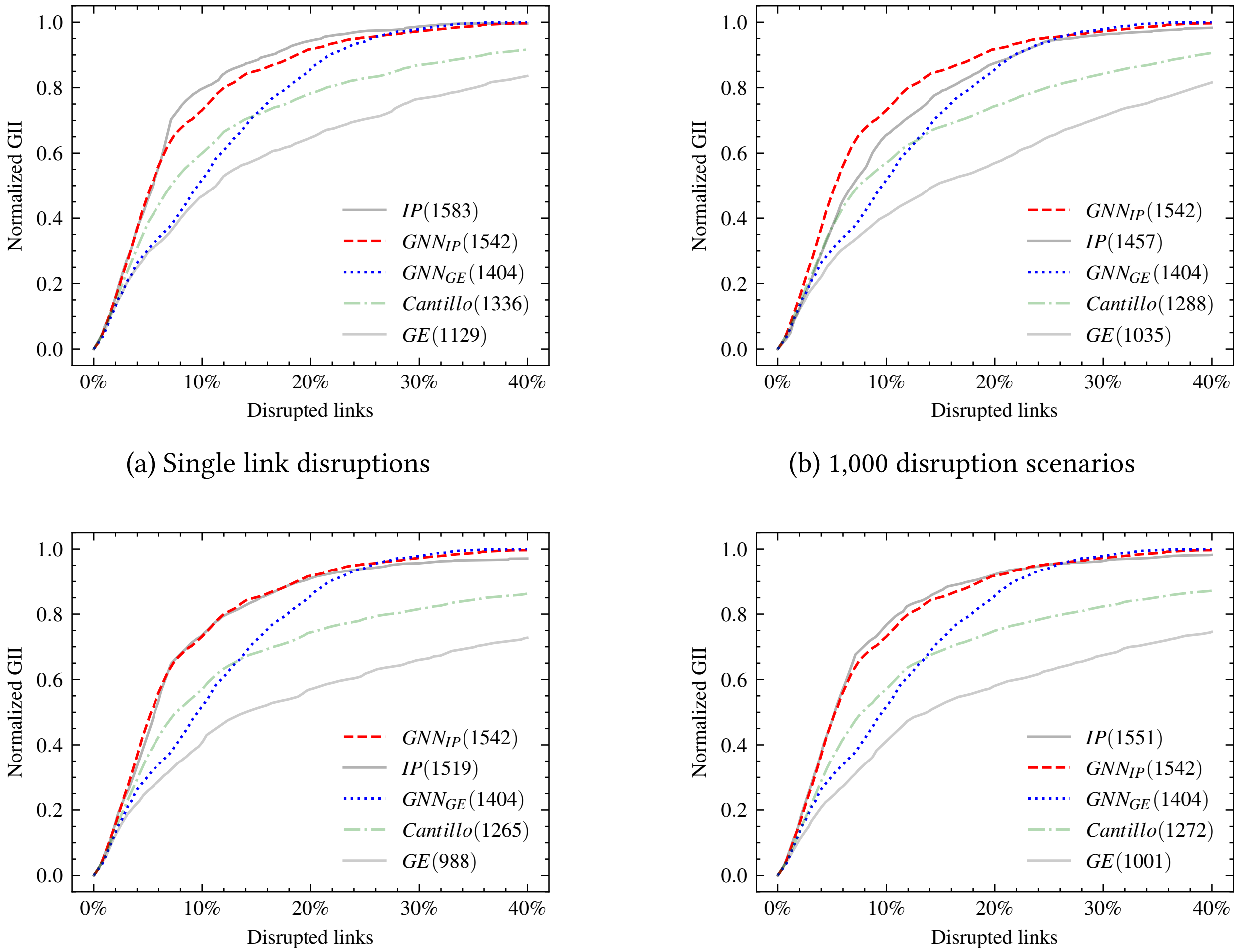
Our GNN_{IP} model achieved remarkable performance (AUC = 1542), nearly matching the ground truth independent paths (IP) index (AUC ranging from 1457 to 1583 across scenarios) and consistently outperforming other approaches. The model successfully replicated the steep initial slope characteristic of the IP index, indicating rapid network degradation when removing the most critical links—a crucial property for humanitarian applications. This near-identical performance to the ground truth metric demonstrates the GCN architecture's effectiveness in capturing the structural patterns that determine path redundancy in transportation networks.

