**GEOGRAPHY DOCTORAL SCHOOL**
Faculty of Science

# Criticality as a Learnable Property:

## Scalable Artificial Intelligence Techniques for Critical Link Analysis in Urban Traffic Networks

**Nourhan Bachir**

Supervisor:
Prof. R. Billen – Uliège

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Science (PhD): Geography

September 2025

# Criticality as a Learnable Property: Scalable Artificial Intelligence Techniques for Critical Link Analysis in Urban Traffic Networks

**Nourhan Bachir**

Examination committee:
Prof. Dr. R. Billen, supervisor – Uliège
Prof. Dr. Ch. Claramunt – École navale
Prof. Dr. M. Cools, president – Uliège
Prof. Dr. H. Harb – AUM
Prof. Dr. S. Limbourg – Uliège
Prof. Dr. Ch. Zaki, co-supervisor, secretary – AUM

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Science (PhD): Geography

September 2025

# Acknowledgements

This thesis marks the culmination of an intense and rewarding journey, one that I could not have completed without the support, guidance, and encouragement of many remarkable individuals.

First and foremost, I would like to express my deepest gratitude to my supervisors, **Prof. Roland Billen**, **Prof. Chamseddine Zaki**, and **Prof. Hassan Harb**, for their unwavering support, insightful feedback, and continuous guidance throughout my research. Their mentorship has not only shaped this thesis but has also profoundly influenced my growth as a researcher. I am especially grateful for the significant opportunity you provided and your unwavering belief in me throughout this journey.

I would also like to thank **Prof. Mario Cools** for the enriching conversations and thought-provoking discussions that have helped broaden my perspective and refine my ideas.

My appreciation extends to my colleagues and friends at the department. I am especially grateful to **Gilles-Antoine**, **Charline**, and **Imane**, whose help and camaraderie made the journey much more manageable and enjoyable. To everyone else in the department who contributed in their own way—thank you for being part of this chapter.

To **my late mother**, this thesis is for you. I know you would have been proud to see me here. May your love and strength continue to guide me.

To **my father**, my steadfast supporter since my earliest days. You are the calm presence that guides me through everything and pushes me to reach my fullest potential. I was a little girl, and I looked up to you as my superhero. I used to follow you around everywhere. My superhero you shall always be, and may I always follow in your footsteps.

To **my family and friends**, who stood by me during moments of doubt and celebrated every win with me. Thank you for always picking me up and forcing me out of my head to be able to continue. Many days I fell into the holes of doubt and it was you who lifted me out of there. Thank you for believing in me and for all your love and support through everything.

To my partner **Ali**, you came in just at the right time with the support, patience, and love at the time I needed it most. Thank you for always being there with the right words and for believing in me when I failed to do so. You celebrated my littlest wins and comforted and supported me through everything else. I'm grateful to walk this path with you.

To the **godfather** who was always there till the end. I'm forever indebted to you and to all your sacrifices and efforts. For you I dreamed big and because of you I reached the stars. May your light always continue to guide me so.

To my beloved **Lebanon**, to those who sacrificed everything for us to continue our paths. Had it not been for you, I would simply not be here. No words can describe how humbled I stand in-front of all that you sacrificed. May we live and thrive and continue to give all that we have got to the country and people that deserve it all.

# Abstract

Urban transportation networks are the backbone of modern cities, yet they remain highly vulnerable to disruptions such as accidents, road closures, construction, or flooding. A single failure in one part of the network can propagate system-wide, causing substantial delays, accessibility issues, and cascading effects that impact urban mobility. Identifying which road segments are truly critical—those whose disruption would lead to disproportionate network-level impact—is therefore essential for resilient and intelligent traffic management.

Conventional approaches to criticality assessment rely heavily on static graph-based metrics (e.g., betweenness centrality) or localized simulation scenarios. While informative, these methods lack behavioral realism, generalizability, and scalability. They also define criticality in advance, treating it as a fixed index or rule-based output. This thesis challenges that assumption by introducing a new paradigm: modeling criticality as a learnable property—a dynamic outcome that emerges from how links behave in practice and under stress, and that can be predicted using data-driven methods.

To support this paradigm, the thesis develops a modular pipeline that integrates spatial features, behavioral movement patterns, and simulation-based indicators into supervised machine learning models. The learning process is informed by traffic simulations, where criticality is approximated through the difference in total trip time after link removal. This allows the model to learn from system-wide responses to disruption, rather than relying on predefined thresholds. The result is a scalable and interpretable method that generalizes across cities and disruption scenarios.

The thesis follows a three-article structure. The first contribution, PEMAP, proposes a conceptual and modular framework for post-event traffic management in urban networks. It is composed of four successive phases: (1) indices assessment and evaluation, (2) criticality index calculation, (3) rerouting decision making, and (4) decision application via vehicular networks. While the second phase—criticality modeling—is explored in depth, the work also engages with Phase 1 through a structured review of existing indices and tools, which informed the methodological design. PEMAP thus serves as a structural foundation that frames the broader vision of disruption-aware traffic resilience, motivating scalable, data-integrated methods that go beyond static metrics and support intelligent decision-making under stress. The second contribution introduces VeTraSPM, a trajectory mining model tailored to vehicle movement data, which accounts for directionality, road connectivity, and pattern repetition to generate new behavioral indices. SAMO then comes in to validate the integration of these indices—support, confidence, and sequential impact—within machine learning models for critical link prediction. The final contribution, SMaL-CLIP, presents a scalable machine learning pipeline trained on hybrid features—including graph metrics, simulation outputs (e.g., flow, speed, delay), and trajectory-derived scores—to predict link criticality in two large-scale simulated urban networks (Luxembourg and Monaco).

Results show that the proposed approach achieves high predictive performance, even when trained on limited data, and can generalize across different urban contexts. The combination of simulation-informed labels, interpretable features, and scalable modeling contributes to a new generation of learning-based tools for disruption-aware traffic planning.

In conclusion, this thesis demonstrates that criticality is not a fixed attribute, but a learnable, context-sensitive property. By treating critical links as emergent outcomes of structural context, mobility behavior, and disruption impact, the proposed methodology lays a

foundation for scalable, adaptable, and explainable prediction in urban traffic networks.

While this work focuses on the second phase of the PEMAP framework—criticality modeling—it serves as a first stepping stone toward operationalizing the broader vision of disruption-aware traffic resilience. Future research can build on this foundation by implementing the remaining PEMAP phases, including intelligent rerouting strategies and real-time decision dissemination. Additional extensions include validating the SMaL-CLIP pipeline using real-world trajectory data, exploring transferability across diverse urban environments, and embedding these predictive models within real-time decision-support systems for disruption-aware urban mobility planning.

**Keywords:** Critical Link Analysis, Urban Traffic, Machine Learning, Resilience, Intelligent Transportation Systems

# Résumé

Les réseaux de transport urbain constituent l'épine dorsale des villes modernes, mais demeurent particulièrement vulnérables aux perturbations telles que les accidents, les fermetures de routes, les travaux ou les inondations. Une seule défaillance localisée peut se propager à l'ensemble du réseau, provoquant des retards importants, des problèmes d'accessibilité et des effets en cascade qui compromettent la mobilité urbaine. Identifier les segments routiers véritablement critiques—ceux dont la perturbation aurait un impact disproportionné sur le réseau dans son ensemble—est donc essentiel pour une gestion du trafic résiliente et intelligente.

Les approches conventionnelles d'évaluation de la criticité reposent largement sur des métriques statiques issues de la théorie des graphes (par exemple, la centralité d'intermédiarité) ou sur des scénarios de simulation localisés. Bien qu'informatives, ces méthodes manquent de réalisme comportemental, de généralisabilité et de passage à l'échelle. De plus, elles définissent la criticité a priori, comme un indice figé ou un résultat basé sur des règles. Cette thèse remet en question cette hypothèse en introduisant un nouveau paradigme : modéliser la criticité comme une propriété apprenable—un résultat dynamique qui émerge du comportement des liens en pratique et sous contrainte, et qui peut être prédit à l'aide de méthodes guidées par les données.

Pour soutenir ce paradigme, la thèse développe une chaîne de traitement modulaire qui intègre des caractéristiques spatiales, des motifs comportementaux extraits de trajectoires, et des indicateurs issus de simulations dans des modèles d'apprentissage automatique supervisé. Le processus d'apprentissage est alimenté par des simulations de trafic, où la criticité est approchée par la différence de temps de trajet total après la suppression d'un lien. Cela permet au modèle d'apprendre à partir des réponses systémiques à la perturbation, plutôt que de s'appuyer sur des seuils prédéfinis. Il en résulte une méthode interprétable, généralisable et évolutive pour prédire la criticité des segments routiers.

La thèse adopte une structure en trois articles. La première contribution, PEMAP, propose un cadre conceptuel et modulaire pour la gestion post-événement des réseaux de transport urbain. Ce cadre est composé de quatre phases successives : (1) l'évaluation des indices, (2) le calcul de l'indice de criticité, (3) la prise de décision de réacheminement, et (4) l'application de la décision via des réseaux véhiculaires (VANETs). Bien que seule la deuxième phase—la modélisation de la criticité—soit explorée en profondeur dans cette thèse, PEMAP sert de fondation structurelle pour encadrer une vision plus large de la résilience urbaine face aux perturbations. Il motive la nécessité de développer des méthodes scalables et intégrées, capables de dépasser les métriques statiques et de soutenir une prise de décision intelligente en situation de stress. La deuxième contribution introduit VeTraSPM et SAMO, des modèles de fouille de motifs séquentiels appliqués aux trajectoires de véhicules, qui permettent de générer de nouveaux indices comportementaux fondés sur la fréquence, la confiance et la régularité des flux. La dernière contribution, SMaL-CLIP, propose une chaîne d'apprentissage automatique scalable, entraînée sur un ensemble hybride de caractéristiques—comprenant des métriques topologiques, des sorties de simulation (ex. : flux, vitesse, retard) et des scores dérivés des trajectoires—pour prédire la criticité des liens dans deux réseaux urbains simulés à grande échelle (Luxembourg et Monaco).

Les résultats montrent que l'approche proposée atteint de hautes performances prédictives, même en étant entraînée sur des données partielles, et qu'elle est capable de généraliser à différents contextes urbains. La combinaison d'étiquettes informées par la simulation, de

caractéristiques interprétables, et de modèles scalables contribue à une nouvelle génération d'outils prédictifs pour la planification du trafic face aux perturbations.

En conclusion, cette thèse démontre que la criticité n'est pas une propriété fixe, mais une caractéristique apprenable et sensible au contexte. En traitant les liens critiques comme des résultats émergents de la structure du réseau, du comportement de mobilité et de l'impact des perturbations, la méthodologie proposée pose les bases d'une prédiction explicable, adaptable et évolutive de la criticité dans les réseaux routiers urbains.

Bien que ce travail se concentre sur la deuxième phase du cadre PEMAP—la modélisation de la criticité—il constitue une première pierre sur un chemin de recherche plus large vers une gestion urbaine du trafic plus résiliente et réactive. Les travaux futurs pourront s'appuyer sur cette base pour implémenter les autres phases du cadre, notamment les stratégies intelligentes de réacheminement et la diffusion des décisions en temps réel. D'autres extensions incluent la validation du pipeline SMaL-CLIP à l'aide de trajectoires réelles, l'exploration de sa transférabilité à d'autres environnements urbains, et l'intégration des modèles prédictifs dans des systèmes décisionnels opérationnels dédiés aux villes intelligentes.

**Mots-clés :** Analyse de Liens Critiques, Trafic Urbain, Apprentissage Automatique, Résilience, Systèmes de Transport Intelligents

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Preamble

Cities move. Every day, millions of individual journeys unfold across urban spaces—commutes to work, deliveries to restaurants, ambulances weaving through traffic. These movements happen not in isolation, but on shared infrastructure: roads, intersections, signals, and roundabouts stitched together into networks of staggering complexity.

These networks are designed to function under pressure. But like any system, they break. An accident at a key junction, a sudden road closure, or a flooded underpass can bring mobility to a standstill—not just nearby, but kilometers away. A single failure can ripple through the system, triggering what researchers often call the cascading effect. And yet, we rarely ask: which parts of the network truly matter the most? Not just which ones are busy, or central—but which ones, if disturbed, would lead to actual systemic collapse?

This seemingly simple question is surprisingly hard to answer. And it is precisely the question this thesis takes up.

Road networks have long been studied through the lens of graph theory. Nodes and edges offer a natural abstraction of junctions and roads. Measures like betweenness centrality or closeness give us a first approximation of importance. But these models, while elegant, are silent on behavior. They don't know how people drive, when congestion spikes, or how the network reacts when something goes wrong.

Other tools—particularly microscopic traffic simulation—offer a different perspective. Simulation allows us to model how traffic behaves under normal or disrupted conditions, providing access to performance metrics such as delay, rerouting pressure, and queue formation. However, these tools are computationally intensive, time-consuming, and difficult to scale—especially when modeling large urban areas or running multiple disruption scenarios. They also rely on assumed behavior rather than observed behavior, making their realism inherently limited.

Then there is trajectory data—collected from GPS traces, floating car datasets, or vehicular sensors—which offers fine-grained insights into how vehicles actually move through the network. When mined correctly, these data can reveal latent movement patterns, user preferences, and functional hierarchies that are invisible to topological models. Yet in practice, trajectory data is often unavailable or incomplete, particularly in cities without large-scale sensing infrastructure. And when available, such data are voluminous, noisy, and analytically challenging, requiring robust preprocessing and scalable mining strategies.

This is where machine learning offers a promising alternative. By integrating structural, behavioral, and simulation-informed indicators into a unified representation, machine learning can shift the focus from predefined metrics to learned representations. Rather than hard-coding what makes a link "important," we can ask: can criticality of a link be *learned* from its inherent characteristics and how it behaves within an urban traffic network?

Yet despite the growing presence of ML in transportation research, its use for predicting link criticality remains underexplored. Most existing methods rely on static heuristics or handcrafted indices, with little attention to scalability or predictive generalization across urban contexts.

This thesis bridges that gap. It proposes a scalable data-driven approach that unifies simulation-based impact modeling, trajectory-informed behavior mining, and machine learning–based prediction. Rather than viewing criticality as a static indicator, it reframes it as a dynamic outcome—one that can be inferred from data and generalized across networks.

And so the question becomes:

> Can road link criticality be modeled as a learnable property?

This is the question at the heart of this thesis. It does not aim to replace existing tools, but to connect and reconfigure them—linking the explanatory power of graph theory with the realism of simulation, the behavioral richness of trajectory data, and the predictive strength of machine learning.

In the chapters that follow, this integration unfolds in three stages. A conceptual framework is introduced to structure the problem of post-event traffic management. A trajectory-based pattern mining model is developed to uncover movement hierarchies and behavioral signals of link importance. Lastly, a scalable machine learning pipeline is presented to predict road segment criticality across entire networks, using hybrid features that span spatial, simulated, and behavioral dimensions.

Ultimately, this thesis is not just about modeling traffic. It is about understanding how complex systems fail—and designing data-driven tools to recognize and anticipate those failures in a scalable and transferable way.

## 1.2 Context

Urban road networks operate under constant pressure. They must accommodate growing populations, increasingly diverse mobility patterns, and an ever-present risk of disruption. Even minor events—such as a car crash, sudden flooding, or a temporary road closure—can lead to far-reaching cascading effects that paralyze traffic flow across entire districts (Redzuan et al., 2022). These vulnerabilities are magnified in complex urban environments, where interdependencies between road segments, vehicle behaviors, and control systems make the network highly sensitive to localized disturbances (Serdar et al., 2022; Shang et al., 2020).

Ensuring the resilience of urban mobility systems in the face of such disruptions has become a core objective of modern transportation planning and intelligent transportation system (ITS) design. Central to this challenge is the notion of *criticality*—a property that captures how essential a given road segment is to the functionality and performance of the broader network (S. Ahmed & Dey, 2020; Jenelius et al., 2006). Critical links are those whose impairment or removal causes disproportionate degradation in accessibility, connectivity, or flow. Identifying these links is essential not only for long-term infrastructure planning but also for real-time traffic management and emergency response.

Traditionally, criticality has been assessed through *graph-based methods*, which model road networks as static topological structures. Metrics such as betweenness centrality, closeness, degree, or clustering coefficient are used to quantify structural importance (Akbarzadeh et al., 2019; El Rashidy & Grant-Muller, 2014; Feng et al., 2022; Gauthier et al., 2018). While these metrics are computationally efficient and widely adopted, they provide only a limited view: they consider topology but neglect behavior, demand variability, congestion dynamics, and rerouting potential. As a result, structurally central links may not always be functionally critical, and vice versa.

To address these shortcomings, researchers have increasingly adopted *simulation-based approaches*. Microscopic traffic simulators such as SUMO, TransCAD, and VISSIM allow analysts to model how traffic responds to disruptions, rerouting, and signal changes (Elsafdi, 2020; Lee et al., 2022). These tools provide dynamic measures of link importance, including changes in total travel time, rerouting impact, delay propagation, and queue formation (El Rashidy & Grant-Muller, 2014; Jenelius, 2010; Li et al., 2020; Scott et al., 2006). By systematically removing links and recording the resulting effects, one can estimate how critical a link is in terms of its influence on system-wide performance. However, these simulations are computationally intensive—especially when applied across thousands of links—and their scalability is limited. Furthermore, their accuracy depends heavily on behavioral assumptions and traffic demand models, which may not fully reflect real-world conditions.

In response to these limitations, recent work has turned toward leveraging *vehicular trajectory data*. These datasets—collected via GPS traces, floating car logs, mobile sensors, or vehicle-to-infrastructure communication—offer direct insight into how road networks are actually used (Akbarzadeh et al., 2018; Ibrahim & Shafiq, 2019; Yu, 2019). When mined effectively, trajectory data can reveal functional hierarchies, preferred paths, and behavioral flows that are invisible to topology alone. However, trajectory datasets present new challenges: they are often noisy, incomplete, and highly variable across cities. Their utility also depends on robust preprocessing and scalable mining algorithms, which are not trivial to implement. Moreover, in many urban contexts, especially in data-sparse or resource-limited regions, such datasets may be unavailable or too limited for full-network analysis.

Despite the strengths and weaknesses of each of these approaches—structural, simulated, and behavioral—a unifying limitation remains: in nearly all existing frameworks, *criticality is predefined*. It is either assumed as a fixed structural score or evaluated within a specific disruption scenario. This assumption limits adaptability, transferability, and scalability. It also prevents criticality from being modeled as a context-sensitive, data-driven property. And while machine learning has gained prominence in transportation research for tasks like demand forecasting and incident detection (Merah et al., 2012; Moreira-Matias et al., 2012), its application to criticality prediction remains nascent. Most models still rely on handcrafted metrics or simulated stress tests (Elsafdi, 2020), without using learning-based techniques to predict or generalize criticality across networks.

This thesis responds to that gap by introducing *criticality as a learnable property*. Rather than define link importance through static indices or manually tuned formulas, it proposes to learn it from data—specifically, from how a link behaves structurally, behaviorally, and under simulated stress. To supervise this learning process, simulation-based stress testing is used to generate empirical labels, defined as the change in total trip time following link removal. These labels are combined with trajectory-derived indices and spatial features in a hybrid representation. The resulting models are designed not only to assess criticality within a network but to scale across cities and disruption types.

Together, these components provide a robust foundation for a shift in how road network criticality is understood, modeled, and applied. By treating criticality as a learnable property, this thesis not only introduces a new research direction but contributes practical tools for developing intelligent, resilient, and adaptive urban mobility systems.

## 1.3 Research Questions

Understanding which parts of a road network are most critical during disruption is essential for designing resilient urban mobility systems. While various methods have been proposed—from structural metrics to simulation and trajectory mining—most remain siloed, reactive, or difficult to scale. The underlying challenge is not a lack of data, but the absence of unified, generalizable approaches for learning criticality across contexts.

Machine learning offers a promising way forward, enabling models to infer criticality from hybrid data representations that integrate structural, behavioral, and simulation-derived indicators. However, this potential remains largely untapped in the literature. Few studies explore how to represent link importance as a learnable property, or how to generalize predictive insights across different networks.

This thesis is motivated by the need to bridge this gap—by investigating whether and how machine learning can support scalable and data-driven criticality assessment in disrupted urban road networks. The central research question guiding this investigation is:

> Can road link criticality be modeled as a learnable property—rather than a fixed index—through machine learning using simulation-informed, structural, and trajectory-derived features?

To guide this investigation, four specific sub-questions are formulated:

- What is criticality, and how can GIS, simulation, graph theory, and AI be effectively combined to assess it in urban road networks? How can this assessment be operationalized for resilient traffic management?
- What types of data-driven indices best capture road segment importance based on trajectory behavior and disruption impact?
- How can microscopic traffic simulation be used to generate informative and scalable datasets for predictive modeling and evaluation?
- Can machine learning models trained on partial data generalize to accurately identify critical links across entire networks or cities?

These questions structure the methodological trajectory of the thesis—from conceptual framing and data abstraction, to feature extraction, stress testing, and predictive modeling. Together, they support the overarching aim of enabling smarter, scalable, and transferable criticality assessment for resilient urban transportation planning.

## 1.4 Objectives and Methodological Approach

Answering the research questions outlined above requires more than a single algorithm or analytical tool. It calls for a methodological pipeline—one that spans from conceptual framing to data-driven prediction, grounded in real-world conditions and adaptable to diverse urban contexts. This thesis addresses the absence of such a unified, intelligent, and scalable approach for evaluating road link criticality by designing and testing a learning-based framework that integrates insights from graph theory, simulation, and trajectory analysis.

The overarching objective of the research is:

> To design and evaluate a machine learning–driven pipeline for scalable critical link assessment in urban traffic networks, using a hybrid feature set that integrates spatial attributes, simulation-based disruption effects, and trajectory-derived movement patterns.

To fulfill this objective, the thesis adopts a staged and data-informed methodology, structured around three major phases:

- **Conceptual structuring:** Defining criticality in the context of urban disruptions and establishing a modular perspective on post-disruption traffic management, grounded in a structured synthesis of existing indices and tools, and situating criticality assessment as a core task requiring scalable, data-integrated solutions.
- **Behavioral feature construction:** Mining frequent and confident movement patterns from vehicular trajectory data to derive interpretable indices of link usage and sequence-based importance.
- **Predictive modeling:** Training supervised machine learning models on hybrid structural, behavioral, and simulation-derived features to predict road segment criticality, with an emphasis on cross-network scalability and transferability.

This methodology is grounded in an experimental logic. It begins with a structured synthesis of existing indices and modeling tools to guide feature selection and simulation design. Microscopic traffic simulation is then used not only to extract quantitative impact measures, but to generate synthetic criticality labels based on changes in total trip time following link removal. Sequential pattern mining techniques are applied to simulated trajectory data to derive domain-aware behavioral indicators that account for road directionality, connectivity, and repetition. These are combined with graph-based and spatial features to form a hybrid representational space.

The resulting feature set is used to train a range of machine learning models on a fraction of the network and predict criticality for the remaining links. These models are evaluated on two simulated urban networks—Luxembourg and Monaco—to test their scalability, generalizability, and interpretability. Feature importance analysis and model comparisons are employed to assess the individual and combined contributions of structural, dynamic, and behavioral attributes.

Ultimately, the methodological approach does not aim to optimize a single model, but to demonstrate that link criticality can be modeled as a learnable property: one that emerges from structure, behavior, and disruption response, and can be predicted with limited data using transferable machine learning tools.

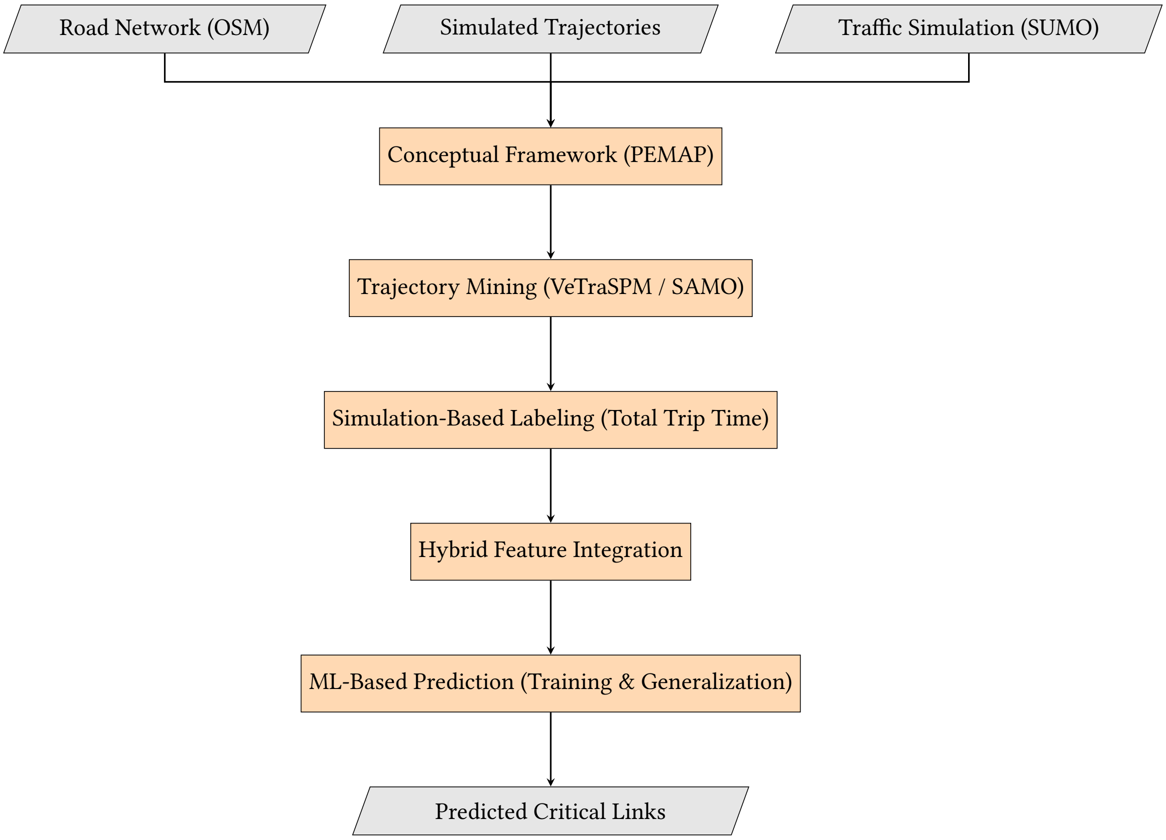



Figure 1.1: Methodological pipeline of the thesis: from data sources to prediction of critical links.

## 1.5 Thesis Outline

The following three chapters are based on three peer-reviewed publications produced throughout the doctoral research project. While originally published in international journals and conferences, the versions presented in this thesis have been revised and adapted to better align with the narrative and methodological coherence of the manuscript. As such, they are not identical to their published counterparts. Each chapter is preceded by a specific preface that introduces the context and situates the contribution within the broader scope of the research.

**Chapter 2** introduces PEMAP (Post-Event MAnagement of transPortation systems), a conceptual framework for post-disruption traffic management. PEMAP defines a four-phase architecture: (1) indices assessment and evaluation, (2) criticality index calculation through simulation-based stress testing, (3) rerouting decision making, and (4) decision application through Vehicular Ad-hoc Networks (VANETs). While the thesis focuses in depth on the second phase, it also addresses Phase 1 by defining criticality in the context of urban disruptions and conducting a structured review of relevant indices and modeling tools. This phase informed the simulation setup and feature design that underpin later chapters. Overall, the framework offers a strategic lens that guided the research structure and highlights the need for integrated, data-driven approaches to disruption-aware urban mobility.

**Chapter 3** presents the development and evaluation of VeTraSPM, a novel domain-specific sequential pattern mining model tailored for vehicular trajectory data. The chapter describes the model's design and its ability to account for the unique characteristics of urban road networks, such as directionality, connectivity, and pattern repetition. From the mined patterns, a series of indices are derived to assess road segment criticality. The chapter also integrates SAMO, a validated extension of VeTraSPM. Through empirical evaluation using simulation data, the model demonstrates its effectiveness in identifying influential road segments and its potential to enhance predictive modeling of link disruptions.

**Chapter 4** introduces SMaL-CLIP, a novel scalable machine learning-based approach to evaluating road criticality. Unlike traditional simulation-based methods, this approach is designed to learn from a small subset of the road network—only 20% of links are required for training—and generalize predictions across the rest of the network. A rich and diverse feature set, comprising over 30 structural, dynamic, and simulation-derived attributes, forms the backbone of the predictive models. Among these features are standard indicators such as travel time, occupancy, edge betweenness centrality, and speed, with SIS-based metrics included as optional enhancements. The framework is rigorously tested on two realistic urban datasets (Luxembourg and Monaco), and demonstrates strong performance across intra- and cross-city scenarios, achieving high accuracy and generalizability. The pipeline incorporates careful feature engineering, model selection (including Random Forest, Gradient Boosting, and MLPs), and hyperparameter optimization, resulting in a system well-suited for large-scale and real-time deployment. This chapter bridges the analytical insights of previous chapters with operational predictive modeling, offering a proof-of-concept for practical deployment.

**Chapter 5** concludes the thesis by synthesizing the findings and reflecting on the contributions made across the three chapters. It revisits the research questions in light of the results, highlights the conceptual shift introduced—modeling criticality as a learnable property—and outlines future research directions. These include operationalizing the full PEMAP framework, applying the approach to real-world trajectory data, and integrating the models into intelligent decision-support systems for disruption-aware traffic management.

Taken together, the chapters illustrate a coherent methodological and conceptual trajectory: from problem structuring and behavioral abstraction to data-driven learning and predictive generalization. While this thesis does not claim to offer a complete solution, it provides a strong foundation for a new class of scalable, explainable, and transferable tools for resilient urban mobility systems.

# Chapter 2

# PEMAP: An Intelligence-based Framework for Post-Event Management of Transportation Systems

# Preface

This chapter is based on a journal article published in 2023 in the journal *Computers and Electrical Engineering* (Bachir et al., 2023). At the outset of the thesis project, the primary objective was to explore how emerging technologies—such as Geographic Information Systems (GIS), microscopic traffic simulation, artificial intelligence, and vehicular communication—could be integrated into a cohesive, intelligent strategy for managing traffic disruptions. The PEMAP framework was developed in response to this ambition and served as the conceptual foundation for the broader research effort.

PEMAP—short for Post-Event MAnagement of transPortation systems—is a modular, intelligence-based framework designed to support urban traffic recovery following disruptive events. It defines a four-phase process: (1) assessment and evaluation of relevant indices, (2) calculation of a criticality index using simulation-informed stress testing, (3) data-driven rerouting decision making, and (4) application of rerouting strategies through Vehicular Ad-hoc Networks (VANETs). Each phase is designed to accommodate domain-specific tools while remaining adaptable to different system contexts and disruption types.

In this thesis, PEMAP also serves as the starting point for a broader reflection on the nature of criticality in urban networks. Phase 1 is used to define criticality not simply as a static topological measure, but as a context-sensitive property that reflects a link's impact on overall network performance under stress. Through a detailed review of indices, metrics, and modeling tools, this phase informed the feature selection and simulation design strategies that underpin the computational models developed in later chapters. In particular, PEMAP highlights the importance of identifying critical links efficiently and at scale—an objective that shaped the design of the machine learning pipeline introduced later.

The motivation for PEMAP stemmed from the observation that existing traffic management tools tend to operate in isolation. Simulation, rerouting, and control mechanisms are often developed as standalone systems, with limited capacity for integration. PEMAP addresses this gap by proposing a structured workflow that links analysis, prediction, and response within a unified framework. It aims to support both policy-level planning and operational traffic management through an intelligent, data-informed pipeline.

While this thesis does not implement the full PEMAP cycle, it uses the framework as a strategic foundation. Particular attention is given to the second phase—criticality index computation—which emerged early on as both a technically complex and underexplored challenge. This phase became the methodological core of the thesis and led to the development of the VeTraSPM pattern mining model and the SMaL-CLIP machine learning pipeline. In this way, PEMAP provided not just a theoretical contribution, but a guiding structure for shaping the sequence and integration of the thesis's key contributions.

This preface situates Chapter 2 within the broader narrative of the thesis. It introduces the vision that connects all subsequent chapters and clarifies how PEMAP functions as both a conceptual anchor and an operational springboard for the research that follows.

## 2.1 Introduction

Communities rely on five essential services and functions to thrive: transport services, telecommunications, power, water services, and community organizations. However, transportation services are the foundation of essential infrastructures since they grant access to other services . The organization and operation of societal and infrastructure systems are significantly impacted by transportation. A transportation network is the structure that allows movement and flow to and from different locations to perform daily life activities. With continued population growth, the need for transportation services is also growing. However, transportation systems are highly susceptible to regular minor as well as unexpected major disruptions. Transportation networks experience a wide variety of relatively transient disruptive occurrences every day, including partial flooding, poor visibility, weather-related traction difficulties, and road degradation . Additionally, these networks may experience unforeseen events like floods, earthquakes, bridge failures, and even malicious acts. These events leave serious impacts on the transportation network and the whole community as a result.

### 2.1.1 Problem Statement

Different events affect the network on different scales depending on the type of the event, its timing, and its location. For example, a 2-car accident on a minor road may partially block this road but otherwise will not really affect the functionality of the whole network. However, a multiple-car accident on a critical link (i.e bridge) at peak hour will certainly gravely affect the transportation network. Furthermore, critical links that lead to emergency spots (hospitals, fire stations, police stations, etc) or community spots (nurseries, schools, community halls, etc) if blocked, cause further panic in the community. Accordingly, critical links are links of utmost importance of whose blockage or certain level of disturbance gravely affects the whole network or a considerable part of it. Disruption of such links causes major stress on the whole network whether by blocking completely the movement from one location to another or by causing major panic.

Data from prior traffic disturbance incidents has demonstrated that transportation authorities recovery and intervention strategies lacked predictive consequence insights ahead (Redzuan et al., 2022). For example, the I-35W bridge collapse over the Mississippi River in Minneapolis on August $1^{st}$, 2007, caused a sudden cut off of the daily commute for around 140,000 trips made seriously hindering the network flow pattern . Life losses and injuries were disastrous, and the monetary, time, and business losses were massive as well. Furthermore, the resulting panic-induced-changes in patterns and ignorant-rerouting caused even more damage by generating more congestion and consequent losses. In such events, the first reaction of people is to just use the closest exit to get away from the disruption, so they head for the other major road that leads to their destination. However, with the blockage of one major path, the path left will become so congested it will be blocked as well disconnecting the network even more. This is called, in the literature, as “cascading effect" (Redzuan et al., 2022). Therefore, it is essential to apply a proper post-event traffic management system that insures the network resilience, particularly in the event of difficulties like high traffic volumes and significant natural disasters .

The framework PEMAP is our proposed solution for intelligent management of traffic post-disruptive-events. This solution is based on extracting important information from the urban road network (URN) and then using that knowledge to take intelligent decisions later at times of events to restore the resilience of the network.

### 2.1.2 Resilience Analysis in Transportation Systems

The study of proper swift recovery from disruptive events falls under what is called, in the literature, as “resilience analysis". Resilience is defined differently in the literature depending on the object of interest and the studied use case. In transportation studies, resilience is commonly defined as the ability of the network to swiftly and efficiently absorb the effects of disruptive events and restore functionality.Most work goals while performing resilience analysis are: impact reduction and swift recovery. However, despite defining resilience in a similar way in Urban Road Network (URN) analysis, different works evaluate and study resilience differently. The performance measurements, language, methodology, and even underlying analytical assumptions utilized in network disturbance studies may differ wildly based on the application, problem area, and the specific research objectives, according to existing research.

URN Resilience evaluation differs from one use case to another as it depends on the specific aspect of resilience considered in the work. This variation hinders the ability to perform a proper comparison between different approaches. Despite that, multiple works in the literature like in (S. Ahmed & Dey, 2020)have attempted to perform a state of art study to cover the different approaches taken and parameters used. In (S. Ahmed & Dey, 2020), the authors identify and classify the different resilience indices used in the literature to evaluate transportation system resilience. They classified them according to each research’s objective, type of transportation network (application domain), type of disruptive event, as well as the methodology used.

In our work, we explored the different concepts of URN resilience analysis and the different indices, methods, and evaluation parameters proposed in the literature as well as the different actions considered after acquiring this knowledge base. As a result, we were able to properly formulate our objective and propose an intelligent framework to redirect traffic post disturbances and improve URN resilience.

### 2.1.3 Contribution

In this work, we focus on post-event traffic management with the following main goal: after the occurrence of an event that affects the traffic, we want to redirect traffic in a way that avoids the blockage of what we evaluated as “critical" links. We define “criticality" of roads as a parameter that measures how crucial the link is to the functionality and efficiency of the whole network. A road with high criticality should be dealt with care so as to avoid it getting blocked. For that reason, our evaluation methods focus on the efficiency of our method in avoiding blockage and congestion on critical roads. Granted, directing traffic to mainly avoid the blockage of critical links might cause the total trip times to increase. It will, however, ensure the connectivity and robustness of the network so that no part of the network is cutout in time of emergency. Hence, the sides of resilience that we are focusing on in our study are mainly robustness and connectivity.

The novelty of our work is integrating the different fields of geographic information systems (GIS), graph theory, traffic simulations, data mining and artificial intelligence, as well as vehicular ad-hoc networks. We propose a framework to improve transportation systems by intelligently redirecting traffic post events to restore functionality. In this framework, we perform analysis of the network to come up with an index that signifies the criticality of different links in a URN. This index is then used to make informed rerouting decisions to avoid further bottle-neck scenarios as well as facilitate the flow of traffic post events. In our study, we are targeting link criticality because we believe that this value is crucial to take the proper decisions to divert traffic from bottle-necking important roads that will lead to possible disconnection of the network.

We propose an intelligence-based framework for post-event management of transportation systems (PEMAP) as an end-to-end solution for road network analysis, decision making, and rerouting post-disturbances. The framework is depicted in Figure 2.1 and is composed of four phases:

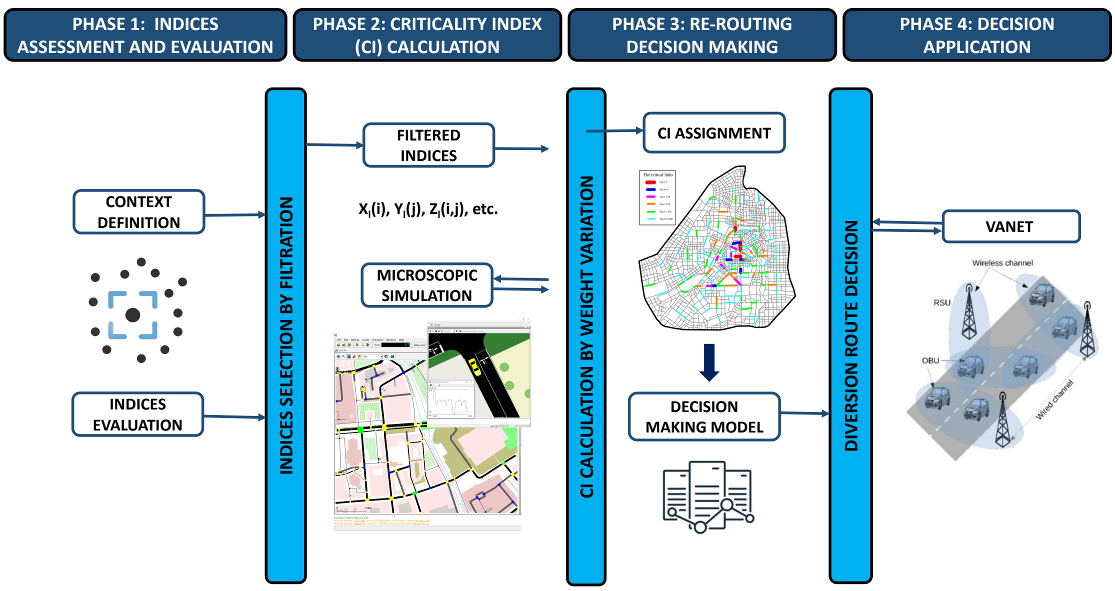



Figure 2.1: PEMAP architecture.

- *Indices Assessment and Evaluation*: in this phase, we study the different indices, evaluation parameters, and methods proposed in the literature, and we perform a comparative analysis to choose the ones that most fit our specific use case and application context to assess the links of the network.

- *Criticality Index (CI) Calculation*: to calculate our criticality index, we combine the different indices with proper weights using microscopic simulation. Then, in order to get the right weights for this index, we study the correlation between the CI for each road and the negative effect of removing this road using stress testing by changing the weights each time.

- *Re-routing Decision Making*: after obtaining the criticality index for each road, we propose a data mining-based model to redirect traffic after disruptions by diverting it from blocking critical roads.

- *Decision Application*: using VANETs, we simulate the application of the proposed model to improve the functionality restoration time of urban networks post-events.

The remainder of this paper is organized as follows: Section 2 presents the different phases of PEMAP framework with an overview of the different works in the literature within each phase. Section 3 presents an evaluation of our proposed framework with respect to other work in the literature as well as the experiments done to validate the work. Section 4 concludes the paper and gives directions for future work.

## 2.2 Our Framework: Post-Event MAnagement of transPortation systems (PEMAP)

Traffic redirection decisions have to take into consideration the structural as well as the dynamic states of the different links in an urban network. Roads, bridges, and tunnels, as well as other road network-related facilities, are crucial elements of the infrastructure that enable the local community business activities . Disturbances on these links are contagious-causing further network flow impediment across the network. These disruptions have grave effects on the functionality of the whole network. Hence, taking into consideration this criticality of roads, intelligent and informed decisions can be made in order to avoid further disturbance of the network. For that reason, we propose Post-Event MAnagement of transPortation systems (PEMAP) framework which is composed of the four previously mentioned phases and described in this section.

### 2.2.1 Indices Assessment and Evaluation

In the first phase of our work we study the measurement indices proposed in the literature that evaluate links based on their criticality or importance. Different approaches and views are implemented to evaluate this index. The goal from this phase is to make use of the previously proposed approaches and views. Deep research and analysis of the different indices have to be performed to select the indices for our specific application context. To achieve this, we have to go through two steps. First, we have to provide an exact definition for criticality which is done through specifying our application context. Second, based on that, we can filter and evaluate the indices proposed in the literature to select the ones that fit our line of work.

### Context Definition

Since there is no one general definition that was used in the literature for criticality, there is a large number of works that were done proposing criticality indices. For which reason we found it mandatory to have this step for context definition to specify the exact context of the criticality index to be calculated in order to define the scope of the work to be done.

In our work, the context and scope of the criticality index is as follows: it aims to highlight **critical** roads which are the most important and whose disruption greatly affects the whole network. For example, roads like bridges connecting two parts of the city, major roads or highways, popular roads in city center and so on. This criticality index does not aim on the other hand to highlight the vulnerability or susceptibility of roads and only their importance.

During our research, we found criticality to be even considered synonymous in some works with “vulnerability", and both were used interchangeably widening even more our pool of research. However, in our work, we make a clear distinction between criticality and vulnerability analysis as we are focusing only on the former.

Most works in the literature studying the vulnerability of roads focus on the *inherent* characteristics, which is mainly the *susceptibility* of this road to disruptive events. In the comprehensive review of resilience modeling concepts (S. Ahmed & Dey, 2020), the authors define it as the “susceptibility of critical components". Their definition and the works they mentioned combines the two concepts of risk (vulnerability) study and criticality (importance) study under one title, e.g. vulnerability. In the work of (Jenelius et al., 2006), the authors highlight this “dual characteristic" and argue that it should be decomposed into two concepts; *weakness* and *importance*.

In Table 2.1, the difference between criticality analysis and vulnerability is highlighted by differentiating the focus, application field, and evaluation of each of them. Both studies are crucial however depending on the context and focus of the work, one concept or the other is studied as it is more relevant.

Table 2.1: Difference between criticality and vulnerability.

| **Index** | **Focus on** | **Application Field** | **Evaluation** |
|---|---|---|---|
| Vulnerability | Risk | Inherent | Weakness Study |
| Criticality | Importance | Post-event | Link Disruption Study |

In our work, our focus is specifically the study of criticality as in the *importance* of the link with respect to the performance and operability of the network post disruptive events. From this importance perspective, a link is considered more critical the graver the consequences of its disruption are.