5.4.2 Global Efficiency Model Performance

The GNN_{GE} model (AUC = 1404) significantly outperformed the traditional global efficiency index (AUC ranging from 988 to 1129), demonstrating how our GAT-based architecture enhances critical link identification beyond conventional methods. The attention mechanism appears to identify critical links that more severely impact network performance than those identified by the standard global efficiency metric, suggesting that attention-weighted neighborhood aggregation captures subtle interdependencies between network components that traditional measures may overlook.

5.4.3 Cross-Scenario Consistency

An important observation is the consistent performance of our GNN models across varying numbers of disruption scenarios (from single-link to 10,000 links scenarios), while traditional methods show greater variability. This highlights a key advantage of our approach: once trained, GNN models maintain reliable performance regardless of scenario complexity, making them particularly valuable in time-sensitive humanitarian response situations where computational efficiency is paramount.



(a) Single link disruptions

(b) 1,000 disruption scenarios

(c) 5,000 disruption scenarios

(d) 10,000 disruption scenarios

Figure 5.2: Global Importance Index Curves with corresponding Area Under the Curve (AUC, in parenthesis) values calculated using the trapezoidal approximation method defined in Equation 4.6. The Global Importance Index was subjected to min-max normalization to ensure comparability across different scales. Results represent the average of 155 networks from the testing dataset across all network topologies, with linear interpolation applied to standardize comparison.

5.5 Computation Analysis

The computational efficiency of vulnerability assessment methods is critical for practical application in humanitarian logistics, particularly for rapid decision-making during disaster response.

Figures 5.3 and 5.5 illustrate the execution time comparison between our GNN approach and traditional methods (both sequential and parallel implementations) across varying network sizes and disruption scenarios.

5.5.1 Performance with Network Size

As demonstrated in Figure 5.3, our GNN approach exhibits dramatically superior scalability with respect to network size. For networks with 4,000 edges, traditional sequential methods required approximately 1,000 seconds, while parallel implementations needed roughly 100 seconds. In stark contrast, our GNN approach completed the same analysis in less than 0.01 seconds, a performance improvement of four orders of magnitude.

Even more importantly, the GNN approach demonstrates near-constant time complexity regardless of network size, maintaining sub-second inference times across all tested configurations after the initial training phase. This constant-time performance characteristic makes our approach particularly valuable for analyzing large-scale infrastructure networks in urban areas or regional transportation systems, where traditional methods would be computationally prohibitive.