### Indices in Literature

Different indices/measures were used in the literature. The most common indices mentioned in literature that we have come across are summarized in Table 2.2. Most of the works in the literature focus on betweenness centrality (BC) index (Akbarzadeh et al., 2019; El Rashidy & Grant-Muller, 2014; Feng et al., 2022; Gauthier et al., 2018).Additionally, works like (Akbarzadeh et al., 2019; Gauthier et al., 2018)studied BC index in its unweighted and weighted forms. In (Akbarzadeh et al., 2019), the authors explore six different types of weights: (1) unweighted, (2) traffic flow, (3) link length, (4) reciprocal capacity, (5) congestion, and (6) travel-time. Similarly, the work (Gauthier et al., 2018) explore four different types of weights for BC: (1) unweighted, (2) travel-time weighted, (3) unweighted on entry/exit nodes only, and (4) travel-time weighted from entry to exit nodes only. The work emphasizes the necessity of combining standard static topological analysis with methods for dynamic stress-testing that take demand into account so they consider demand as well to calculate their importance metric. Other works combined BC with other non-graphical indices. The work (Feng et al., 2022) combined betweenness centrality index with link length, clustering coefficient, degree, and road network connectivity indices. In (Li et al., 2020) the authors studied weighted BC index with flow index. The work (El Rashidy & Grant-Muller, 2014) studied in addition to BC and flow, length, link capacity, and congestion as well.

Table 2.2: Indices in Literature.

| **Index/Measure** | **References** |
|---|---|
| Flow | (El Rashidy & Grant-Muller, 2014; Jenelius, 2010; Jenelius et al., 2006; Li et al., 2020; Scott et al., 2006) |
| Exposure | (Jenelius et al., 2006) |
| Length | (El Rashidy & Grant-Muller, 2014; Feng et al., 2022) |
| Importance | (Gauthier et al., 2018; Jenelius et al., 2006) |
| Road network connectivity | (Feng et al., 2022; Scott et al., 2006) |
| Betweenness Centrality (BC) | (Akbarzadeh et al., 2019; El Rashidy & Grant-Muller, 2014; Feng et al., 2022; Gauthier et al., 2018) |
| Clustering coefficient | (Feng et al., 2022) |
| NRI | (Li et al., 2020; Scott et al., 2006) |
| Degree | (Feng et al., 2022) |
| Congestion | (El Rashidy & Grant-Muller, 2014) |
| Link capacity | (El Rashidy & Grant-Muller, 2014; Scott et al., 2006) |
| Weighted BC | (Akbarzadeh et al., 2019; Gauthier et al., 2018; Li et al., 2020) |

Conversely, other works didn't consider BC at all (Jenelius, 2010; Jenelius et al., 2006; Li et al., 2020; Scott et al., 2006; Sullivan et al., 2010). Works like (Jenelius, 2010)consider only flow index while (Jenelius et al., 2006) considers demand as well.

### 2.2.2 Criticality Index (CI) Calculation

Several criticality indices have been proposed in the literature and are presented in Table 2.3. Despite some of these works having a different naming for criticality (i.e. vulnerability), their definitions conform with what we previously defined in our context definition as criticality. Two perspectives stood out for us during our research. The first perspective considered the impact of the link disruption in terms of the travel cost (El Rashidy & Grant-Muller, 2014; Jenelius et al., 2006; Scott et al., 2006). The second perspective evaluated criticality of a link depending on the analysis of diversion routes (Jenelius, 2010; Redzuan et al., 2022).

The authors in (Scott et al., 2006) believed that a fundamental change in network design is crucial, so they presented a system-wide approach and defined a new measure, network robustness index (NRI), for criticality analysis which represents the change in *simulated* travel-time cost resulting from the closure of a link. In the work of (Jenelius et al., 2006), four metrics were derived based on the change in the *generalised* travel cost. In (El Rashidy & Grant-Muller, 2014), however, the authors studied two *aggregated* indices by integrating various vulnerability indices, physical-based or operational-based, and weighting them using stress tests. On the other hand, the authors of (Jenelius, 2010) considered that a non-critical link in normal conditions may become very critical when neighboring links are disrupted and flow is redirected to it. For that reason, they introduced flow and impact based measures to study the importance of a link as an alternative route. Similarly, in (Redzuan et al., 2022), supporting vulnerability was introduced which represents the criticality of a link as one that allows reconnecting the network after the disruption of a major link. They used this measure in their calculation of segment vulnerability index which is the criticality of the link based on the availability of diversion routes. Their main concern was to avoid “cascading effect" post disturbances and to reach equilibrium flow in such situations.

Indeed, both perspectives are interesting to us because we are targeting critical links as critical in their ownselves in normal conditions as well as post disturbances as important diversion routes. In our work, we explore, analyze, and evaluate the indices used in literature in order to filter the most prominent indices and combine them into on criticality index using microscopic simulation..

Table 2.3: Criticality Indices in Literature.

| Ref. | Criticality Index | Measures | Context | Data |
|---|---|---|---|---|
| (Scott et al., 2006) | Network Robustness Index (NRI) | Traffic flow, Traffic capacity, Network connectivity | Metric that evaluates difference in travel-time cost caused by rerouting traffic in the system post a link's disturbance | Three hypothetical road networks, each with different levels of connectivity |
| (Jenelius et al., 2006) | Demand weighted importance, Unsatisfied demand importance, Demand-weighted exposure, Unsatisfied demand exposure | Travel cost, Travel demand | Operational measures to study link importance and site exposure derived based on the increase in generalised travel cost when links are closed | Network and travel data of northern Sweden from the SAMPERS model (2001) |
| (Jenelius, 2010) | Flow-based redundancy importance, Impact-based redundancy importance | Net amount of rerouted flow, Total delay incurred when the link itself is closed | Metrics to study importance of road links as backup alternatives | Swedish national travel demand model system SAMPERS (2001) |
| (Redzuan et al., 2022) | Segment vulnerability index, Supporting vulnerability | Shortest path distance, Diversion path distance | Indices for measuring network vulnerability when facing disruptions based on the availability of diversion routes | Malaysian Peninsular from OSM data (static) |
| (El Rashidy & Grant-Muller, 2014) | Physical-based aggregated vulnerability index, Operational-based aggregated vulnerability index | Link capacity, Traffic flow, Link length, Link free flow, Traffic congestion density | Aggregated indices combining different vulnerability attributes, with weights assigned through stress testing | Synthetic road transport network of Delft city |

### Filtered Indices

Figure 2.2 visualizes the distribution of the different indices mentioned in the literature reviewed which we went over in the previous phase. The indices filtered are the indices chosen by us depending on the context explained and the criticality index definition. For our work, we choose to study the most prominent indices in the literature which are highlighted in the distribution pie chart:

- **Betweenness Centrality (BC) Index:** measures the importance of a specific link in facilitating traffic flow between pairs of nodes in a transportation network. Thus, it can identify links that act as critical connectors between different regions or nodes. While the traditional edge betweenness centrality considers unweighted networks, there is also the option of calculating a weighted betweenness centrality, which takes into account the weights or costs associated with the links. Indeed, the betweenness centrality of a link can be calculated using graph theory algorithms.Accordingly, the BC formula can be expressed as follows:

  $$\text{Betweenness Centrality} = \sum_{s \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.1}$$

  where $\sigma_{st}$ is the total number of shortest paths between all pairs of two distinct nodes $s$ and $t$ in the network, and $\sigma_{st}(v)$ is the number of those shortest paths that pass through the link $v$. The sum is taken over all pairs of distinct nodes $s$ and $t$.

  On the other hand, the formula of the weighted BC may differ depending on the specific weighting scheme used for the links. Mostly, the weighted BC involves summing the weights of the edges along the shortest paths passing through the link of interest. Thus, the calculation will depend on the weights assigned to the links and the method used for incorporating them into the betweenness centrality.

  Indeed, the calculation of BC, whether weighted or unweighted, requires graph theory algorithms that compute the shortest paths between all pairs of nodes in the network. Once the shortest paths are obtained, the number of shortest paths that pass through the specific link is considered. This measure provides insights into the importance of the link in terms of its influence on traffic flow and network connectivity.

- **Flow Index:** it measures the relationship between traffic volume and link capacity thus, providing insights into congestion or link utilization. Flow index is calculated as follows:

  $$\text{Flow Index} = \left(\frac{\text{Traffic Volume}}{\text{Link Capacity}}\right) \times 100 \tag{2.2}$$

  where the traffic volume represents the current traffic volume of a link and the link capacity denotes the maximum traffic volume that the link can handle under ideal conditions. Subsequently, the traffic volume can be obtained based on the traffic counts,

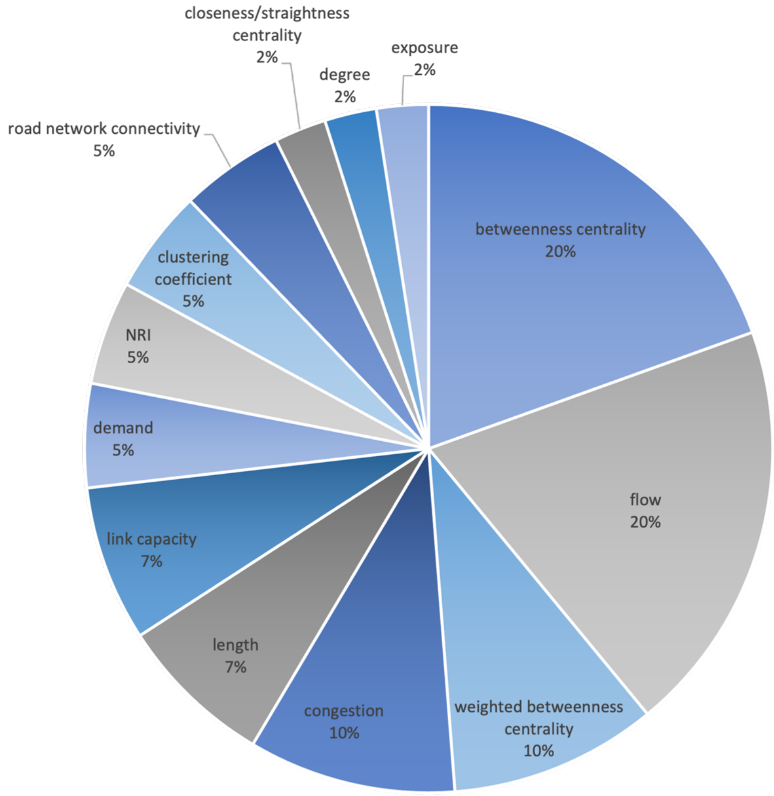



Figure 2.2: Indices Distribution in Literature.

the historical data and the simulation results. Whilst, the link capacity is estimated through field observations, traffic simulation models, and analytical formulas.

- **Congestion Index:** it is a situation where traffic demand exceeds the available capacity on a link. This results in reducing vehicle speeds, increasing travel times and degrading network performance. Congestion assessment is typically based on a set of predefined thresholds or criteria related to the transportation network. For instance, if the flow index exceeds a certain threshold (e.g., $80\%$), the link is considered congested. As a result, the formula for congestion itself varies depending on the specific threshold or criteria set for the network.

- **Link Capacity Index:** it represents the maximum traffic volume that a link can handle under ideal conditions, without significant congestion or delays. Basically, link capacity index is determined through field observations, traffic simulation models, or analytical formulas that consider various factors such as lane width, speed limit, and roadway characteristics. Thus, the link capacity formula is calculated depending on the specific models or approaches used for the estimation.

- **Demand Index:** it refers to the amount of traffic or the number of vehicles that seek to travel through a link or the transportation network within a given time period. It is typically obtained from traffic data sources, such as traffic counts or historical data, and expressed as the number of vehicles per unit of time (e.g., vehicles per hour). Indeed, the demand index varies based on factors such as time of day, day of the week, or specific events. Historical data, traffic counts, travel surveys, or travel demand models can be used to acquire the necessary information for demand calculation.

- **Length Index:** it measures the relative length of a link compared to other links in the transportation network. Length index provides insights into the spatial characteristics of the network and the connectivity between nodes. Mathematically, the length index is calculated by dividing the length of a specific link by the average length of all links in the network. The formula for the length index is shown as follows:

$$\text{Length Index} = \frac{\text{Length of the Link}}{\text{Average Length of all Links}} \tag{2.3}$$

where the link length represents its physical distance or geographical span, and the average length of all links provides a reference value for comparison. More specifically, the link length is mostly obtained from geographic information systems (GIS) or through manual measurements on maps. Whilst, the average length of all links in the network can be calculated by summing up the lengths of all links and dividing it by the total number of links.

The above link indices provide valuable insights into the performance, congestion, importance, and spatial characteristics of individual links in transportation networks. Their calculations involve various data sources, such as historical data, traffic counts, travel surveys, simulation models, and geographic information systems (GIS).

### CI Calculation

Taking these filtered indices, we propose combining them into one Criticality Index (CI). Subsequently, CI is calculated as a function of the different filtered indices where each index has a specific weight.

Mathematically, let's consider $l$ as a specific link of a URN $U$. Let's also consider that we were able to filter $n$ indices in the previous phase: $X_1$, $X_2$, ..., $X_n$. Each of these indices is calculated as a function of some number of characteristics (i.e. $i$, $j$, ..., $z$) of $l$ with respect to the whole network $U$. Accordingly, the criticality index of $l$ with respect to $U$ ($CI_{l/U}$) is calculated as follows:

$$CI_{l/U}(i, j, ..., z) = \alpha \times X_{1l/U}(i, j) + \beta \times X_{2l/U}(j, ..., z) + ... + \theta \times X_{nl/U}(i, j, z) \tag{2.4}$$

where $\alpha$, $\beta$, and $\theta$ are the respective weights of each filtered index.

In order to determine and calibrate these weights, we propose to use the concept of stress testing and evaluation used in (El Rashidy & Grant-Muller, 2014; Li et al., 2020). For this purpose, we remove (block) links of the network, and we study the negative effect of this removal. If the link is critical, the effect should be graver. Hence, the weights for the indices have to be changed and calibrated to reach the best correlation between the resulting CI and the negative effect of removal for a specific link. The flowchart in Figure 2.3 shows how this process is done.

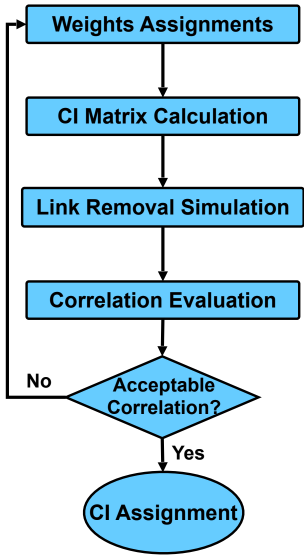



Figure 2.3: CI Weights Calibration.

### Microscopic Simulation for CI Weights Calibration

Two main approaches were taken in the literature for the evaluation of a calculated criticality index. The first approach is usually demand-insensitive and strictly topological. In this approach criticality is evaluated using a number of graph-theory measures.

For example, in (Feng et al., 2022) the authors used shortest path length as an index for evaluating the criticality of roads after link removal. They considered the most significant links as those whose removal causes the greatest rise in the shortest path between two specified node locations. The authors in (Akbarzadeh et al., 2019) quantified the amount of change in the performance of a network using the size of the giant component: number of nodes in the largest set of connected nodes of the network etc.

On the other hand, the second approach involves using a micro-simulator to insert disturbances into the network and measure how much it can adapt to them. This approach models variations in demand while evaluating traffic network performance. In transportation evaluation, microsimulation is a prevalent and cutting-edge technology that offers a practical and immersive approach. By allowing for observation of near-realistic scenarios even after modifications, it has gained wide adoption in literature (Codecá & Härri, 2018; Codecá et al., 2017; El Rashidy & Grant-Muller, 2014; Elsafdi, 2020; Khan et al., n.d.; Lee et al., 2022; Li et al., 2020; Scott et al., 2006; Tian et al., 2021).

We opted to use microsimulation in this work, as it captures various aspects that the former approach misses, such as congestion, speed fluctuations, accidents, and post-roadblock recovery. These aspects are essential to our context and contribute to the realism of the application.

Different simulation tools and approaches were taken in the literature summarized in Table 4.1. In the previously mentioned work (Scott et al., 2006), the authors used TransCAD to study the temporal effect of a link removal. In the work of (El Rashidy & Grant-Muller, 2014), the authors proposed combining multiple vulnerability attributes into a single index considering both physical and operational characteristics. To do this, they used fuzzy logic and exhaustive search using the micro-simulation tool OmniTrans to determine the weight of each fuzzified metric. Another work, (Li et al., 2020), proposed a new approach that takes into consideration traffic flow betweenness index (TFBI) to evaluate criticality of links. To reduce the computation time compared to full-scan methods, they proposed a two step approach where they evaluate first the traffic flow betweenness and rerouted travel demand metrics and then calculate the criticality index by calibrating the weight of the two metrics using MATLAB R2014a software for simulation.

Table 2.4: Microscopic simulation tools in literature.

| **Ref.** | **Micro-Simulation Tool** | **Application** | **Licenses** |
|---|---|---|---|
| (Elsafdi, 2020; Tian et al., 2021) | *PTV Vissim* | Multi-modal | Commercial |
| (Lee et al., 2022) | *FLO-2D* | Field-specific (floods) | Commercial |
| (Scott et al., 2006) | *TransCAD* | Travel demand forecasting | Commercial |
| (Li et al., 2020) | *MATLAB R2014a* | Multi-modal | Commercial |
| (El Rashidy & Grant-Muller, 2014) | *OmniTrans* | Multi-modal and multi-temporal | Commercial |
| (Codecá & Härri, 2018; Codecá et al., 2017; Khan et al., n.d.) | *SUMO* | Multi-modal and multi-temporal | Open-source |

An interesting concern that is usually raised in these simulation approaches is addressed in the work of (Sullivan et al., 2010), which is how the disruption of roads is simulated. The traditional approach used in most works is the complete removal of the link. However, they propose another link disruption modeling approach where they use a high percentage capacity reduction of the link instead of complete removal i.e. $99\%$ capacity reduction.

In our work, we propose using SUMO simulator for micro-simulation as it is the most flexible open source tool and it has a good support with VANET frameworks (i.e Veins), which will be used in our last application phase.

### Evaluation Measures

The evaluation of the negative effect post-link-disruption in microscopic simulations has been studied differently in the literature depending on the use case. As shown in Table 2.5, the most used evaluation parameter is the difference in time/cost between normal condition and after link removal (El Rashidy & Grant-Muller, 2014; Jenelius, 2010; Li et al., 2020; Scott et al., 2006). In the work of (Redzuan et al., 2022), the authors focused additionally on the connectivity reliability.

Table 2.5: Evaluation Measures in Literature.

| **Ref.** | **Evaluation Measure** |
|---|---|
| (El Rashidy & Grant-Muller, 2014; Jenelius, 2010; Li et al., 2020; Scott et al., 2006) | Time/cost difference |
| (Redzuan et al., 2022) | Connectivity reliability |
| (Elsafdi, 2020) | Volume/capacity ratio and average (volume, speed, and delay time) traveling across alternate bridges |
| (Lee et al., 2022) | Length of overlapping sections with critical routes |
| (Tian et al., 2021) | Length of failure road link and total parking delay |

In the work of (Elsafdi, 2020), however, the authors used volume/capacity ratio and average (volume, speed, and delay time) traveling across alternate bridges for evaluation. On the other hand, in (Lee et al., 2022), the length of the sections overlapping with the routes providing critical services was the evaluation parameter considered. In the work of (Tian et al., 2021), the authors evaluated the results of their approach in terms of decrease in the length of failure road link and the total parking delay.

As for evaluation parameters, we will be mainly considering time cost, but we will explore other use-case-related parameters that will better reflect our use case.

### 2.2.3 Re-routing Decision Making

The most common routing protocol used is the shortest (fastest) path provided by typical vehicle navigation systems like Google maps. In a normal use case, when an event occurs that causes people to reroute, they depend on the typical vehicle navigation systems i.e. Google Maps to take another route. However, these systems provide all users with the same routes without considering the whole network flow. Such solutions suggest routes depending only on the fastest route they can come up with for this specific user at that specific instant. This causes all the users to be redirected to the same alternative route (fastest) creating a bottle neck there as well. Hence, rerouting systems have to be more predictive of the state of the network flow and be at least one step ahead of it.

#### Importance of Proactive Rerouting

A better solution than providing the shortest path to every individual vehicle on its own, would be to consider the whole network flow of cars; their origins, their destinations, their criticality factor, etc. Now, both of these solutions have their advantages and disadvantages. The first kind of solutions are easy to implement, but, as mentioned before, when every car is provided the same “fastest route", that route will no longer be the fastest, and, post-disrupting-events, all the main roads will be blocked from the sudden increase in flow. That’s why they’ve been called “selfish" in the literature. Also, these solutions are reactive as they consider only the conditions of the roads at that moment, so they fail to respond to the different events that may occur. On the other hand, the other kind of routing, called “altruistic" in literature, provides a proactive global solution that considers future moves of vehicles. However, ironically, this solution has a fairness issue where some cars will get faster routes comparably. Also, these solutions are harder to apply because they need real-time information regarding all car origins and destination as well as the conditions of the roads which present much higher complexity. An optimal solution for this is a system that balances between the benefits of the selfish and the altruistic approaches.

#### Rerouting Protocols in Literature

We did not come upon any other work in the literature that makes use of the criticality index specifically for rerouting decisions post-disruptive-events. However, many works studied rerouting traffic to avoid congestion by proposing rerouting algorithms that studied each car’s choices and provided the best route considering the whole network. Some of these interesting works are summarized in Table 2.6.

A new two-step rerouting system (NRR) is introduced in the work of (S. Wang et al., 2016) that proposes using heuristic rerouting decision making based on a cost function that considers the car destination as well as local traffic conditions. In this work, the authors are motivated to provide a solution that helps drivers to take the best route to avoid unexpected congestion by balancing between the altruistic and selfish solutions using a two-step approach. Further more, in the work of (Khan et al., n.d.), a new rerouting strategy is introduced (EBkSP) that focuses on balancing the loads between the different routes by rerouting traffic to the paths with lowest “popularity". On the other hand, the work of (Zhou, 2018) investigates using neural networks for decision making to make locally optimal choices. Contrary to basic heuristics, neural networks can consider variables such as the length of a road section, the centrality of an edge, and the speed restriction.

#### Criticality Index in Rerouting Decision Making

We believe that a rerouting protocol that is makes use of the calculated CI, based on a CI weighted links map, will be able to make more intuitive and intelligent decisions. Since the criticality index combines static (betweenness centrality, length, capacity) and dynamic (flow, congestion, demand) indices that offer insight that is crucial to make informed decisions. With a rerouting protocol that recognizes criticality of links, it will be able to take

Table 2.6: Rerouting Protocols in Literature.

| Ref. | Objective | Focus | Algorithm |
|---|---|---|---|
| (S. Wang et al., 2016) | Aid drivers in making the most appropriate next road choice to avoid unexpected congestion | Car's destination and local traffic conditions | Next Road Rerouting (NRR) |
| (Khan et al., n.d.) | Traffic re-routing to avoid congestion | Popularity of roads based on future vehicle positions | Multi-path load balancing considering future vehicle positions (EBkSP) |
| (Zhou, 2018) | Investigating how neural networks can yield a better optimal path than simple shortest path algorithms (proof of concept) | Structural and graphical characteristics of roads | Neural Networks |

this criticality to produce rerouting decisions that focus on maintaining the availability of the most critical links which will in turn maintain the connectivity and reliability of the network. For this reason, in our work, we study the integration of the criticality index into redirection decision making. Since no other work is done specifically to do this, we explore and look into the interesting rerouting protocols previously mentioned modifying the cost function or scope to consider mainly the CI of links.

### 2.2.4 Decision Application

Traffic management systems proposed in the literature have different approaches for application of proposed strategies. Traffic light control systems and automobile navigation systems are the two most popular methods for handling congestion used in the literature. However, both solutions are unable to effectively handle en-route occurrences (S. Wang et al., 2016). In order to apply the rerouting decisions, we propose using Vehicular Ad-Hoc Networks (VANETs) to communicate disturbances to authorities and transfer the decisions to the cars to redirect them to the proper paths. VANETs are networks composed of mobile devices, car on-board units (OBUs), and stationary devices, road-side units (RSUs) and trusted authorities (TAs) i.e. Traffic Operation Centers (TOCs). These networks have two main kinds of communication: vehicle-to-vehicle communication (V2V) between cars and vehicle-to-infrastructure (V2I) communication between cars and RSUs or TAs.

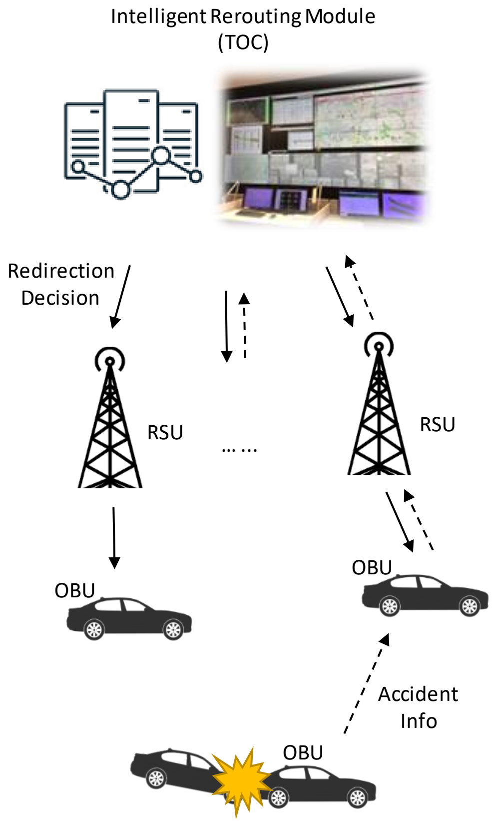



Figure 2.4: VANET Communication for Event & Decision Relay.

In our work, we make use of both kinds of communication to inform authorities of unexpected events as soon as possible as well as to request and relay redirection routes from TOC to vehicles as shown in Figure 2.4. When an accident or event occurs, the information is communicated between cars until it reaches the closest OBU. The OBU will then communicate this message to the TOC which replies with the proper diversion route decisions made using the Intelligent Rerouting Module. The RSU then relays these decisions to the affected vehicles.

Many works in the literature used VANETs to improve transportation systems. In the work of (Jayapal, 2022), the authors propose a VANET-based distributed, cooperative system for traffic congestion sensing and information dissemination without requiring human involvement, to reroute incoming traffic and lessen congestion. Similarly, in the work of (Toulni et al., 2014), the authors use VANETs for congestion avoidance. Using VANET communication, they work on determining the best route to cut down on travel time and fuel usage by examining the valuable and reliable real-time traffic data. In (M. Wang et al., 2015), to efficiently enable real-time information exchange among automobiles, road-side units (RSUs), and a vehicle-traffic server, the authors provided a real-time route planning algorithm that aims to enhance a road network overall spatial efficiency and lower the overall cost of a car trip in order to prevent traffic jams with the use of both vehicular ad hoc networks (VANETs) and cellular systems of the public transportation system. Additionally, in (Santamaria et al., 2018), the authors focus on the development of a novel strategy for mobile node journeys in vehicle environments that may gather data for warning of dangerous or emergency situations by utilizing on-board sensors. On the other hand, the authors of (Knorr et al., 2012) introduce a plan to use vehicle-to-vehicle communication to lessen traffic congestion using continuously broadcast beacons to monitor flow of traffic and alert other drivers of potential traffic jams. Vehicles that get this notice are expected to maintain a wider distance from the vehicle in front of them. Their approach however targeted highway use cases only.

In this work, we will use VANETs to communicate disruptive events to trusted authorities as well as relay the redirection decisions received from the proposed model in real-time. This approach ensures rapid notification and recovery post-events.

## 2.3 Evaluation

For the evaluation of this framework, each phase has to be implemented in turn and the iterative results will be compared to the state of art approaches. That is because the novelty of our proposed framework is that no other work in the literature proposes a similar end-to-end solution that combines the study of criticality index using the different fields mentioned and the post-event decision making and application using Data Mining and VANET.

To evaluate our proposed framework several tools and technologies are needed. In order to calculate the criticality index, an “accident" or road blockage have to be simulated to analyze the effect of this blockage. Instead of testing this by real life example, events can be simulated. For a long time now, micro-simulation tools have been used to test multiple transportation approaches in the literature. Furthermore, for decision making an intelligent model has to be calculated and implemented. We chose to do this using Python and C++ languages. For the last phase of VANET simulation and testing, a combination of a micro-simulation tool and VANET network simulation tool have to be used. For that purpose we chose to use Veins framework that combines the two most popular open-source tools, SUMO simulator and OMNet++. SUMO simulator is also the chosen micro-simulation tool to be used in the CI calculation phase.

### 2.3.1 Proposed Framework vs State-of-Art: Similarities and Dissimilarities

Despite our work being the only work in the literature that introduces such an end-to-end solution concept, we have come upon some interesting works in the literature that took a similar approach in one or more aspect of our work. We have summarized the similarities and dissimilarities in Table 2.7.

In the work of (Elsafdi, 2020), they make a similar assumption that having better knowledge of the traffic conditions of roadways helps absorb sudden traffic surges. In their work, they recognize “crucial links" and focus on increasing the resiliency of the network in case of these links failure, specifically bridges. However, the actions they take are to improve the “inherent" resilience which falls on the side of preventive measures not post events.

Table 2.7: Comparison of frameworks proposed in literature with similar concept to PEMAP.

| **Ref.** | **Similarities to our work** | **Dissimilarities from our work** |
|---|---|---|
| (Elsafdi, 2020) | Studied importance of links and taking actions accordingly | The actions they took were to increase "inherent" resilience (preventive) |
| (Lee et al., 2022) | Proposed an approach to direct waste transportation focusing on clearing "critical routes" after floods | Dealt with waste transportation only, and their definition of critical routes was based only on those connecting critical services |
| (Henry et al., 2019) | Integrated analyzing dynamic and topological characteristics of links and used stress testing | The approach is purely analytical; no post-event solution studied |

On the other hand, in (Lee et al., 2022), the authors propose an approach to reroute waste transportation post-floods to mainly decrease overlapping of those routes with what they identified as critical routes. However, their evaluation of critical routes mainly focused on routes that provide critical services not the structural and dynamic characteristics of roads. In (Henry et al., 2019), the authors identify the difference between analyzing dynamic and topological (static) characteristics and they propose an approach to integrate both metrics by weighing the different indices and assessing and highlighting the effect of area-wide disruptions using stress testing. Their work though is purely analytical and does not propose a post-event management solution to recover from disruptions.

### 2.3.2 Proposed Framework vs State-of-the-Art: A Comparative Study

In order to emphasize the novelty previously mentioned, we summarize the works proposed in the literature in the following Table 2.8. All of the surveyed works cover one or two phases where as our proposed framework is the only end-to-end solution that covers the four different phases.

While the process of disruption management has attracted considerable research attention, much of it has been directed at the pre-disruption stage. Most of the works in the literature studying criticality analysis do not use this value for post-disruptive events management. Rather, these works focus on fixing inherent and structural states of the links to improve resilience. PEMAP on the other hand focuses on the post-disruption stage along with its management. The unpredictability of disruption magnitude and nature suggests that the post-disruption management process may be as important, if not more so, than pre-determined pre-disruption strategies. An effective post-disruption management process would directly affect actual ability to recover from sudden and serious disruptions.

Moreover, in our work, we make use of the study and the analysis of the characteristics and graph-based metrics of links in urban road networks in order to make rerouting decisions. On the other hand, the approaches proposed in the literature focus on real-time and do not consider the structural and graphical aspects of links to handle re-directions accordingly.

Table 2.8: Phases Covered in Literature vs in PEMAP

| **Ref.** | ***Phase 1*** | ***Phase 2*** | ***Phase 3*** | ***Phase 4*** |
|---|---|---|---|---|
| (Redzuan et al., 2022) | X | X | | |
| (Jenelius et al., 2006) | | X | | |
| (El Rashidy & Grant-Muller, 2014) | X | X | | |
| (Li et al., 2020) | X | X | | |
| (Jenelius, 2010) | X | X | | |
| (Scott et al., 2006) | | X | | |
| (Elsafdi, 2020) | X | | | |
| (Tian et al., 2021) | X | X | | |
| (Lee et al., 2022) | X | | X | |
| (Khan et al., n.d.) | | | X | |
| (S. Wang et al., 2016) | | | X | |
| (Zhou, 2018) | | | X | |
| (Jayapal, 2022) | | | | X |
| (Toulni et al., 2014) | | | X | X |
| (M. Wang et al., 2015) | | | X | X |
| (Santamaria et al., 2018) | | | | X |
| (Knorr et al., 2012) | | | | X |
| (Henry et al., 2019) | X | X | | |
| **PEMAP*** | X | X | X | X |

*Our proposed approach.

### 2.3.3 Experiments

To evaluate our proposed framework several tools and technologies are needed. In order to calculate the criticality index, an “accident" or road blockage have to be simulated to analyze the effect of this blockage. Instead of testing this by real life example, events can be simulated. For a long time now, micro-simulation tools have been used to test multiple transportation approaches in the literature. Furthermore, for decision making an intelligent model has to be formulated and implemented. We chose to do this using Python and C++ languages. For the last phase of VANET simulation and testing, a combination of a micro-simulation tool and VANET network simulation tool have to be used. For that purpose we chose to use Veins framework that combines the two most popular open-source tools, SUMO simulator and OMNet++. SUMO simulator is also the chosen micro-simulation tool to be used in the CI calculation phase.

### Data and Simulation Scenario

Using SUMO simulator, we had two options: either to create a scenario from scratch, or to use a well created and tested scenario provided by the SUMO community. In our work, we decided to do the latter because it allows our work and results to be compared to those of others' using the same scenario. Further more, creating a SUMO scenario from scratch that is as realistic as possible is time consuming and out of the scope of this work. Instead we decided to make use of the existing realistic scenario LuST (Codecá et al., 2017) of the city of Luxembourg. A scenario of Monaco, MoST (Codecá & Härri, 2018), is also used in our work to validate the ability of our framework to be applied to different cities of different sizes and transport nature. The variation and numbers of both scenarios are shown in Table 4.2.

Table 2.9: SUMO Simulation Scenario Numbers

| Ref. | Name | City | Area | Nodes | Edges | Trips |
|---|---|---|---|---|---|---|
| (Codecá et al., 2017) | LuSTScenario | Luxembourg | 155.95 km$^2$ | 2,247 | 5,779 | 215,526 |
| (Codecá & Härri, 2018) | MoSTScenario | Monaco | 22 km$^2$ | 2,004 | 4,404 | 7,990 |

As can be seen, despite the number of nodes and edges being close, the other numbers in the scenarios vary greatly. That is because the scenario of Luxembourg simulates traffic across a much bigger city. Further more, it simulates traffic for a whole day with the realistic traffic variation shown in Figure 2.5. The figure shows the variation of the traffic at different times of the day and highlights how the scenario realistically depicts traffic at peak (rush) hours.

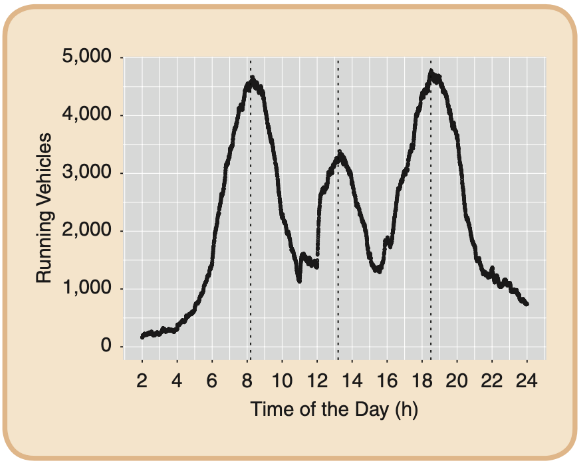



Figure 2.5: Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand (Codecá et al., 2017).

### Results

Using Python sumolib and networkx libraries we were able to parse and extract the graph of each city. The graph generated is a multi-bi-directional graph with the edges representing the roads of the city and the nodes representing intersections between roads. In our work, we considered lanes with the same starting and ending points as one edge to avoid redundancy.

Taking both scenarios, we extracted the graphical representation of the city roads as mentioned, as well as the traffic flow and trips which we analyzed and used to apply our approach. We calculate for each edge (road) the different indices mentioned previously. Figure 2.6 shows the criticality of links based on unweighted betweenness centrality (BC). Figure 2.7 shows a clear difference in criticality evaluation after adding length index.

Figure 2.6: Unweighted BC Weighted Map of Luxembourg.

In Figure 2.8, BC is used along with length and speed which are used to calculate the cost of each link. On the other hand, Figure 2.9 shows the map of Luxembourgh weighted by flow extracted from the simulation results.

After that we apply our algorithm to calculate the criticality indices and the results are shown in Figure 2.10 and Figure 2.11 of the cities of Luxembourg and Monaco respectively.

The variation between the results in Figures 2.6, 2.7, 2.8, and 2.9 in comparison to those in Figure 2.10 show how our work provides an edge and more realistic results by better highlighting critical roads at the center of the city and in main roads.

## 2.4 Conclusion

In this paper, we have proposed a framework called PEMAP that allows to redirect traffic post disruptive events based the intelligence extracted from the criticality analysis of the different links in the urban road network. The proposed framework is made of the four

Figure 2.7: BC, Length Weighted Map of Luxembourg.

phases: indices assessment and evaluation, criticality index calculation, re-routing decision making, and decision application. In the first two phases of criticality analysis, we use the fields of geographic information systems, graph theory, microscopic simulation, and social networks. In the last two phases, we use data mining for decision making and VANET for real-time application.

We also emphasized in this work the novelty of this work which is the combination of different fields into one end-to-end system that performs analysis and decision making for post-event traffic management. In order to provide such framework, we explored the literature of each field individually and its contributions to intelligent transportation systems. We, then, properly defined our problem statement and purpose as well as distinctly emphasized our understanding of criticality analysis in order to perform an analysis of the existing works and approaches. We also performed comparative evaluation with existing works and provided experiment results to further validate the proposed framework.

In the future work, we intend to continue studying the applying the different phases of PEMAP framework. This work is the first in a series to realize PEMAP. We'll take each phase in our next works, study the state of art, and implement our approach.

Figure 2.8: BC, length and speed (cost) Weighted Map of Luxembourg.

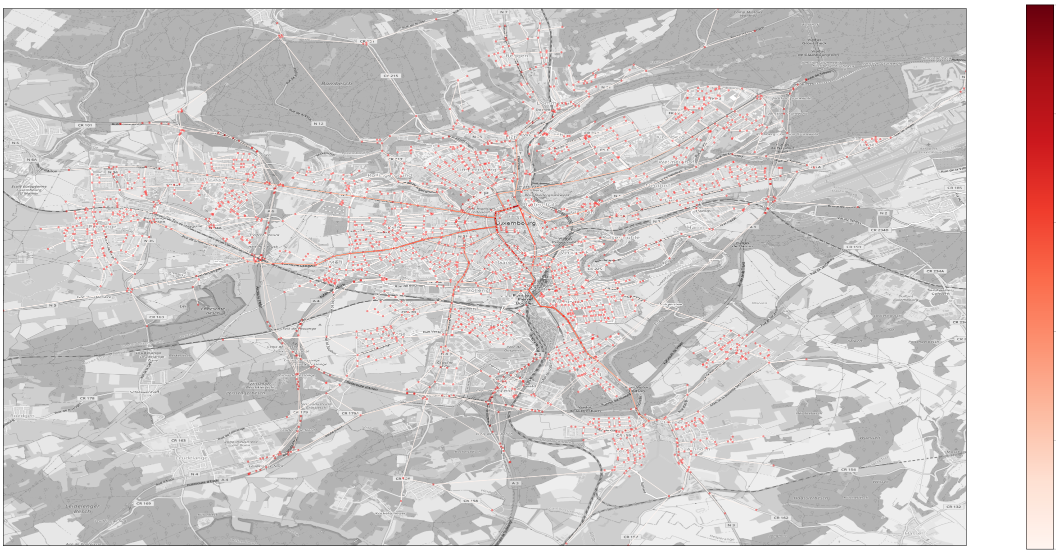


Figure 2.9: BC, Flow Weighted Map of Luxembourg.

Figure 2.10: Criticality Index Weighted Map of Luxembourg.

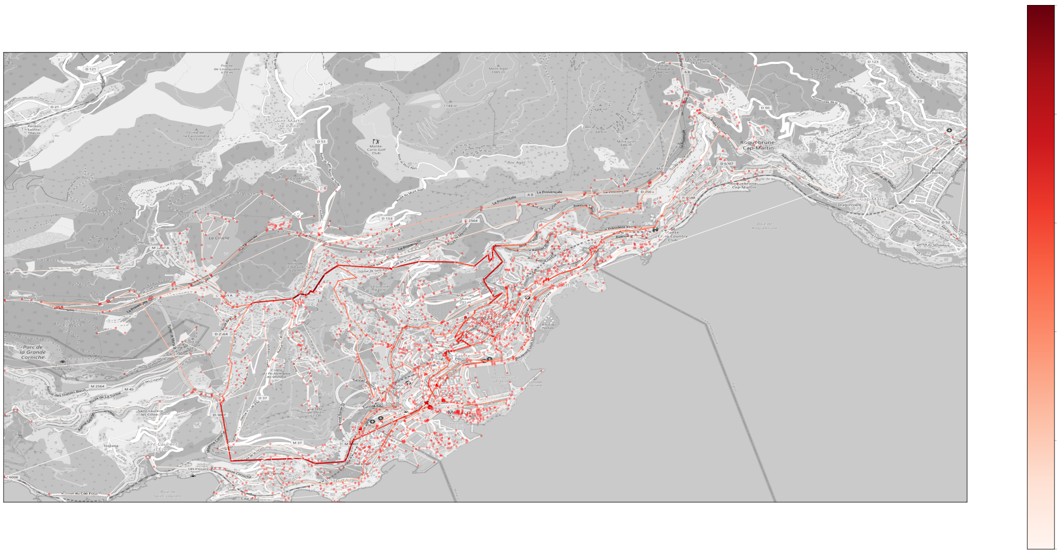


Figure 2.11: Criticality Index Weighted Map of Monaco.

Chapter 3

# VeTraSPM: Novel Vehicle Trajectory Data Sequential Pattern Mining Algorithm for Link Criticality Analysis

# Preface

This chapter is based on an article published in 2025 in the journal *Vehicular Communications* (Bachir et al., 2025b). Following the conceptualization of the PEMAP framework presented in the previous chapter, the research focus shifted toward addressing one of its core phases—critical link identification. A central challenge emerged in this context: the need to assess the relative importance of road segments based on real-world vehicular behavior, rather than purely topological or static indicators. This recognition led to the development of VeTraSPM, the core contribution presented in this chapter.

VeTraSPM, short for Vehicle Trajectory Data Sequential Pattern Mining, was designed to extract frequent and confident movement patterns from vehicle trajectory data while respecting the inherent characteristics of urban traffic. Unlike conventional sequential pattern mining algorithms, VeTraSPM explicitly accounts for directionality in road segments (e.g., one-way restrictions), network connectivity (via junctions), and sequence repetition (where road segments may appear multiple times in a journey). These traffic-specific constraints are often ignored in traditional pattern mining methods, limiting their applicability to real-world urban networks.

The chapter begins by formalizing the mining problem in the context of transportation and introduces key technical innovations of the VeTraSPM algorithm. It employs a vertical projection approach for efficient memory use, combined with positioning tables that track movement patterns within trajectory sequences. A novel pattern extension strategy, PT-Ext, is proposed to recursively grow and prune movement patterns while preserving their spatial and directional integrity. From these patterns, a set of novel indices are derived namely: Frequency-based Movement Score (FqMS)–later renamed to Support Frequency Index (SFI), Confidence-based Movement Score (CMS)–later renamed to Confidence Impact Score (CIS), and Sequential Impact Score (SIS). These metrics quantify the significance of road links based on their role in movement behavior and are designed to be both interpretable and adaptable to different analysis contexts.

Beyond algorithmic efficiency, the chapter demonstrates VeTraSPM's practical value through comparative experiments with baseline algorithms such as Apriori. The results show significant improvements in computational performance and scalability, particularly under low-support thresholds typical of urban movement data. A case study using trajectory datasets from Luxembourg and Monaco further validates the model's ability to reveal critical road segments that are not necessarily identifiable through static or topological analysis alone.

The methodology and results presented in this chapter represent a pivotal step in the thesis. VeTraSPM introduces the behavioral layer of the PEMAP framework, complementing the structural and simulated perspectives with insights grounded in real-world trajectories. In doing so, it also lays the foundation for the predictive modeling work that follows in Chapter 4. The SIS and other indices developed here are later tested as features in machine learning models to evaluate their added value in predicting criticality.