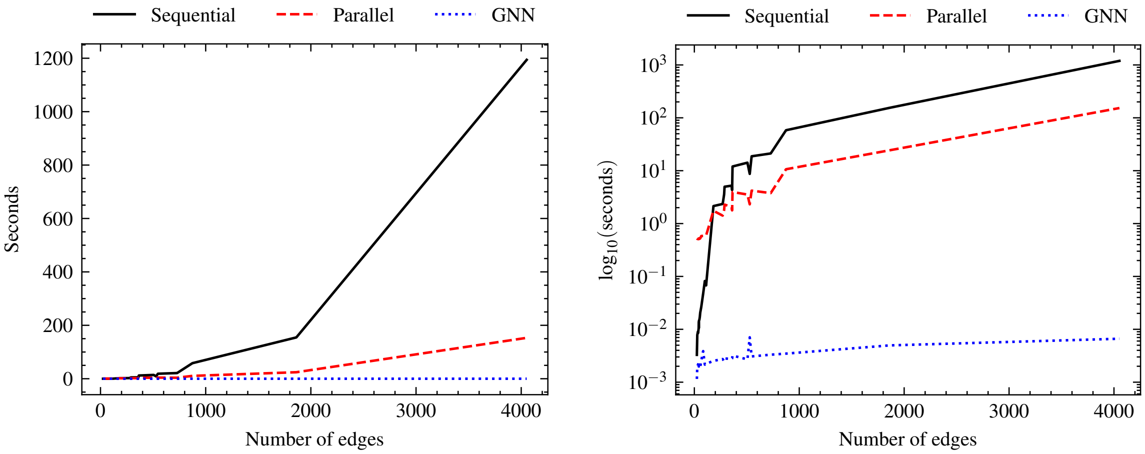


Figure 5.3: Time complexity for network size. Models are grouped by general framework: GNN (average of GNN_{GE} and GNN_{IP}), Sequential (average of GE and IP), and Parallel (average of $\text{GE}_{\text{parallel}}$ and $\text{IP}_{\text{parallel}}$). The real and logarithmic scale highlights the performance difference across 50 networks ranging from 24 to over 4,000 edges, all evaluated with single-link disruption scenarios.

5.5.2 Performance with Number of Source and Terminal Nodes

When analyzing the impact of source and terminal node counts on computational efficiency, our GNN approach again demonstrates remarkable performance advantages, as illustrated in Figure 5.4. For traditional methods, execution time increases substantially as source and terminal nodes grow, with sequential implementations requiring up to 35 seconds for networks with 80 source/terminal nodes, while parallel methods still need approximately 13 seconds. This degradation occurs because traditional methods must evaluate paths between all source-terminal pairs, scaling as O(|S|×|T|).

In stark contrast, our GNN approach maintains consistent sub-second performance (approximately 0.003 seconds) regardless of how many source and terminal nodes are present. This source-terminal independence represents a critical advantage for humanitarian response applications, where supply points and affected communities can vary dramatically between different disaster scenarios. Our approach enables comprehensive vulnerability assessment regardless of supply-demand relationship complexity.

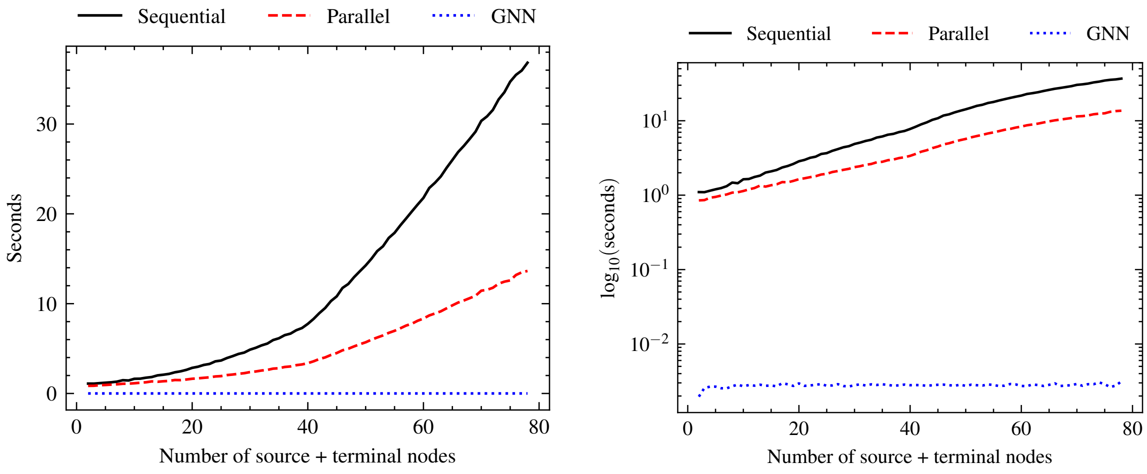


Figure 5.4: Time complexity for number of source and terminal nodes. Models are grouped by general framework: GNN (average of GNN_{GE} and GNN_{IP}), Sequential (average of GE and IP), and Parallel (average of $\text{GE}_{\text{parallel}}$ and $\text{IP}_{\text{parallel}}$). The real and logarithmic scale highlights the performance difference across different number of source and terminal nodes for a grid network of 81 nodes.