Although a subsequent study—presented at an international conference in 2024 (Bachir et al., 2024)—extends and further validates the SIS-based metrics, the chapter focuses solely on the original development and assessment of VeTraSPM. The additional experiments and refinements are included at the end of this chapter under a dedicated "Subsequent Work" section for completeness and transparency.

By situating VeTraSPM within the broader methodological arc of the thesis, this preface emphasizes its role as both a standalone contribution and a crucial building block in the thesis's interdisciplinary approach to post-event traffic resilience.

## 3.1 Introduction

Transportation networks are essential lifelines that enable the movement of people, goods, and services, contributing to economic growth, social connectivity, and overall well-being. However, these networks often face disruptions due to accidents, natural disasters, construction, or other unforeseen events, requiring transportation authorities to manage traffic efficiently and mitigate the impact of such disruptions. Identifying critical links—key segments of the network where disruptions cause significant traffic bottlenecks—has become vital to supporting resilient transportation systems. Traditional approaches rely on static attributes and fail to account for the dynamic nature of vehicle trajectories, limiting their ability to assess link criticality accurately.

Sequential pattern mining is an effective technique to analyze temporal data and discover meaningful patterns from sequences. However, when applied to vehicle trajectory data, existing algorithms encounter several limitations. Firstly, they do not accommodate directional constraints in one-way roads, where, for example, "AB" might be a valid sequence, but "BA" is not. Secondly, they fail to capture the connectivity constraints among links, meaning only certain links can follow each other based on the road network's structure (e.g., linked by a junction). Thirdly, the potential repetition of links (e.g., a vehicle revisiting the same location) is often ignored, leading to inaccurate pattern detection in complex urban traffic networks. These limitations restrict the utility of existing algorithms in real-world scenarios.

To address these challenges, the introduced VeTraSPM (Vehicle Trajectory Data Sequential Pattern Mining), a novel algorithm tailored to the characteristics of vehicle trajectory data. VeTraSPM accurately discovers frequent movement patterns and confident rules, considering directionality, connectivity, and repetition. The algorithm employs vertical projection techniques, allowing for efficient memory and space management, enabling it to handle large datasets. Additionally, VeTraSPM incorporates partitioning and parallelization strategies to further enhance scalability, making it practical for real-world large-scale urban traffic systems. These optimizations ensure the algorithm's efficiency even with extensive datasets and multiple cores, improving both time and space performance during pattern mining.

In this work, three novel metrics—Frequent Movement Pattern Score (FqMS), Confident Movement Pattern Score (CMS), and Sequential Impact Score Index (SIS)—are proposed to assess link criticality based on the consistent appearance of links across frequent and confident movement patterns at various levels. The insights from these metrics are visualized to demonstrate how they can guide traffic managers in identifying critical links and making proactive decisions.

The experimentation in this study is conducted on the real-world transportation network of Luxembourg, incorporating traffic data from the SUMO LuST microscopic simulation scenario (Codecá et al., 2017), the performance of VeTraSPM is compared against the baseline algorithms, demonstrating that VeTraSPM offers superior efficiency, especially when processing large datasets. LuST scenario is based on authentic traffic patterns incorporating genuine demographic data and activity demand through the ACTIVITYGEN tool.

The evaluation dataset that was used for this scenario includes over six million Floating Car Data (FCD) samples collected in Luxembourg City as mentioned by the authors. With detailed metrics covering road networks, intersections, and various mobility aspects, using this scenario's data ensures a realistic assessment of the proposed methodology.

The organization of the paper is as follows: Section 2 provides an overview of related work in the field of vehicular data mining and analysis. Section 3 outlines the proposed methodology, explaining the new propositions, the application of association rules algorithm, and the determination of link criticality. Section 4 explains the implementation of the approach. Section 5 presents the experimentation and evaluation of the case study, comparing the performance of the proposed approach with existing methods. Finally, Section 6 concludes the paper, summarizing the contributions and highlighting avenues for future research.

This work offers new tools for urban planners and policymakers, facilitating more effective traffic management strategies in dynamic urban settings.

## 3.2 Related Works

This section provides an overview of the existing sequential data mining algorithms commonly used in trajectory data analysis. It also reviews various research works that have attempted to analyze critical links in transportation systems using different methodologies.

### 3.2.1 Existing Sequential Data Mining Algorithms

Frequent pattern mining plays a key role in sequential pattern discovery, with algorithms generally categorized into Apriori-based and FP-growth-based approaches. The Apriori algorithm (Agrawal & Srikant, 2000) is widely used for association rule mining by identifying frequent itemsets and deriving rules based on support and confidence measures. However, its need for multiple database scans makes it computationally expensive and inefficient for large or complex datasets (Han et al., 2000, 2004; Thabtah, 2007).

PrefixSpan (Pei et al., 2004) addresses these limitations by avoiding candidate generation, focusing instead on frequent subsequences, which reduces computational overhead. An enhanced version, PrefixSpan-x (Xue et al., 2016), further improves memory efficiency by pruning unnecessary patterns. Top-k sequence mining algorithms (Fournier Viger et al., 2013; Garofalakis et al., 2002; Kemmar et al., 2017) discover the most relevant patterns by applying global constraints such as quantity or item relationships.

Constraint-based algorithms refine pattern discovery by introducing additional rules. These include regular expression constraints, weight-based constraints, and length constraints (Oza & Kawade, 2015), which narrow the search space to improve relevance. Closed sequence mining (Tzvetkov et al., 2003; J. Wang & Han, 2004) eliminates redundant sequences, keeping only the most significant patterns. Incremental mining algorithms (Masseglia et al., 2003) update discovered patterns dynamically, avoiding full re-computation when new data becomes available.

Vertical data mining algorithms, such as SPAM (Ayres et al., 2002), SPADE (Zaki, 2001), CM-SPADE (Fournier Viger, 2014), ClaSP (Gomariz et al., 2013), and VMSP (Fournier Viger et al., 2014), offer further efficiency by converting datasets into vertical representations. These methods are particularly useful for large datasets, with SPAM leveraging depth-first search through item and sequence extensions to improve speed and memory usage. SPADE and CM-SPADE build on this by partitioning data into equivalence classes, further optimizing complex sequence mining.

Despite their efficiency, these vertical mining algorithms are not well-suited for vehicle trajectory data, which features ordered sequences, directional constraints, and repetition. Vehicle trajectories are continuous and may contain loops or revisits, patterns that these algorithms struggle to capture. This limitation reduces their effectiveness for critical link analysis in transportation systems, where such nuances are essential.

Table 3.1 presents an overview of key sequential pattern mining algorithms and their characteristics. Although these algorithms are effective for various domains, they are limited in handling the specific properties of vehicle trajectory data, such as directionality, repetition, and sequence order.

Vehicle trajectory data requires specialized techniques that can accommodate its continuous nature and repetitive patterns. Conventional algorithms struggle with these features, reducing their applicability to transportation systems. Thus, despite the advancements in vertical mining algorithms, such as SPAM, SPADE, CM-SPADE, ClaSP, and VMSP, further improvements are needed to address the structural complexity of vehicle trajectories while maintaining computational efficiency.

Table 3.1: Summary of Sequential Pattern Mining Algorithms and Techniques

| Algorithm | Category | Key Features and Limitations |
|---|---|---|
| Apriori | Apriori-based | Identifies frequent itemsets but requires multiple scans, leading to high time complexity. |
| PrefixSpan | Prefix-based | Reduces search space but requires significant memory for long sequences. |
| PrefixSpan-x | Enhanced Prefix-based | Optimizes memory by pruning unnecessary patterns. |
| Top-k Sequence Mining | Constraint-based | Discovers top-k patterns with global constraints like quantity and item relations. |
| SPADE | Vertical projection | Efficient for large datasets but struggles with ordered, repetitive sequences. |
| CM-SPADE | Vertical mining | Partitions data into equivalence classes; memory-intensive for complex sequences. |
| ClaSP | Class sequential mining | Identifies class-specific patterns but is limited by trajectory constraints. |
| SPAM | Vertical bitmap-based | Uses depth-first search for efficient mining; limited for handling repetitive patterns in trajectories. |
| VMSP | Vertical mining | Efficient for long sequences but lacks support for ordered and repetitive patterns. |
| Regular Expression Constraints | Constraint-based | Uses regex, weight, and length constraints to enhance pattern relevance. |
| Weighted Sequence Mining | Weighted patterns | Mines patterns with weights, improving relevance in specific contexts. |
| Closed Sequence Mining | Closed pattern discovery | Prunes redundant sequences, retaining only the most significant patterns. |
| Incremental Mining | Incremental learning | Dynamically updates patterns but may struggle with large datasets. |

### 3.2.2 Sequential Data Mining on Trajectory Data

Several studies have applied sequential pattern mining to vehicular data. Yu, 2019 applied Apriori to discover frequent movement paths from taxi trajectories but encountered high time complexity due to repeated database scans. Ibrahim and Shafiq, 2019 combined clustering with the SPADE algorithm to generate insights from large taxi datasets, though their method struggled with ordered and repetitive patterns.

Y. Wang et al., 2023 used Apriori for planning flexible bus services by analyzing multiday path clusters but did not account for order and directionality. Hu et al., 2020 applied frequent pattern mining to assess public transport stability but faced scalability issues. Moreira-Matias et al., 2012 employed PrefixSpan to detect bus bunching, though their method overlooked repetitive sequences.

Table 3.2: Applications of Sequential Pattern Mining in Vehicular Data Analysis

| **Study** | **Algorithm Used** | **Limitations** |
|---|---|---|
| Yu, 2019 | Apriori | High time complexity, multiple scans. |
| Ibrahim and Shafiq, 2019 | SPADE | Struggles with order and repetition. |
| Y. Wang et al., 2023 | Apriori | Ignores road connectivity and directionality. |
| Hu et al., 2020 | FP-growth | Limited scalability and prediction accuracy. |
| Moreira-Matias et al., 2012 | PrefixSpan | High memory usage, overlooks order and repetitions. |

Table 3.2 presents key works in vehicular data analysis, demonstrating the limitations of basic algorithms in handling trajectory-specific challenges.

### 3.2.3 Sequential Data Mining for Critical Link Analysis

Despite its importance, critical link analysis using sequential pattern mining remains underexplored. Hu et al., 2020 focused on public transport stability but did not address critical link identification. H. Zhang and He, 2021 explored trajectory prediction in VANETs without assessing critical links. Qi et al., 2017 developed a timeliness-aware algorithm for delay-tolerant networks but their approach lacks consideration of the order of links, potentially affecting the quality of predictions. Merah et al., 2012, 2013 applied basic sequential mining techniques for real-time vehicle tracking and route discovery but lacked advanced mining methods to improve prediction accuracy.

Table 3.3 highlights the gaps in current research, emphasizing the need for an advanced approach that leverages association rules and sequential data mining techniques to identify critical links within transportation networks.

Table 3.3: Applications of Sequential Mining in Critical Link Analysis

| Study | Application | Focus and Limitations |
|---|---|---|
| Hu et al., 2020 | Public transport stability | No focus on critical link analysis. |
| H. Zhang and He, 2021 | VANET trajectory prediction | Focuses on movement paths, lacks critical link assessment. |
| Qi et al., 2017 | Delay-tolerant networks | Timeliness-aware, lacks consideration of the order of links. |
| Merah et al., 2012, 2013 | Real-time tracking and route discovery | Route prediction, relies on a basic implementation of sequential pattern mining. |

## 3.3 Vehicle Trajectories Sequential Pattern Mining (VeTraSPM)

An interesting approach for the discovery of critical links in transportation networks is the extraction of meaningful patterns from vehicle movement sequential data. The existing algorithms of sequence pattern discovery, like association rules algorithms such as Apriori suffer from their low accuracy when applied on vehicle trajectories data. In order to deal with this issue, this paper presents a new vehicle movement sequential data mining model for sequence pattern mining abbreviated *VeTraSPM* based on the vehicle trajectory data structure.

### 3.3.1 Preliminary Definitions

- *Trajectory Sequence (Tr):* $V = \{V_1, V_2, ..., V_n\}$ represents a group of vehicles that travel for a certain period in a given geographical area. $Tr_i = \{e_x, e_y, ..., e_n\}$ represents the trajectory sequence of vehicle $i$ and $TrDB$ represents the set of sequences of all vehicles.

- *Sequence length:* The length of a sequence is simply the number of all links within this sequence. i.e. consider $Tr_k = \{e_x, e_y, e_z\}$ then $length(Tr_k) = length(\{e_x, e_y, e_z\}) = 3$.

- *Sub-sequence:* $Tr_i = \{e_x, e_y, \ldots, e_n\}$, the trajectory sequence of vehicle $i$, is a sub-sequence of the trajectory sequence of vehicle $j$, $Tr_j = \{\ldots, e_x, e_y, \ldots, e_n, e_m, \ldots\}$, where $length(Tr_i) \leq length(Tr_j)$ and $Tr_j$ contains the whole sequence of $Tr_i$ in the same order i.e. $\{e_x, e_y, e_z\}$ is a sub-sequence $\{e_l, e_x, e_y, e_z, e_m\}$, but not a sub-sequence of $\{e_x, e_l, e_y, e_m, e_z\}$

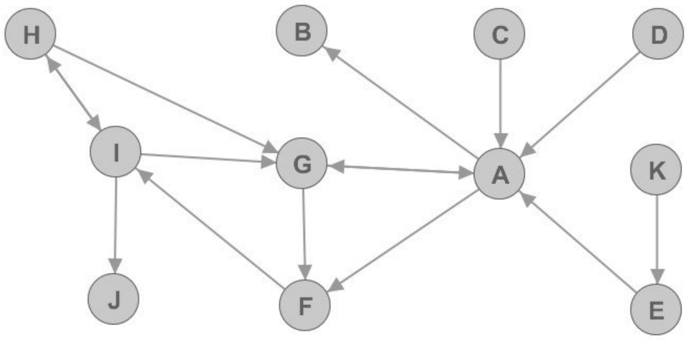



Figure 3.1: Example directed multi-graph

- *Movement pattern (Mp):* A movement pattern $M$ represents a specific pattern (sub-sequence) to be detected in the trajectory data i.e $M = \{e_x, e_y\}$, $M' = \{e_z\}$.
- *Movement pattern order:* The order of a movement pattern is the length of the respective sequence. i.e. $order(M) = length(\{e_x, e_y\}) = 2$, $order(M') = length(\{e_z\}) = 1$. A movement pattern of order 1 is called "unit movement" which is in this case one edge/road/link like $M'$.
- *Movement rule:* A movement rule $R$ is defined as association rule between two movement patterns expressed as $M \rightarrow M'$.
- *Support:* The support of movement pattern $M$ is the number of appearances of this movement pattern as a sub-sequence in all trajectory sequences $S$. The support of the rule $R = M \rightarrow M'$, is the support of movement pattern $MM'$ in the mobile database. i.e. $support(R) = support(MM') = support(\{e_x, e_y, e_z\})$
- *Confidence:* The confidence of movement rule $R$ can be defined as: $confidence(R) = \frac{support(MM')}{support(M)}$.
- *Frequent Movement Pattern Set (FqM):* For a given support threshold *minsup*, a frequent movement pattern is a pattern whose support is not lower than *minsup*. A Frequent Movement Pattern Set denoted as $FqM_k$ is the set of movement patterns of order $k$ whose support is not lower than the *minsup* i.e. consider movement pattern $MM' = \{e_x, e_y, e_z\}$, if $support(\{e_x, e_y, e_z\}) \geq minsup$ then $MM'$ is a frequent movement pattern and $MM' \in FqM_3$.
- *Confident Movement Pattern Set (CM):* For a given confidence threshold *minconf*, a confident movement pattern is a frequent movement pattern whose confidence is greater than or equal to *minconf*. For each $FqM_k$ set of frequent movement patterns of order $k$, $CM_k$ set of confident movement patterns can be extracted i.e. where $confidence(MM') = \frac{support(\{e_x,e_y,e_z\})}{support(\{e_x\})}$, if $confidence(MM') \geq minconf$, then $MM'$ is a confident pattern and $MM' \in CM_3$.

Take the following example based on the graph in Fig. 3.1 and the trajectories database (TrDB) provided in Table 3.4. Let $M$ be the movement pattern defining sequence $\{a, g\}$ and

Table 3.4: Example of vehicle driving trajectories database (TrDB)

| **Sequence** ($i$) | **Trajectory Seq.** ($Tr_i$) | **Sequence** ($i$) | **Trajectory Seq.** ($Tr_i$) |
|---|---|---|---|
| 1 | [f, i, **g, a**] | 8 | [**g, a**, b] |
| 2 | [e, a, b] | 9 | [k, e, **a**, f, i, h, **g**] |
| 3 | [c, **a, g**, f, i, h, **g, a**] | 10 | [c, **a**, f, i, **g**] |
| 4 | [d, a, b] | 11 | [i, **g, a**, b] |
| 5 | [e, **a, g**, f, i, j] | 12 | [f, i, j] |
| 6 | [i, h, g, f] | 13 | [**a, g**, f, i] |
| 7 | [h, i, **g, a**, b] | 14 | [d, a, b] |

$M'$ be the movement pattern defining sequence $\{f\}$. The confidence of trajectory association rules $R = M \rightarrow M'$ is the ratio of the number of trajectory sequences in the trajectory database that contain movement pattern $MM'$, $support(MM')$, to the number of trajectory sequences that contain movement pattern $M$, $support(M) = support(\{a, g\}) = 3$. Consequently, $confidence(R) = support(MM')/support(M) = support(\{a, g, f\})/support(\{f\}) = 3/3 = 1$. A rule of confidence 1 is called a *perfect rule* which signifies that whenever the path $M$ is taken $M'$ is always the next step.

### 3.3.2 Vertical Projection of Trajectory Data

In order to come up with the frequent rules and the rules with best confidence, all the different combinations of items (edges) have to be explored and all the sequences in the database are queried over and over to find the support of each possible movement rule. This process is time consuming and memory intensive which is why efficient implementations have been studied in the literature.

An efficient implementation for sequence association rules generation is by performing vertical projection of the sequences and presenting *Unit Movement Pattern* with one *Sequence-Pattern Identity List* that shows where this item has shown up. Using this approach:

- The database is queried only once, and each unit movement's presence in sequences is projected into one list called *Trajectory Identity List* (TIL).
- To explore the support of combinations, it is enough then to query these lists.

For example, consider the *TrDB* shown in Table 3.4, the resulting *TILs* are shown in Table 3.5.

In order to calculate the support of a movement pattern it is enough to perform the intersection between the different unit movements' *TILs.*

Consider the case of calculating the support of $\{a, g\}$ using the explained algorithm. The support would then be the length of the list resulting from the intersection of $a$'s and

Table 3.5: Example of Trajectory Identity Lists (TILs)

| Unit Movement Pattern | Trajectory Identity List |
|---|---|
| a | [1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14] |
| b | [2, 4, 7, 8, 11, 14] |
| c | [3, 10] |
| d | [4, 14] |
| e | [2, 5, 9] |
| f | [1, 3, 5, 6, 9, 10, 12, 13] |
| g | [1, 3, 5, 6, 7, 8, 9, 10, 11, 13] |

$g$'s TILs. The resulting list would be [1,3,5,7,8,9,10,11,13] which is of length 9. This length should indicate the support of the movement pattern $\{a, g\}$ which is the number of its occurrence in the database. However, in the database, only 3 occurrences can be observed (highlighted in green) of this movement pattern. This error occurred because this approach fails to take into considerations the two following factors:

- the **order** of the items as well as
- the possibility of **recurrence** of the item in the same sequence.

Consequently, while this approach increases efficiency by decreasing the amount of database queries, applying this to vehicle trajectory database while these two critical factors are overlooked makes it produce wrong results.

### 3.3.3 New Definitions

In this study approach, it was necessary to add the following definitions and concepts:

1. Seeing as the sequences are trajectory movements, each edge can only be followed by a specific set of edges called set of outgoing edges and defined as follows:

$$O_x = \{y \in E \mid y \text{ is an outgoing edge from } x\} \tag{3.1}$$

   Here, *E* represents the set of all edges in the graph, $x$ represents an arbitrary edge in the set *E*, and $y$ represents all edges that are outgoing from $x.$ i.e. $O_a = \{b, f, g\}$ according to the graph in Fig. 3.1.

   This property decreases substantially the search space, time, and complexity while generating the frequent sequences; instead of exploring all possible combinations, the set of outgoing edges is used to generate the next valid movement patterns.

2. *Positioning Table* (PT) is proposed in this study in order to project the order as well as the possible recurrence of a unit movement in a trajectory sequence. A PT is a

two-dimensional table generated for a movement pattern that records all the positions where this pattern appears in the trajectory database (TrDB). Each row in this table records the positions of appearance of the movement pattern within a specific trajectory sequence using an *ordered-positioning list* (OPL) where each item in this list is a position of occurrence of this pattern within the specified sequence.

- $OPL(M)_i$ represents the Ordered-Positioning List for movement pattern M in the trajectory sequence $Tr_i$. This list contains all the positions within this sequence where pattern $M$ appears, recorded in a specific order. Mathematically, $OPL(M)_i$ can be represented as follows:

$$OPL(M)_i = \{pos_j \mid pos_j \text{ is the position of } j^{th} \text{ occurrence of pattern M in } Tr_i, \text{ for } 1 \leq j \leq n_i\} \tag{3.2}$$

  In this equation $n_i$ is the total number of occurrences of pattern $M$ in the trajectory sequence $Tr_i$.

- $PT(M)$ represents the Positioning Table for movement pattern M in the trajectory database TrDB. It contains only the Ordered-Positioning Lists ($OPL_i$) for pattern M where M exists in the $i^{th}$ trajectory sequence. Mathematically, $PT(M)$ can be represented as follows:

$$PT(M) = \{OPL(M)_i \mid M \text{ exists in the } i^{th} \text{ trajectory sequence}\} \tag{3.3}$$

To summarize, $OPL(M)_i$ contains all positions of occurrence of pattern $M$ in a sequence $Tr_i$, and $PT(M)$ contains the Ordered-Positioning Lists for pattern $M$ in the entire trajectory database i.e. consider the unit movement patterns $\{a\}$ and $\{g\}$ and the $TrDB$ in Table 3.4, the generated positioning tables $PT(\{a\})$ and $PT(\{g\})$ are shown in Figs. 3.2a and 3.2b respectively.