5.5.3 Performance Across Disruption Scenarios

When analyzing execution time with respect to the number of disruption scenarios (Figure 5.5), traditional methods show a clear linear relationship between scenario count and computation time. Evaluating 10,000 scenarios required approximately 1 second per scenario for sequential processing, with parallel processing offering only moderate improvement. The GNN approach, however, maintains consistent sub-0.01 second performance regardless of scenario count, as its inference time is independent of the number of scenarios considered—the architectural patterns learned during training implicitly account for various disruption possibilities.

This computational independence from scenario count represents a fundamental advantage for humanitarian response planning, where multiple disruption hypotheses must often be rapidly evaluated to determine resource allocation priorities. The ability to instantaneously assess vulnerability under varying disruption assumptions enables dynamic replanning as conditions evolve during disaster response operations.

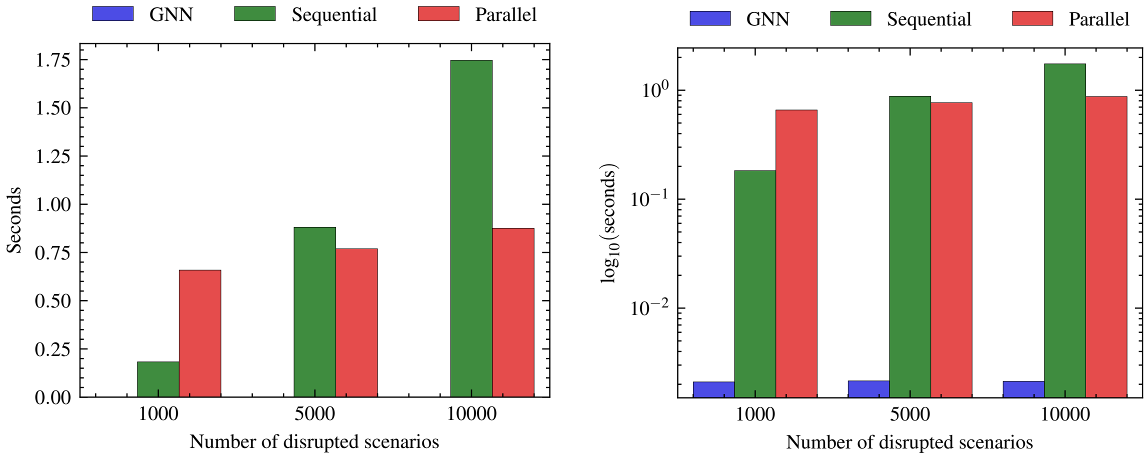


Figure 5.5: Time complexity by number of disruption scenarios. Models are grouped by framework: GNN, Sequential, and Parallel implementations. The real and logarithmic scale shows average execution time across 100 networks, each evaluated with 1,000, 5,000, and 10,000 scenarios.

5.5.4 Computational Performance Summary

Our GNN approach demonstrates transformative improvements across all computational dimensions:

Metric	**Sequential**	**Parallel**	**GNN**	**Improvement**
Network Size (4000 edges)	1000s	100s	< 0.01s	**10,000×**
Source/Terminal Scaling	$O(S \times T)$	$O(S \times T)$	$O(1)$	**Independent**
Disruption Scenarios	$O(D)$	$O(D)$	$O(1)$	**Independent**

Table 5.6: Computational Performance Comparison Across Methods

Key Performance Advantages

1. **Network Size Independence**: While traditional methods scale polynomially with network size, our GNN maintains constant-time performance regardless of network complexity.

2. **Source-Terminal Agnostic**: Traditional methods require $O(|S| \times |T|)$ path evaluations; our approach is completely independent of supply-demand configuration.

3. **Disruption Scenario Scalability**: Traditional methods must re-evaluate each scenario; our models implicitly account for all scenarios learned during training.

4. **Memory Efficiency**: GNN inference requires minimal memory compared to traditional methods that must store and process multiple network states.

This represents a **paradigm shift** from computationally prohibitive traditional approaches to real-time vulnerability assessment capabilities essential for humanitarian response operations. This dramatic reduction in computational requirements without sacrificing accuracy (as demonstrated in Sections 5.3 and 5.4) represents a transformative advancement for vulnerability assessment in humanitarian contexts. By enabling near-instant analysis of even extensive transportation networks, our approach facilitates rapid response planning and real-time decision support during disaster scenarios—capabilities that were previously unattainable due to computational constraints.

Chapter 6

Conclusions and Future Work

This research addressed a critical computational challenge in transportation network vulnerability assessment for humanitarian logistics: the prohibitive computational cost of identifying critical links in large-scale networks. We developed a novel Graph Neural Network (GNN) approach that overcomes this limitation, enabling vulnerability assessment at previously infeasible network scales. Our contributions can be summarized in three key areas.

First, we demonstrated that different network vulnerability metrics benefit from specialized GNN architectures—GAT-based models excel at capturing efficiency-based criticality through attention mechanisms, while GCN architectures better model connectivity patterns for independent paths-based vulnerability assessment.

Second, our comprehensive validation using the Global Importance Index confirmed that our models successfully identify links whose removal maximally degrades network performance, with the GNN_{IP} model achieving performance comparable to traditional vulnerability indices but at a fraction of the computational cost.

Finally, our computational analysis revealed a transformative improvement in efficiency, with GNN-based critical link identification executing up to four orders of magnitude faster than traditional methods for large networks.

The practical implications of this research are significant for humanitarian logistics. By reducing vulnerability assessment from hours or days to mere seconds, our methodology enables rapid response planning during disaster scenarios. Decision-makers can now quickly identify and prioritize protection of critical infrastructure components, potentially improving resource allocation and accelerating aid delivery to affected communities.

Future Work

Several promising research directions emerge from this work:

1. **Conditional Multi-Model Framework:** While our research demonstrated specialized models for different vulnerability metrics, future work could explore a conditional approach that intelligently selects or combines multiple focused models based on network characteristics or specific humanitarian contexts. This could yield a unified vulnerability assessment tool adaptable to diverse operational scenarios.

2. **Hazard-Specific Disruption Modeling:** Our current approach focuses on general link criticality, but transportation networks face diverse hazards with differing spatial and temporal characteristics. Future research could incorporate hazard-specific disruption scenarios (e.g., floods affecting lowland areas, earthquakes targeting structurally vulnerable links, or wildfires threatening forested corridors) to develop more contextualized vulnerability assessments.

3. **Dynamic Vulnerability Assessment:** Current approaches, including ours, typically provide static snapshots of network vulnerability. Extending our GNN methodology to incorporate temporal dynamics could enable modeling of evolving vulnerability during disaster progression, recovery operations, or changing environmental conditions.

4. **Integration with Humanitarian Logistics Optimization:** While our work focuses on identifying critical links, future research could integrate this capability with broader humanitarian logistics optimization frameworks. This integration could create end-to-end systems that not only identify vulnerable infrastructure but also automatically suggest optimal resource allocation and routing strategies that minimize risk.

5. **Transfer Learning Across Network Types:** Investigating whether vulnerability patterns learned from one transportation network can transfer to others could enhance model

efficiency and address data scarcity challenges in humanitarian contexts. This could be particularly valuable for rapidly assessing vulnerability in regions with limited historical data or during sudden-onset disasters.

These future directions aim to further bridge the gap between theoretical vulnerability assessment and practical humanitarian response, ultimately enhancing the resilience and effectiveness of transportation networks during disasters when they are most critically needed.

Appendix A

Data generation of synthetic networks

The network dataset generation process follows a systematic approach to create a diverse collection of transportation network structures with varying topological characteristics, sizes, and operational parameters. This methodology enables comprehensive vulnerability analysis across different network types that might be encountered in humanitarian logistics scenarios.

Generation Framework

The dataset is generated through a multi-layered parametric process that creates networks by systematically varying:

1. **Network Topology Types** (11 distinct topologies)
2. **Size Categories** (3 size ranges: small, medium, large)
3. **Parameter Sets** (3 variations per topology)
4. **Random Seeds** (2 different seeds for reproducibility)
5. **Origin-Destination (OD) Patterns** (5 patterns)
6. **Edge Weight Configurations** (5 variations)

This combinatorial approach generates a comprehensive dataset covering a wide spectrum of network characteristics, which is crucial for transportation network vulnerability analysis where topology and congestion patterns significantly impact link criticality rankings.