### 3.3.4 Position Tables Extension Approach (PT-Ext)

To find the position of sequences the positioning tables are extended using the proposed strategy *PT-Ext*. Consider $PT(X)$ and $PT(Y)$ are the positioning tables of the movement patterns $X$ and $Y$ respectively. $Y$ is a unit movement $\{y\}$, of order 1, and let $X$ be of order $k$; $X = \{x_1, ..., x_k\}$. Considering that the following rules apply:

- Both $X$ and $Y$ belong to the frequent movement pattern sets $FqM_k$ and $FqM_1$ respectively.

- $y$ is an outgoing edge for the edge $x_k$ ($y \in O_{x_i}$) where $x_k$ is the $k^{th}$ (last) unit movement in $X$.

Checking that all these rules apply, *PT-Ext* then goes over each row (Sequence id, $i$) in $PT(X)$ and checks for the following:

| **Sequence** ($i$) | $OPL_i$ |
|---|---|
| 1 | (**4**) |
| ~~2~~ | (2) |
| 3 | (2, **8**) |
| 4 | (2) |
| 5 | (2) |
| 7 | (**4**) |
| 8 | (**2**) |
| 9 | (3) |
| 10 | (2) |
| 11 | (**3**) |
| 13 | (1) |
| ~~14~~ | (2) |

(a) Positioning Table of $\{a\}$

| **Sequence** ($i$) | $OPL_i$ |
|---|---|
| 1 | (3) |
| 3 | (**3**, 7) |
| 5 | (**3**) |
| ~~6~~ | (3) |
| 7 | 3) |
| 8 | (1) |
| 9 | (7) |
| 10 | (5) |
| 11 | (2) |
| 13 | (**2**) |

(b) Positioning Table of $\{g\}$

| **Sequence** ($i$) | $OPL_i$ |
|---|---|
| 3 | (**3**) |
| 5 | (**3**) |
| 13 | (**2**) |

(c) Positioning Table of $\{a, \mathbf{g}\}$

| **Sequence** ($i$) | $OPL_i$ |
|---|---|
| 1 | (**4**) |
| 3 | (**8**) |
| 7 | (**4**) |
| 8 | (**2**) |
| 11 | (**3**) |

(d) Positioning Table of $\{g, \mathbf{a}\}$

Figure 3.2: Example joining positioning tables

1. Index of $X$ in $S_i$ is not the last one otherwise it can't be extended.

2. The sequence id, $i$, exists in the tables of both movement patterns $X$ and $Y$.

3. The index of $y$ is directly subsequent to the index of the initial movement pattern $X$ in $S_i$ i.e. index of $X$ in $S_i$ is $d$ and the index of $y$ is $d+1$.

If both rules apply, the index of the subsequent movement pattern is added to the movement pattern positioning list for the new extended sequence, $OPL(X')$, where $X'$ is the extension of $X$ by $y$ aka $X' = \{x_1, ..., x_k, y\}$.

In the tables shown in 3.2a and 3.2b, the initial positioning tables of $\{a\}$ and $\{g\}$ respectively are shown. Extending the positioning table of $\{a\}$ by $\{g\}$ since $g$ is an outgoing edge for $a$, the sequence ids that show up in both tables $PT(\{a\})$ and $PT(\{g\})$ have to be checked (discarding sequences $2, 4$ and $6$). Then, the remaining common sequences' order-positioning lists are checked to see whether the index of the considered outgoing edge $index(\{g\})$ is $index(\{a\})+1$. As shown in the tables, that is the case in the sequences $3, 5$, and $13$. Now in the positioning table of $\{a, g\}$, for these three sequences, a list is added that contains the index of $\{g\}$, the subsequent pattern. The resulting PT of $\{a, g\}$ is shown in Fig. 3.2c.

Extending position tables using *PT-Ext* makes sure that all the errors previously discussed are considered and solved. Furthermore this algorithm is more efficient as it only extends frequent patterns, excluding non-frequent patterns, and explores only possible extensions, those included in the set of outgoing edges of the last unit movement as opposed to exploring all (frequent) edge combinations.

### 3.3.5 VeTraSPM Model

This paper presents a new approach for trajectory sequences pattern mining abbreviated *VeTraSPM*. VeTraSPM employs a path *Positioning Table* abbreviated PT for data storing, mining and pattern expansion. The PT of each single-edge-path are firstly obtained through scanning sequence database once. Then PTs are mined to identify the frequent single items. Based on the new expansion strategy named *PT-Ext*, the frequent patterns are extended and the extended tables are created and updated. The extended PTs then are explored to obtain the frequent extended patterns. Finally, a pruning technique is used for to avoid the generation of unnecessarily large number of candidate patterns.

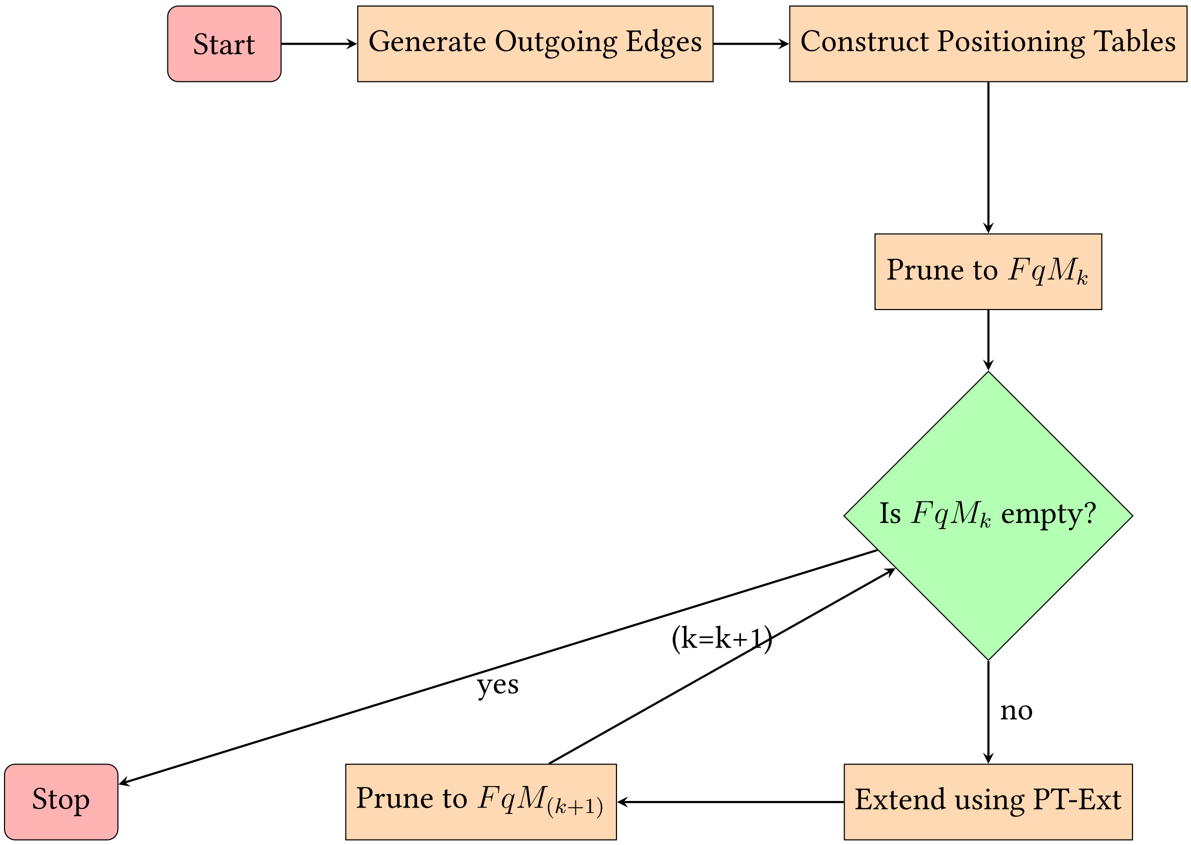



Figure 3.3: VeTraSPM

Figure 3.3 illustrates the sequential flow of operations in the VeTraSPM algorithm, designed to identify frequent movement patterns in vehicle trajectory data. The process consists of several key steps:

1. **Generate Outgoing Edges:** The algorithm begins by analyzing the map to identify all *outgoing edges* for each edge in the network. This step captures the possible connections between road segments based on their physical connectivity, such as junctions or intersections.

2. **Construct Positioning Tables:** Next, the vehicle trajectory database is scanned to *construct positioning tables* for movement patterns of order $k$. These tables map the

locations of movement sequences across the dataset, helping to keep track of where specific road sequences occur.

3. **Prune to Obtain Frequent Patterns ($FqM_k$):** The positioning tables are then pruned to extract the frequent movement patterns $FqM_k$ of order $k$. This step ensures that only movement patterns meeting a minimum frequency threshold are retained.

4. **Check if $FqM_k$ is Empty:** After generating the frequent movement patterns for the current order $k$, the algorithm checks if the set $FqM_k$ is empty.

   - *If $FqM_k$ is not empty*, the algorithm proceeds to extend the patterns to the next order, $k + 1$.
   - *If $FqM_k$ is empty*, the algorithm terminates, as no further patterns can be generated.

5. **Extend Patterns using PT-Ext:** When the frequent patterns $FqM_k$ are non-empty, they are *extended to higher-order patterns* ($k + 1$) using the *PT-Ext joining strategy*. PT-Ext ensures efficient extension by merging compatible patterns and generating new positioning tables for the extended patterns.

6. **Recursive Pruning for Higher-Order Patterns:** The newly generated positioning tables for higher-order patterns are pruned again to obtain the next set of frequent movement patterns, $FqM_{(k+1)}$. This process ensures that only significant patterns are retained at each step.

7. **Termination Condition:** The process repeats recursively, generating and extending frequent patterns until no new patterns can be found, i.e., when the set $FqM_k$ becomes empty. At this point, the algorithm stops.

### 3.3.6 Sequential Impact Scores (SIS): New Criticality Indices based on VeTraSPM

In the context of this study, let $n$ represent the order of the last non-empty set of frequent movement patterns. Consider a specific movement pattern $M$, to gauge the significance of this pattern across various orders of frequent movement patterns, the following novel metric is introduced: $SIS$.

Let $FqMS(M)$ be the frequent movement pattern score of movement pattern $M$ and $CMS(S)$ be the confident pattern score of pattern $M$.

$$FqMS(M) = 1 \cdot x_1 + \left(\frac{1}{i}\right) \cdot x_i + \ldots + \left(\frac{1}{n}\right) \cdot x_n \tag{3.4}$$

$$CMS(M) = 1 \cdot y_1 + \left(\frac{1}{i}\right) \cdot y_i + \ldots + \left(\frac{1}{n}\right) \cdot x_n \tag{3.5}$$

Here's the breakdown of the equations:

- The terms $1 \cdot x_1$ and $1 \cdot y_1$ signify the contribution of occurrences of pattern $M$ in the first-order set of frequent movement patterns ($FqM_1$) and the first-order set of confident movement patterns ($CM_1$) respectively.

- The terms $\left(\frac{1}{i}\right) \cdot x_i$ and $\left(\frac{1}{i}\right) \cdot y_i$ encapsulates the importance of pattern $M$ in the $i^{th}$ order set of frequent movement patterns ($FqM_i$) and confident movement patterns ($CM_i$) respectively. Here, the factor $\frac{1}{i}$ is utilized to weigh patterns by their order inversely, assigning higher importance to lower-order patterns.

- Similarly, the terms $\left(\frac{1}{n}\right) \cdot x_n$ and $\left(\frac{1}{n}\right) \cdot y_n$ take into account the occurrences of pattern $M$ in the last non-empty set of frequent movement patterns ($FqM_n$) and confident movement patterns ($CM_n$), with a weight proportional to $\frac{1}{n}$.

The Sequential Impact Score of movement pattern $M$ is defined by the following equation:

$$SIS(M) = FqMS(M) + CMS(M) \tag{3.6}$$

In essence, the Sequential Impact Score $SIS(M)$ combines the occurrences of pattern $M$ across different orders, where the weight assigned to each order is inversely proportional to its value. This makes sure that the most frequent and confident patterns have been captured and that links were assigned higher importance according to their consistent appearance in the frequent movement patterns and confident movement patterns of higher order. These indices provide a comprehensive assessment of the sustained significance of a movement pattern across diverse levels of sequence complexity.

By calculating the Sequential Impact Scores for each movement pattern, insights can be gained into the patterns that consistently exhibit significance across different orders of frequent movement patterns and confident movement patterns. This information proves invaluable for decision-making and comprehending how the relevance of patterns evolves.

## 3.4 Implementation of VeTraSPM and SIS Calculation

This section describes the implementation of VeTraSPM and $SIS$ calculation, along with optimizations aimed at improving memory efficiency, scalability, and computational performance. Several strategies are employed, including lazy evaluation, partitioning, memory-efficient data structures, early termination, and parallelization, to ensure the algorithm can handle large datasets efficiently. A comparative analysis of the original and optimized versions is provided to demonstrate the benefits.

### 3.4.1 Research Flow

The research follows a structured flow consisting of the following major steps, shown in Figure 3.4.

- Data Preparation: The *DataPreparation* function takes an *xml_file* as input and processes it to extract the necessary data. It first imports the data using *ImportData* and then converts this data into a directed graph using *CreateDirectedGraph*. The function returns the generated graph for further processing.

- Edge & Table Construction: The second function, *EdgeAndTableConstruction*, receives the directed graph as input. This function is responsible for generating outgoing edges from the graph using *GenerateOutgoingEdges*. Additionally, it constructs positioning tables from the outgoing edges using *CreatePositioningTables*. The resulting positioning tables are returned to continue the mining process.

- Frequent Pattern Mining: In the *FrequentPatternMining* function, the positioning tables are used to extract frequent patterns. Initially, the patterns are pruned based on a support threshold using *PatternPruning*. After that, the pruned patterns are extended using *ExtendPatterns*. A loop is executed to iteratively prune and extend the patterns until no new frequent patterns are found. This ensures that only the most relevant patterns are retained. The frequent patterns are then returned for final analysis.

- Final Analysis: *FinalAnalysis* processes the frequent patterns to calculate $SIS$ using the *CalculateSIS* function. Once $SIS$ is calculated, critical links are identified in the network through the *IdentifyCriticalLinks* function.

- Visualization: The final step is carried out where the identified critical links are visualized for interpretation using *VisualizeCriticalLinks*.

### 3.4.2 Generate Outgoing Edges and Positioning Tables Construction

The first step is to read the simulation network as a traffic road network model. In this work, a directed multi-graph is constructed from the city map using the *networkx* library, maintaining the traffic network's connections and circulation patterns. A directed multi-graph supports multi-directional edges and loops, allowing a realistic representation of road networks. Each road is mapped as an edge-weighted by length, cost, and average speed.

Once the graph is built, the outgoing edges for each road link are extracted. This is achieved through Algorithm 1, which iterates over the graph, mapping each edge to its corresponding outgoing edges. A directed multi-graph representation of the traffic network is loaded, and for each edge in the graph, the outgoing edges are identified. The algorithm iterates over all edges and stores their outgoing edges in a dictionary. The outgoing edges are stored as NumPy arrays for more efficient indexing and numerical operations. The time complexity is $O(E)$, where $E$ represents the total number of edges. With partitioning, the workload is distributed into smaller subsets, reducing the complexity to $O(E/P)$, where $P$ is the number of partitions.

For the second step, the vehicle driving trajectories database (TrDB) is first loaded. By processing the data only when necessary, lazy evaluation is employed, thus reducing peak

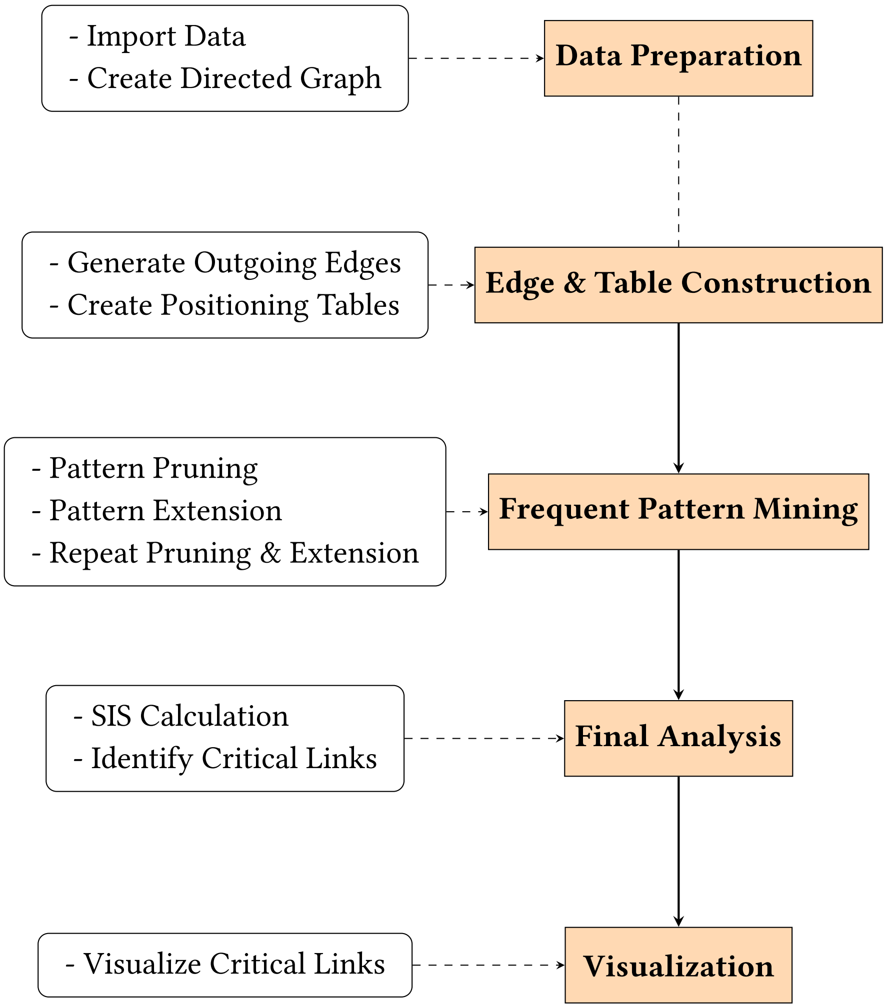



Figure 3.4: Model Flow

memory usage. Algorithm 2 shows how instead of loading the entire XML file and its contents into memory all at once, Vehicle routes can be processed one at a time and yield the result incrementally. This way, the data is processed on demand rather than all at once greatly reducing the space complexity from $O(V \cdot M)$ where $V$ is the number of vehicles and $M$ is the average number of movements per vehicle to $O(1)$. This strategy is especially useful when dealing with large datasets, as it minimizes memory usage while maintaining the same time complexity.

TrDB is then scanned once to construct the initial unit movements positioning tables which is the vertical projection of the database. These tables indicate for each edge in which routes it has appeared and at which position. Algorithm 3 takes the vehicle driving trajectories database (TrDB) named *tr_db* as input and produces the dictionary *edge_pt*. The algorithm begins by initializing an empty dictionary called *edge_pt*. It then lazily iterates through each trajectory in *tr_db*, which is processed one by one to avoid loading the entire dataset into memory. For each trajectory (route), the algorithm iterates through the enumerated list of unit movements (edges). During this iteration, the index of each unit movement within the trajectory is appended to the list associated with the correspond-

**Algorithm 1** Construct Edge Outgoing Dictionary

**Require:** *G* (directed multi-graph representation of the map), *P* (number of partitions)
**Ensure:** *out_edges* (dictionary mapping each edge to its corresponding array of outgoing edges)

Initialize an empty dictionary called *edge_out_dict*
Partition *G* into *P* subgraphs *G1, G2, ..., GP* for parallel processing
**for** each partition *Gp* in parallel **do**
    **for** *edge* in *Gp* **do**
        *out_edges[edge] = np.array(edge.out_edges())* ▷ Store as a NumPy array for efficient access
    **end for**
**end for**
**return** *out_edges*

**Algorithm 2** Parse TrDB from XML File

**Require:** *xml_path* (path to an XML file containing vehicle routes)
**Ensure:** A generator yielding each vehicle's route as a list of movement units

**for** *veh_route* in *xml_path* **do**
    **yield** *veh_route.edges()* ▷ Yield each vehicle route instead of appending
**end for**

ing unit movement and trajectory ID in the dictionary *edge_pt*. The algorithm returns the completed dictionary *edge_pt*, where each unit movement is mapped to its respective positioning table as a Numpy array, containing the trajectory IDs and indices for efficient sequential pattern mining. Employing partitioning and parallelization as well minimizes memory consumption, reducing the time complexity from $O(V \cdot M)$ to $O((V \cdot M)/P)$.

Using this positioning table concept, the space and time complexity are greatly reduced compared to parsing the original table each time. Parsing the original table at each order would be space and computationally expensive as it would be of the time complexity $O(N \cdot V \cdot M)$ where $N$ is the total number of movement patterns at all levels. In contrast, the time complexity of parsing the partitioning table at each order $k$ would be $O(N_{(k-1)} \cdot M_{(k-1)})$ where $N_i$ is the number of movement patterns at order $i$ only and $M_i$ is the average number of occurrences of the movement patterns of order $i$. The space complexity is similarly reduced as only the order-specific movement patterns are considered and their specific indices.

### 3.4.3 Pruning and Extending Positioning Tables

To identify frequent movement patterns, low-frequency patterns are pruned using Algorithm 4. This algorithm calculates the support of each pattern and retains only those that meet the minimum support threshold. The given Algorithm 4 takes the table *init_pt*, table of positioning tables of current order $k$ movement patterns, and a minimum support value

**Algorithm 3** Construct Unit Movement (Edge) Positioning Tables

**Require:** *tr_db* (generator yielding vehicle driving trajectories (TrDB)), *P* (number of partitions)

**Ensure:** *edge_pt* (dictionary mapping each unit movement to a NumPy array for its positioning table)

Initialize an empty dictionary called *edge_pt*
Split *tr_db* into *P* partitions
**for** *partition* in *tr_db* **do** ▷ Process partitions in parallel
    **for** *traj* in *partition* **do**
        **for** *um_index, unit_movement* in *enumerate(traj)* **do**
            **if** *unit_movement* not in *edge_pt* **then**
                *edge_pt[unit_movement] = np.empty((0,), dtype=int)* ▷ Initialize NumPy array
            **end if**
            *edge_pt[unit_movement] = np.append(edge_pt[unit_movement], um_index)*
        **end for**
    **end for**
**end for**
**return** *edge_pt*

*min_sup* as input.

**Algorithm 4** Prune Movement Patterns by Minimum Support

**Require:** *init_pt* (dict), *min_sup*, *P* (number of partitions)
**Ensure:** *frequent_movement_patterns_list* (list)

```
Split init_pt into P partitions
Initialize an empty list called frequent_movement_patterns_list
for partition in init_pt do                          ▷ Process partitions in parallel
    for mv_pattern in partition do
        total_support ← 0
        is_frequent ← False
        for traj_id in init_pt[mv_pattern] do
            total_support ← total_support + len(init_pt[mv_pattern][traj_id])
            if total_support ≥ min_sup then
                Add mv_pattern to frequent_movement_patterns_list
                is_frequent ← True
                break
            end if
        end for
        if is_frequent then
            break
        end if
    end for
end for
return frequent_movement_patterns_list
```

**Algorithm 5** Positioning Table Extension (PT-Ext)

**Require:** *frequent_movement_patterns_list*, *init_pt*, *edge_pt*, *last_edge_table*, *frequent_out_edges*, *P* (number of partitions)

**Ensure:** *pt_extended*

```
Initialize an empty dictionary called pt_extended
Split frequent_movement_patterns_list into P partitions
for partition in frequent_movement_patterns_list do                ▷ Process in parallel
    for movement_pattern in partition do
        mp_pt ← init_pt[movement_pattern]
        last_edge ← movement_pattern[-1]
        for index in mp_pt[trajectory_id] do
            if trajectory_id not in last_edge_table[last_edge] then
                for out_edge in frequent_out_edges[last_edge] do
                    if trajectory_id in edge_pt[out_edge] then
                        for out_index in edge_pt[out_edge][trajectory_id] do
                            if out_index = index + 1 then
                                movement_id ← movement_pattern ⊕ out_edge
                                pt_extended[movement_id][trajectory_id].append(out_index)
                            end if
                        end for
                    end if
                end for
            end if
        end for
    end for
end for
return pt_extended
```

It iterates through each movement pattern in the table, calculating the total support for each by summing the lengths of its list of positions in the trajectories associated with it. Once the total support of a movement pattern exceeds or equals the specified *min_sup*, it is returned in a list called *frequent_movement_patterns_list*. Finally, the algorithm returns the resulting list ($FqM_k$), which contains movement patterns of order $k$ that meet the minimum support threshold. Before proceeding to the next step, the generated list ($FqM_k$) is saved in the database for later use. Also, the corresponding confident movement pattern set ($CM_k$) of the same order is calculated using $PT(FqM_k)$ and $PT(FqM_1)$ and saved. Considering the exhaustive case where each movement pattern appears no more than one time in a trajectory, the time complexity of Algorithm 4 at each level $k$ would be $O(N_k \cdot M_k)$. Using partitioning, this complexity is improved to $O(N_k \cdot M_k/P)$.

The identified frequent patterns are extended using the PT-Ext algorithm (Algorithm 5). This process involves adding valid extensions to existing patterns, based on the outgoing edges of the last road link in each pattern. The algorithm takes as input:

- the resulting list of frequent movement patterns of order $k$, *frequent_movement_patterns_list*,
- the initial positioning table *init_pt* of movement patterns of order $k$; first time around it's the same as *edge_pt* (k=1),
- the edge positioning table *edge_pt* of unit movement patterns of order $1$,
- *last_edge_table*, a table containing for each unit movement pattern (edge) a list indicating the ids of the trajectory sequences it is the last edge in (final destination),
- the dictionary of **frequent** outgoing edges for each unit movement pattern *frequent_out_edges* which resulted from pruning *out_edges* and *edge_pt* by minimum support.

The result is the new "extended" positioning tables of movement patterns of order $k + 1$, *pt_extended*.

The idea in this algorithm is to go over only the already identified frequent movement patterns of order $k$ and extend them with only the identified frequent outgoing edges of their last unit movement pattern. The algorithm initializes an empty dictionary called *pt_extended*. It then starts by iterating over frequent movement patterns. For each movement pattern (*movement_pattern*), it retrieves its positioning table (*mp_pt*). In order to increase efficiency, *last_edge_table* are used to exclude the trajectory ids where the last edge of the considered movement pattern is the last one in. After that, the algorithm goes over the frequent outgoing edges of the last edge (*last_edge*) in this movement pattern. For each frequent outgoing edge (*out_edge*), the algorithm iterates through the trajectory IDs and their corresponding indices in the *mp_pt*. If the trajectory ID is present in both its positioning table and that of the outgoing edge, and the outgoing edge index is one greater than the that of the considered movement pattern, a new movement pattern is constructed by extending the initial movement pattern with that outgoing edge. The algorithm adds the extension index to the *pt_extended* dictionary under the new movement pattern id and trajectory ID. Finally, the extended positioning table (*pt_extended*) is returned by the algorithm. This positioning table is now of movement patterns one order greater than the movement patterns in the initial positioning table which is now of order $k + 1$.

The proposed algorithm focuses on avoiding exploring all the different combinations and appending all different movement patterns since only the outgoing edges of the last edge in the movement pattern are viable concatenations at each point. This greatly reduces the search space as well as the time complexity of the extension process. Instead of exploring $O(N^n)$ where $n$ is the max order reached with discover-able frequent sequences, $O(N \cdot E_{out})$ is explored where $E_{out}$ is the average number of outgoing edges for each edge. Also, by checking whether the last edge in the movement pattern is not the last edge in the trajectory before proceeding, unnecessary iterations are avoided which further reduces time complexity. The algorithm's time complexity is $O(N_k \cdot E_{out} \cdot M_k)$. Considering that the average number of outgoing edges is negligible comparably, then the time complexity is basically $O(N_k \cdot M_k)$. By using partitioning and parallelization however it becomes $O((N_k \cdot M_k)/P)$.

### 3.4.4 VeTraSPM Implementation

VeTraSPM is implemented by looping over and over in a recursive process of pruning the generated higher order movement patterns and extending them until the pruning results in an empty list as shown in Algorithm 6. The resulting PT tables from Algorithm 5 are passed into Algorithm 4 and the resulting set of frequent patterns $FqM_{(k+1)}$ is passed again into Algorithm 5, over and over until the resulting set of frequent patterns resulting from Algorithm 4 is empty.

Each recursion involves pruning $O(N_k \cdot M_k/P)$ and extending $O((N_k \cdot M_k)/P)$. Hence the time complexity of VeTraSPM is $O(n \cdot (N_k \cdot M_k)/P)$, where $n$ is the number of recursive steps (orders). The time complexity increases with the number of orders $n$, but each pruning and extension step ensures that only relevant patterns are kept, reducing unnecessary computations.

### 3.4.5 Calculating Sequential Impact Score

For the purpose of calculating *SIS* of each edge, as mentioned before, after each pruning performed by Algorithm 4, the resulting frequent movement patterns and confident movement pattern lists of order $k$, $FqM_k$ and $CM_k$ are saved. When the whole process is done and the recursive process ends, the $FqM$ and $CM_k$ lists are generated and for each edge appearance they are parsed and its *SIS* value is updated according to the number of its occurrences in the lists at each order as shown in Algorithm 7.

The time complexity of SIS calculation is $O(n \cdot F \cdot E)$, where $n$ is the number of iterations (orders), $F$ is the number of frequent patterns, and $E$ is the average number of edges per movement pattern. Using partitioning, the complexity reduces to $O((n \cdot F \cdot E)/P)$, making it suitable for large-scale trajectory datasets.

**Algorithm 6** VeTraSPM

```
Require: min_sup, edge_pt, out_edges, last_edge_table, P
Ensure: fqmp_res, cmp_res
Initialize an empty list fqmp_res
Initialize an empty list cmp_res
Initialize order_nb to 1
frequent_movement_patterns_list ← prune_by_support(edge_pt, min_sup)
if len(frequent_movement_patterns_list) < 1 then
    return False                    ▷ Early termination if no frequent patterns
else
    fqmp_res.append(np.array(frequent_movement_patterns_list))
    cmp_res.append(np.array([]))
end if
Initialize an empty dictionary frequent_out_edges
for x in partition do
    frequent_out_edges[x] ← [out_edge for out_edge in out_edges[x] if out_edge in frequent_movement_patterns_list]
end for
pt_extended ← edge_pt
Increment order_nb by 1
while True do
    pt_extended ← pt_ext(frequent_movement_patterns_list, pt_extended, edge_pt, last_edge_table, frequent_out_edges, P)
    frequent_movement_patterns_list ← prune_by_support(pt_extended, min_sup, P)
    if len(frequent_movement_patterns_list) < 1 then
        break                       ▷ Early termination to stop recursion
    end if
    Initialize an empty dictionary confident_rules
    edge_support ← get_sequences_support(edge_pt)
    sequences_support ← get_sequences_support(pt_extended)
    for movement_pattern in frequent_movement_patterns_list do
        edge_0 ← movement_pattern.split()[0]
        rule_conf ← sequences_support[movement_pattern] / edge_support[edge_0]
        if rule_conf ≥ 0.6 then
            confident_rules[movement_pattern] ← rule_conf
        end if
    end for
    fqmp_res.append(np.array(frequent_movement_patterns_list))
    cmp_res.append(np.array(confident_rules))
    Increment order_nb by 1
end while
return fqmp_res, cmp_res
```

**Algorithm 7** Calculate Sequential Impact Score

**Require:** *fqmp_res*, *cmp_res*, *P* (number of partitions)
**Ensure:** *sis* (NumPy array of SIS values)

```
Initialize a NumPy array sis with zeros for each edge
Split fqmp_res into P partitions
Split cmp_res into P partitions
for partition in fqmp_res do                                  ▷ Process in parallel
    for order_nb, order in enumerate(partition) do
        for fmp in order do
            for edge in fmp do
                if edge ∉ sis then
                    sis[edge] ← 1 / (order_nb + 1)
                else
                    sis[edge] ← sis[edge] + 1 / (order_nb + 1)   ▷ Efficient NumPy-based update
                end if
            end for
        end for
    end for
end for
for partition in cmp_res do                                   ▷ Process in parallel
    for order_nb, order in enumerate(partition) do
        for cmp in order do
            for edge in cmp do
                sis[edge] ← sis[edge] + 1 / (order_nb + 1)
            end for
        end for
    end for
end for
return sis
```

### 3.4.6 Optimization Strategies Used

As mentioned earlier, several optimization strategies were employed to improve the performance of the algorithm. By applying these strategies, the **VeTraSPM algorithm becomes scalable** to larger datasets. The recursive processes are sped up by partitioning, parallelization, and early termination. Memory-efficient structures and lazy evaluation reduce peak memory usage, improving scalability for large datasets as well. The improvements in both **time and space complexity** help ensure that the algorithm can handle real-world trajectory data efficiently without running into **memory bottlenecks**.

## 3.5 Experimentation and Evaluation

This section presents the experimentation and evaluation of this study. The proposed algorithm is compared with the efficient implementations of Apriori algorithm used in works (Agrawal & Srikant, 2000).

### 3.5.1 Experimental set-up and data sources

The experiments were conducted on a machine running **Windows 11** with **32GB RAM** and an **AMD Ryzen 9 6900HS @3.30GHz** (8 cores, 16 threads). All cores were utilized for parallel processing tasks, such as partitioning and processing movement patterns, edge dictionary construction, and positioning table extensions. *NumPy arrays* and *lazy evaluation* via generators were used to manage memory efficiently. The peak memory usage remained within the available 32GB RAM, while the CPU parallelization reduced execution time. No GPU acceleration was used, and data was streamed into memory to prevent overwhelming the system during XML parsing.

The chosen micro-simulation tool for the critical link analysis phase is the SUMO simulator. In this study, existing realistic scenario LuST (Codecá et al., 2017) of the city of Luxembourg is used. Additionally, the MoST scenario (Codecá & Härri, 2018) based on Monaco is also used in this work to validate the framework's applicability across cities of different sizes and transportation characteristics. The details and statistics for both scenarios are outlined in Table 3.6.

Table 3.6: SUMO Simulation Scenario Numbers

| Ref. | City | Area | Total Nodes | Total Edges | Total Trips |
|---|---|---|---|---|---|
| Codecá et al., 2017 | Luxembourg | 155.95 $km^2$ | 2,247 | 5,779 | 215,526 |
| Codecá and Härri, 2018 | Monaco | 22 $km^2$ | 2,004 | 4,404 | 7,990 |

As explained in the work (Codecá et al., 2017), to achieve authentic traffic patterns in the SUMO LuST (Luxembourg scenario), the simulation relies on genuine demographic data provided by the government. For the city's public transport component, data from the public transport database is utilized to obtain information about bus routes. The overall traffic demand encompasses buses, local, and transit mobility. While buses and transit mobility follow predetermined routes, local mobility requires generation and optimization to simulate realistic traces.

To assess the realism of the LuST Scenario's traffic demand, a dataset collected in the City of Luxembourg between March and April 2015 was used by the authors. This dataset comprises over six million Floating Car Data (FCD) samples from more than 14 thousand trips conducted between 06:00 and 22:00. Comparisons revealed similar distributions between the simulated scenario and the real dataset, indicating that LuST is capable of providing realistic traffic demand and mobility traces.

The evaluation further demonstrated that speed distributions in the simulation closely resemble those in the real dataset, with discrepancies primarily attributed to the absence of pedestrian mobility rather than changes in the road topology. Additionally, their results highlighted that dynamic rerouting enhances the interactive scenario's realism, closely aligning with precomputed optimized mobility patterns.

### 3.5.2 Parameters, Values, and Justification

In this study, the frequency of pattern occurrences serves as a quantitative measure of the prevalence of specific movement patterns within the dataset. Counting the occurrences of patterns provides us with a foundation for prioritizing patterns for further analysis. By focusing on patterns with higher occurrence frequencies, the understanding of dominant vehicular behaviors is refined while maintaining a data-driven perspective.

In order to do this, two key parameters have been identified, each with specific values and justifications. These parameters play a crucial role in fine-tuning this approach to extract meaningful patterns and relationships from vehicle trajectory data.

#### Minimum Support Threshold

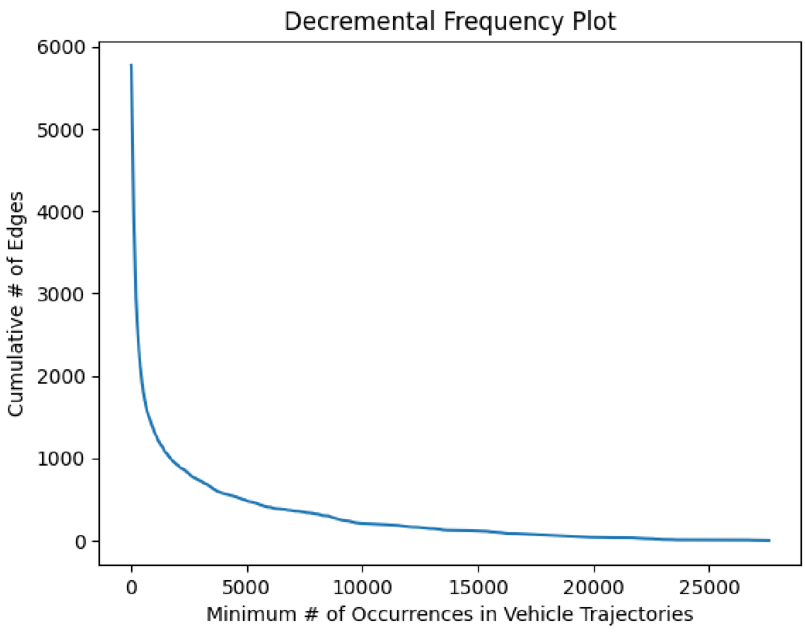



Figure 3.5: Cumulative Nb. of Frequent Edges by Minimum Nb. of Occurrences Threshold

The minimum support threshold serves as a fundamental parameter in this methodology. It determines the threshold frequency a pattern must satisfy to be considered for further analysis. To comprehensively explore patterns across various popularity orders, a spectrum of threshold values was chosen: 3%, 5%, 8%, 10%, and 12% whose respective minimum number of occurrences is (6,465), (10,776), (17,242), (21,552), and (25,863) respectively.

This range enables us to strike a balance between capturing rare patterns that might offer unique insights and identifying frequently occurring patterns that could indicate important trends.

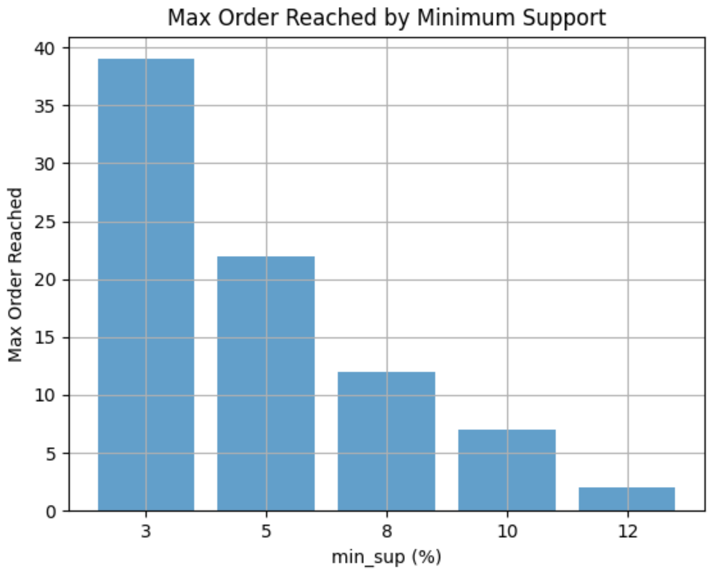



Figure 3.6: Maximum Order Reached by Minimum Support

As the minimum support threshold is manipulated, a notable phenomenon arises: the frequency of identified patterns changes. As shown in Figure 3.5, higher thresholds lead to a decrease in the number of identified patterns, as patterns must surpass a higher popularity bar to be considered. This relationship is critical as it ensures that the method remains sensitive to the minimum support threshold while carefully curating patterns with substantive implications.

### Minimum Confidence Threshold

The confidence threshold parameter is central also for the criticality link analysis process. It quantifies the strength of the rules derived from patterns, indicating the reliability of the associations between different trajectory events. The confidence threshold is expressed as a variable fraction, allowing us to adapt its value according to the characteristics of the dataset under investigation. This adaptability ensures that the rules generated accurately reflect the inherent uncertainty present in real-world vehicle movement data. In this work, different confidence values were explored: 0.6, 0.7, 0.8, 0.9, and 1. As this value increases only edges in more “confident” rules and corresponding patterns are assigned higher weight.

### Choosing Minimum Support and Confidence Thresholds

The delicate balance between including less common patterns and excluding exceedingly frequent ones is a nuanced consideration in the proposed methodology. Striking this balance ensures that both the long-tail patterns that might provide unique insights and the highly frequent patterns that may underscore critical vehicular interactions are captured. Studying different minimum support thresholds, a variation in the max order reached was observed. As the threshold increases, the max order reached decreases as shown in Figure 3.6.

Since the confidence threshold is also central in this algorithm, the variation of the confident rule (pattern) counts generated as different levels across different minimum support and confidence thresholds were studied. In order to visualize this variation, a heatmap is used as shown in Figure 3.7.

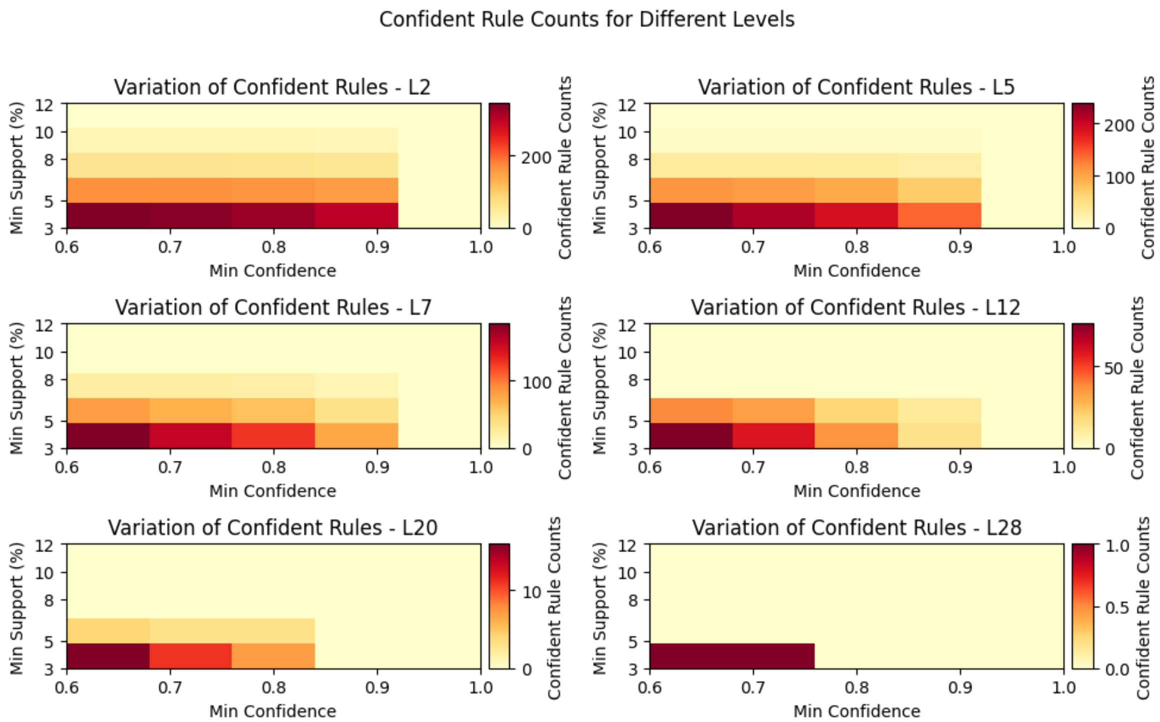



Figure 3.7: Confident Rule Counts for Different Orders

In order to strike a balance in a way so that the proposed algorithm remains versatile and adaptable to the analytical objective, a minimum support has to be chosen that is able to reach higher levels but also capture only more frequent movement patterns. Similarly, a minimum confidence threshold has to be chosen that according to the chosen support generates enough confident rules (patterns) to satisfy the algorithm at different orders.