Network Topology Selection

The dataset includes the following topologies:

- **Regular Structures:**
  - Grid networks: Regular lattice arrangement
  - Cycle networks: Simple circular arrangement
  - Star networks: Central node connected to all other nodes
  - Wheel networks: Cycle with a central node connected to all perimeter nodes
  - Frucht graphs: 12-node 3-regular graph with specific properties
  - Heawood graphs: Bipartite graph with distinctive characteristics
- **Complex Network Models:**
  - Barbell graphs: Two complete graphs connected by a path
  - Watts-Strogatz networks: Small-world networks with high clustering and short path lengths
  - Random Geometric networks: Nodes placed randomly in a metric space
  - Scale-Free networks: Networks with power-law degree distribution
  - Barabási-Albert networks: Scale-free networks generated through preferential attachment

These topologies represent a spectrum of network characteristics with varying density, efficiency, and transitivity - factors that have been identified as significant in determining network resilience in transportation systems.

Node Assignment Strategy

Each generated network includes two critical node types that reflect humanitarian logistics requirements:

- **Source nodes**: Representing distribution points (5-15% of total nodes)
- **Terminal nodes**: Representing affected communities requiring assistance (20-40% of total nodes)

This distribution mirrors realistic transportation networks where origin points tend to be fewer than destination points, enabling analysis of network vulnerability with respect to accessibility to critical facilities.

Edge Weight Assignment

A multi-attribute weighting scheme is employed using three factors:

- Time factor (50% ±20% weight)
- Distance factor (30% ±10% weight)
- Cost factor (20% ±10% weight)

This approach enables realistic modeling of transportation networks where multiple criteria influence path selection, similar to multi-criteria approaches used in humanitarian logistics for link criticality assessments.

Network Metrics Calculation

For each generated network, key topological metrics are calculated:

- **Transitivity**: Measure of clustering in the network
- **Density**: Ratio of actual to potential connections

These metrics provide a basis for classifying networks and are crucial for understanding how network topology influences the consistency of criticality metrics in transportation vulnerability analysis.

References

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Cantillo, Victor, Luis F Macea, and Miguel Jaller (2018). "Assessing vulnerability of transportation networks for disaster response operations". In: *Networks and Spatial Economics* 19.1, pp. 243–273. DOI: 10.1007/s11067-017-9382-x.

Faturechi, Reza and Elise Miller-Hooks (2014). "Measuring the performance of transportation infrastructure systems in disasters: A comprehensive review". In: *Journal of infrastructure systems* 21.1, p. 04014025. DOI: 10.1061/(ASCE)IS.1943-555X.0000212.

Ghahremani-Nahr, J., H. Nozari, and A. Szmelter-Jarosz (2024). "Designing a humanitarian relief logistics network considering the cost of deprivation using a robust-fuzzy-probabilistic planning method". In: *International Journal of Humanitarian Action* 9.19. DOI: 10.1186/s41018-024-00163-8. URL: https://doi.org/10.1186/s41018-024-00163-8.

Haritha, P.C. and M.V.L.R. Anjaneyulu (2024). "Comparison of topological functionality-based resilience metrics using link criticality". In: *Reliability Engineering System Safety* 243, p. 109881. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2023.109881. URL: https://www.sciencedirect.com/science/article/pii/S0951832023007950.

Holguín-Veras, José, Noel Pérez, Miguel Jaller, Luk N Van Wassenhove, and Felipe Aros-Vera (2013). "On the appropriate objective function for post-disaster humanitarian logistics models". In: *Journal of Operations Management* 31.5, pp. 262–280. DOI: 10.1016/j.jom.2013.06.002.

Ip, Wai Hung and Dingwei Wang (2011). "Resilience and friability of transportation networks: evaluation, analysis and optimization". In: *IEEE Systems Journal* 5.2, pp. 189–198. DOI: 10.1109/JSYST.2010.2096670.

Jafino, Bramka Arga, Jan Kwakkel, and Alexander Verbraeck (2020). “Transport network criticality metrics: a comparative analysis and a guideline for selection”. In: *Transport Reviews* 40.2, pp. 241–264. DOI: 10.1080/01441647.2019.1703843.

Jenelius, Erik, Tom Petersen, and Lars-Göran Mattsson (2006). “Importance and exposure in road network vulnerability analysis”. In: *Transportation Research Part A: Policy and Practice* 40.7, pp. 537–560. DOI: 10.1016/j.tra.2005.11.003.

Jiang, Weiwei and Jiayun Luo (2022). “Graph neural network for traffic forecasting: A survey”. In: *Expert Systems with Applications* 207, p. 117921. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.117921. URL: https://www.sciencedirect.com/science/article/pii/S0957417422011654.

Kashav, Vishal and Chandra Prakash Garg (Nov. 2024). “Fortifying humanitarian supply chains: evaluating sustainability enablers for strengthened resilience of humanitarian supply chains during calamities and pandemics”. In: *Journal of Humanitarian Logistics and Supply Chain Management*. Open Access. ISSN: 2042-6747.

Latora, Vito and Massimo Marchiori (2001). “Efficient behavior of small-world networks”. In: *Physical review letters* 87.19, p. 198701. DOI: 10.1103/PhysRevLett.87.198701.

Mattsson, Lars-Göran and Erik Jenelius (2015). “Vulnerability and resilience of transport systems–A discussion of recent research”. In: *Transportation Research Part A: Policy and Practice* 81, pp. 16–34. DOI: 10.1016/j.tra.2015.06.002.

Munikoti, Sai, Laya Das, and Balasubramaniam Natarajan (2022). “Scalable graph neural network-based framework for identifying critical nodes and links in complex networks”. In: *Neurocomputing* 468, pp. 211–221. DOI: 10.1016/j.neucom.2021.10.053.

Sullivan, James L, David C Novak, Lisa Aultman-Hall, and Darren M Scott (2010). “Identifying critical road segments and measuring system-wide robustness in transportation networks with isolating links: A link-based capacity-reduction approach”. In: *Transportation Research Part A: Policy and Practice* 44.5, pp. 323–336. DOI: 10.1016/j.tra.2010.02.003.

Taylor, Michael A. P. and Glen M. D’Este (2005). “Transport Network Vulnerability: a Method for Diagnosis of Critical Locations in Transport Infrastructure Systems”. In: *Critical Infrastructure: Reliability and Vulnerability*. Ed. by Alan T. Murray and Tony H. Grubesic. Berlin, Heidelberg: Springer, pp. 9–30. DOI: 10.1007/978-3-540-68056-7_2.

Taylor, Michael A.P. (2017). “Chapter One - Introduction”. In: *Vulnerability Analysis for Transportation Networks*. Ed. by Michael A.P. Taylor. Elsevier, pp. 1–17. ISBN: 978-0-12-811010-2. DOI: https://doi.org/10.1016/B978-0-12-811010-2.00001-0. URL: https://www.sciencedirect.com/science/article/pii/B9780128110102000010.

Wu, Zonghan, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu (2021). “A comprehensive survey on graph neural networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1, pp. 4–24. DOI: 10.1109/TNNLS.2020.2978386.

Zhang, X., E. Miller-Hooks, and K. Denny (2015). “Assessing the role of network topology in transportation network resilience”. In: *Journal of Transport Geography* 46, pp. 35–45. ISSN: 0966-6923. DOI: https://doi.org/10.1016/j.jtrangeo.2015.05.006. URL: https://www.sciencedirect.com/science/article/pii/S0966692315000794.

Zhou, Yizhou, John Wang, and Hubo Yang (2019). “Resilience of transportation systems: concepts and comprehensive review”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.12, pp. 4262–4276. DOI: 10.1109/TITS.2018.2883766.