For this reason in the experimentation 3% and 0.8 were used as the minimum support and confidence thresholds respectively. These minimum support and confidence thresholds ensure that a sufficient amount of frequent movement patterns are explored and a sufficient confident movement patterns are generated.

### 3.5.3 Results

To present the outcomes of this study, the results are divided into two key parts. The first part is dedicated to the efficiency and accuracy evaluation of VeTraSPM, where its performance is assessed against standard Apriori, enhanced Apriori, PrefixSpan, and SPADE. The second part delves into the results of the proposed $SIS$, providing insights into the network's critical links based on both frequent and confident movement patterns. Together, these results offer a comprehensive understanding of the algorithmic efficiency and impact assessment capabilities of the proposed methodologies.

### 3.5.4 Execution Time Results

The execution time of VeTraSPM is compared with several baseline algorithms: base Apriori, enhanced Apriori (which explores outgoing edges first), PrefixSpan, and SPADE. The comparison was made across varying minimum support (min_sup) thresholds, from 3% to 12% as shown in Figure 3.8. At lower thresholds like **3% min_sup**, VeTraSPM completes the task much faster than the other algorithms, where base Apriori and enhanced Apriori take significantly longer. As the **min_sup** increases, VeTraSPM continues to outperform, taking only a fraction of the time required by the other methods. For example, at **5% min_sup**, it remains significantly faster, with enhanced Apriori, base Apriori, and the other methods requiring much longer execution times. At higher thresholds, such as **10%** and **12%**, the execution times across all algorithms converge further, but VeTraSPM still maintains its edge, completing much faster, even when the search space reduces. Overall, Figure 3.8 demonstrates how VeTraSPM consistently outperforms the baseline methods across all thresholds, particularly at lower support levels where computational demand is higher.

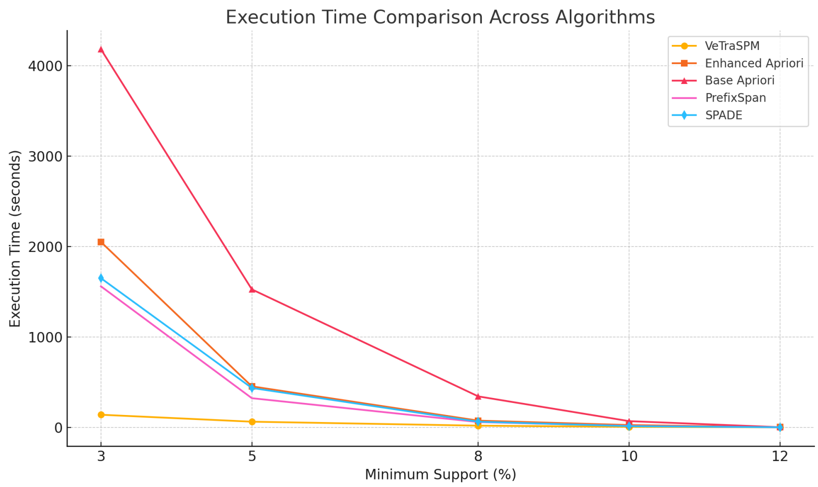



Figure 3.8: VeTraSPM Execution Time vs. Existing Implementation

VeTraSPM shows substantial efficiency gains, outperforming base Apriori by an order of magnitude at lower min_sup values due to reduced candidate generation. Despite enhancements, the Apriori variant lags behind VeTraSPM, which benefits from its optimized memory handling and vertical projection. VeTraSPM outperforms both PrefixSpan and SPADE at lower min_sup values, handling the sequential and repetitive nature of trajectory data more effectively. While PrefixSpan and SPADE improve with higher min_sup, VeTraSPM remains consistently faster.

VeTraSPM remains efficient across varying min_sup values, handling large datasets effectively. Its time complexity is reduced through vertical projection, and it uses memory more efficiently, avoiding the computational overhead seen in base Apriori.

### SIS Results

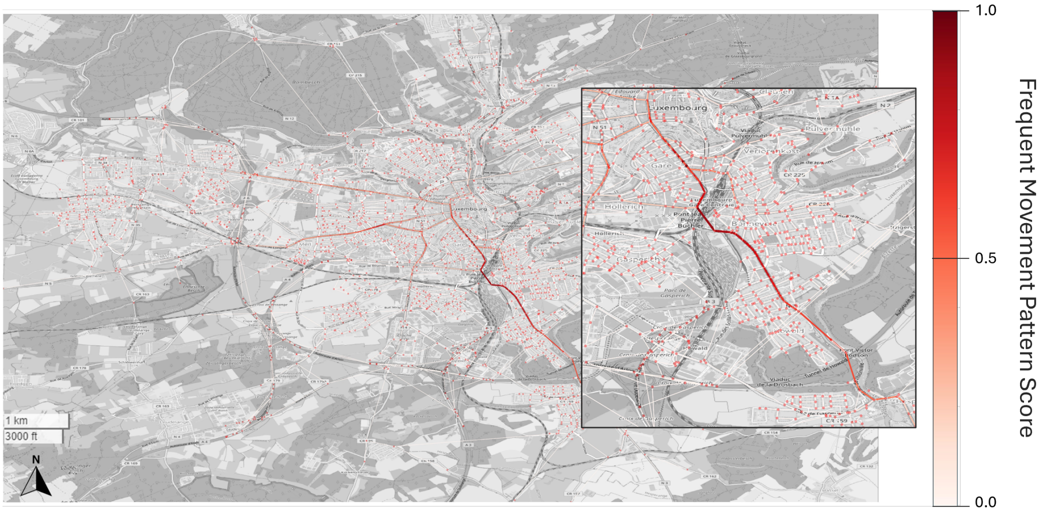



Figure 3.9: $FqM$ Score

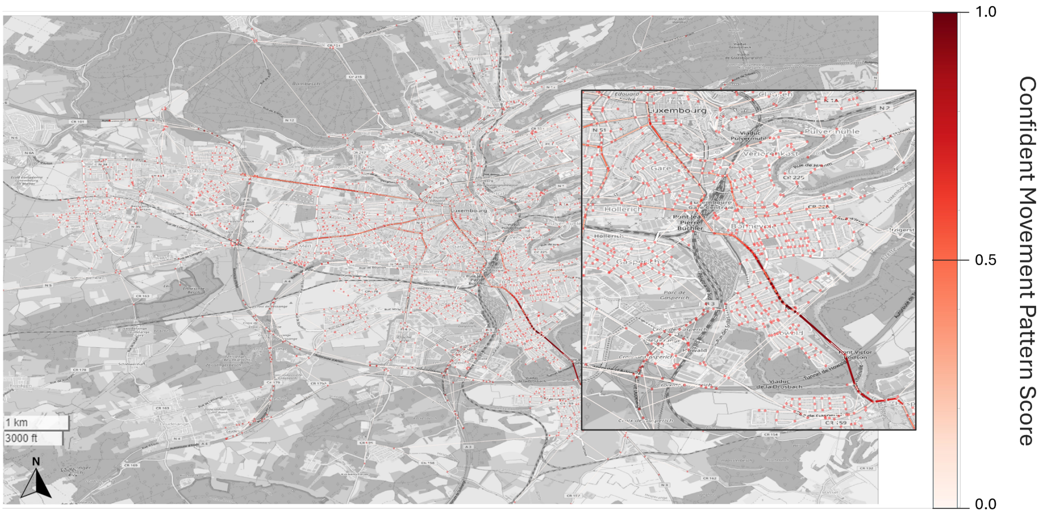



Figure 3.10: $CM$ Score

In this study, both the sets of frequent movement patterns ($FqM$) and confident movement patterns ($CM$) are leveraged to compute the proposed $SIS$.

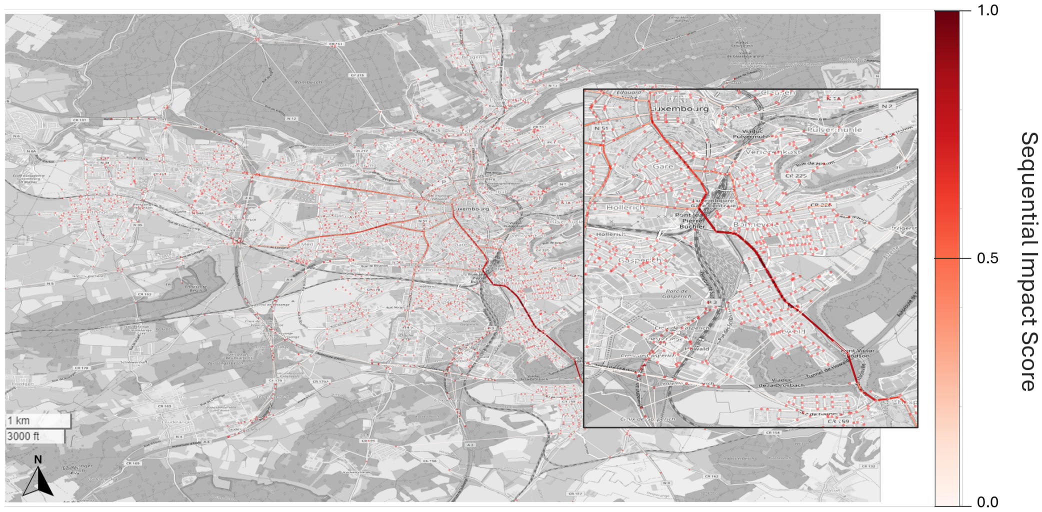



Figure 3.11: Sequential Impact Score

When visualizing the impact score derived from $FqM$, illustrated in Figure 3.9, the focus in this study is directed towards the most commonly traversed edges and sequences, placing particular emphasis on edges embedded within frequent movement sequences. Consequently, the visualization effectively highlights prominent roads within the city center, while also capturing the popularity of highways.

Conversely, the visualization of the impact score calculated through $CM$, as depicted in Figure 3.10, sheds light on frequently traversed edges within the most confident patterns. Here, a minimum confidence threshold of 0.8 ensures that there is an 80% probability of accessing these edges within a frequent pattern. Consequently, this visualization prioritizes highways and connecting edges, showcasing its preference for routes with higher confidence levels. It is noteworthy that links in the city center, which are prominently scored in $FqM$, receive a comparatively lower score in $CM$. This disparity stems from the city center environment, where the presence of alternative routes is more likely.

The introduced innovative metric, the Sequential Impact Score, integrates the insights from both $FqM$ and $CM$, thereby encapsulating distinct aspects covered by each set. The goal is to highlight critical links—those that are frequently traveled within frequent movement patterns, as well as those with a higher confidence level of being traveled. In the event of disruptions, links identified by the proposed metric are crucial, as they represent areas where alternative paths are either absent or have a low probability (20%) of existence. This comprehensive approach ensures a nuanced understanding of the significance of different links in the context of movement patterns and their potential impact on the overall network.

## 3.6 Conclusion and Future Work

In this paper, VeTraSPM is introduced, a novel algorithm for identifying critical road links in urban traffic networks through sequential pattern mining on vehicle trajectory data.

The experimental results demonstrate that VeTraSPM provides significant improvements in critical link analysis compared to existing algorithms. By leveraging efficient data structures, parallel processing, and other optimization techniques, the algorithm achieves both high scalability and computational efficiency, making it suitable for large-scale urban traffic networks. The use of the Sequential Impact Score (SIS) provides a novel approach to assessing road criticality. However, future work is required to validate $SIS$ in different traffic conditions and geographical locations.

VeTraSPM is designed to generalize across different urban contexts due to its flexibility in handling diverse trajectory data, making it adaptable to different geographical locations and time spans. However, the algorithm relies on certain assumptions about traffic patterns, which may not apply in areas with unpredictable traffic flows or highly erratic behavior. Additionally, VeTraSPM's current design is limited in its adaptability to real-time disruptions, such as accidents, events, or construction. Incorporating dynamic data streams from real-time traffic sensors would enhance its ability to respond to such changes.

Future work should explore integrating real-time traffic data feeds to enhance the algorithm's responsiveness to sudden traffic disruptions. Additional factors like weather conditions and driver behavior should be incorporated to provide a more comprehensive understanding of traffic patterns. Moreover, the algorithm could be extended to other contexts, such as pedestrian traffic or logistics networks, to broaden its applicability.

In practical terms, VeTraSPM has the potential to assist urban planners and traffic authorities in optimizing traffic management by identifying bottlenecks and critical road segments. The results generated by VeTraSPM can be visualized through tools such as heat maps and interactive dashboards, providing actionable insights for traffic managers and urban planners. VeTraSPM can be integrated into existing traffic management systems for continuous monitoring and planning of infrastructure improvements. With future modifications, it can serve as a valuable tool for real-time traffic monitoring and infrastructure planning, ultimately contributing to more efficient and resilient urban transportation systems.

# Subsequent Work

# SAMO: A Sequential Pattern Mining Model for Evaluating Road Criticality in Urban Traffic Networks

As an extension of the work presented in this chapter, we evaluated how the proposed indices—$FqM$, $CM$, and $SIS$, renamed to $SFI$, $CIS$, and $SIS$ respectively—perform when used within supervised learning models to predict road link criticality. This investigation was presented as part of a peer-reviewed conference paper in 2024.

We conducted simulations by systematically removing one link at a time and calculating the resulting change in Total Round Trip Time (TRTT). This change was treated as a proxy for the true criticality of the removed link. Using both static and dynamic features derived from the SUMO simulation output, we trained a range of machine learning models to predict these criticality values.

## Evaluation Based on Prediction Model Results and Top Features

In the criticality prediction models, besides the proposed indices, we utilized a set of static and dynamic indices derived from the SUMO simulation output. The indices used: **Static Indices:** Length, Width, Max Speed, Cost, EBC, Type. **Dynamic Indices:** Support, Relative Support, Sampled Seconds, Travel Time, Overlap Travel Time, Density, Occupancy, Speed, and Speed Relative.

We then employed various machine learning models for prediction. The mean squared error (MSE) results for each model, both with and without the proposed indices, are presented in Table 3.7.

Additionally, we further analyzed the performance of models trained on top features. Table 3.8 presents the updated rankings of $CIS$, $SFI$, and $SIS$ for selected models within the top features.

## Analysis and Insights

The inclusion of the proposed indices led to significant improvements in model performance, as evidenced by the reduction in MSE values across all models. This indicates that the proposed indices capture critical aspects of link importance that are not fully represented by traditional features. Furthermore, the analysis of top features in various models highlighted the significance of our proposed indices as follows:

- $CIS$ ranked within the top 5 features in several models, demonstrating its robustness in capturing link criticality.

Table 3.7: MSE results for machine learning models.

| **Model** | **MSE without Indices** | **MSE with Indices** | **MSE with Indices and Top Features** |
|---|---|---|---|
| Random Forest | 0.4238 | 0.3897 | 0.3782 |
| Gradient Boosting | 0.4849 | 0.4143 | 0.3977 |
| K-Nearest Neighbors | 0.4896 | 0.4258 | N/A |
| Linear Regression | 0.5116 | 0.4462 | 0.4499 |
| Ridge Regression | 0.5172 | 0.4565 | 0.4501 |
| MLP | 0.5732 | 0.4663 | N/A |
| Decision Tree | 0.8654 | 0.4665 | 0.4665 |
| Support Vector Regression | 0.5068 | 0.4912 | N/A |
| Lasso Regression | 0.7609 | 0.5698 | 0.5683 |
| Gaussian Process Regression | 170.1954 | 138.9047 | N/A |

Table 3.8: Rankings of Proposed Indices in Top Features

| **Model** | $CIS$ | $SFI$ | **SIS** |
|---|---|---|---|
| Random Forest | 4th | 18th | 27th |
| Gradient Boosting | 2nd | N/A | N/A |
| Linear Regression | 23rd | 22nd | 39th |
| Ridge Regression | 12th | 24th | 27th |
| Decision Tree | 3rd | 8th | 36th |
| Lasso Regression | 4th | N/A | N/A |

- $SFI$ and $SIS$ showed more variability in their rankings, indicating that their effectiveness might depend on the specific model and context. For instance, $SFI$ did not appear in the top features for some models but ranked 8th and 18th in others, showing moderate importance.

The most frequently top-ranked features alongside our indices include occupancy, density, ebc (edge betweenness centrality), and various traffic volume measures. Overall, our proposed indices provide a valuable addition to traditional metrics, offering a more comprehensive evaluation of link criticality. The improvement in prediction accuracy and the consistent presence of our indices among top features across different models underscore their potential for practical applications in traffic management and network optimization.

# Chapter 4

# SMaL-CLIP: Scalable MAchine Learning Framework for Critical Link Identification and Prediction

# Preface

This chapter is based on a journal article published in 2025 in the journal *Innovation and Knowledge* (Bachir et al., 2025a). It presents the final and most operational contribution of the thesis: a scalable machine learning-based framework for evaluating road segment criticality in urban road networks. Building upon the conceptual foundation laid by PEMAP and the behavioral insights derived from trajectory mining in the VeTraSPM chapter, this work marks a transition from exploratory analysis to predictive modeling. It addresses the core challenge of developing an efficient, generalizable, and data-efficient method to assess road importance across diverse urban environments.

Traditional criticality assessment approaches—whether based on simulation, graph analysis, or trajectory data—often suffer from limited scalability and require full network-level information. These constraints present a significant barrier when dealing with large or data-scarce networks. The approach proposed in this chapter overcomes these limitations by enabling supervised models to be trained on a limited subset of road segments (as little as 20%) and still achieve high predictive accuracy on the remaining links. This is made possible through the integration of a wide range of structural and dynamic features, extracted from both network attributes and simulation outputs.

The chapter introduces SMaL-CLIP, a scalable learning framework that incorporates over thirty features, including edge geometry, topological characteristics, speed, occupancy, trip times, and other flow-based indicators. These features are carefully preprocessed, normalized, and evaluated for redundancy, enabling the models to capture complex interactions within traffic dynamics. While pattern-based metrics from previous chapters—such as SIS, SFI, and CIS—are optionally included, they are treated as complementary features rather than central predictors. The core innovation lies in the ability of the framework to learn generalizable representations of link criticality from partial data.

A comprehensive experimental campaign was conducted using the LuST (Luxembourg) and MoST (Monaco) simulation scenarios. These cities were selected for their contrasting urban layouts and traffic conditions. SMaL-CLIP was evaluated in both intra-city (training and testing on the same city) and cross-city (training on one city, testing on another) settings. Results show consistent predictive performance across all scenarios, with Random Forest and Gradient Boosting achieving the highest precision (up to 73%) and lowest error rates (as low as 7% mean error). SMaL-CLIP's generalization ability, demonstrated through strong cross-city transferability, supports its applicability to real-world settings where labeled data is often limited.

This chapter also offers insights into model interpretability, scalability, and feature contribution analysis. It highlights the features most influential to prediction accuracy and discusses practical considerations for deploying such models in operational traffic systems. The methodological approach presented here sets the stage for future work in domain adaptation, temporal learning, and integration with real-time data streams.

By situating this work as the culmination of the thesis, this preface emphasizes its role in validating the overall research vision proposed through PEMAP. It illustrates how structural design, behavioral mining, and predictive intelligence can be fused into a practical tool for enhancing traffic resilience in smart cities.

## 4.1 Introduction

The evaluation of critical links within urban road networks is a pivotal task for transportation planning and traffic management. Accurate identification of these links is essential for ensuring efficient traffic flow, prioritizing infrastructure maintenance, and optimizing resource allocation (Shapouri et al., 2023). Traffic networks are increasingly complex, requiring innovative solutions to maintain their functionality, particularly as urban populations and vehicle usage continue to rise. Effective criticality analysis ensures cities can mitigate congestion, prioritize repairs, and enhance safety, thereby supporting both economic growth and quality of life for urban dwellers.

Traditional methods, such as those based on the Network Robustness Index (NRI), assess the criticality of links by analyzing the impact of removing individual links on key traffic metrics like total travel time and vehicle throughput. However, while effective in small-scale scenarios, these approaches are computationally prohibitive for large-scale networks. For example, evaluating networks with thousands of edges can require several hours of simulation per link, making real-world application infeasible. These methods often rely on simulations that do not scale well with increasing network size or complexity, leading to delays in actionable insights.

Moreover, existing methods described as predictive models in the literature often rely exclusively on structural or functional features of road networks, such as road type, length, or traffic density. This dichotomy fails to capture the interplay between static road attributes and dynamic traffic behaviors, leading to incomplete assessments of link criticality. Notably, no existing research has applied machine learning models to this domain, leaving a significant gap in leveraging advanced predictive methodologies for this critical task.

To address these challenges, we propose SMaL-CLIP, a scalable machine learning framework that predicts the impact of link disruptions using a comprehensive feature set that integrates 25 structural and functional attributes derived from road network data and traffic simulations. In addition, it incorporates three novel indices—Support Frequency Index (SFI), Confidence Impact Score (CIS), and Sequential Impact Score (SIS)—extracted from sequential pattern mining using VeTraSPM (Bachir et al., 2025b). These indices offer new insights into movement patterns and enhance the predictive capability of the machine learning models (Bachir et al., 2024).

The core innovation of SMaL-CLIP lies in its ability to generalize from only 20% of the data, making it suitable for rapid, city-scale evaluations, and demonstrating high predictive accuracy and reducing computational costs compared to traditional exhaustive simulations. A similar concept of using a smaller subset of a dataset for training was applied in another work (Zhong & Liu, 2024). This scalable methodology not only improves accuracy but also enables rapid evaluation of large networks, making it suitable for dynamic, real-world urban environments. The study hence offers the following contributions:

- Introduces a scalable method for predicting critical links in urban road networks, SMaL-CLIP, training on a small subset of links (20%) while achieving high accuracy and computational efficiency.

- Combines structural and functional features with newly proposed attributes to improve predictive precision.
- Validates scalability and generalizability through cross-city testing on the Luxembourg and Monaco road networks, covering over 4,000 edges each.
- Rigorously evaluates preprocessing strategies, feature selection, and hyperparameter tuning to ensure robust model performance.
- Demonstrates the efficacy of ensemble methods, such as Random Forest and Gradient Boosting.

By significantly reducing computational demands while maintaining high prediction accuracy, this work offers a scalable, data-driven framework for identifying critical links in urban networks. Beyond immediate applications in traffic management and infrastructure planning (Z. Wang et al., 2024; S. Zhang et al., 2024), the framework holds potential for broader use cases, such as disaster response (Bachir et al., 2023; Javadpour et al., 2023), smart city development (Cao et al., 2025; Rosca et al., 2024), and strategic resource allocation (Henke et al., 2024), reinforcing its value as a foundational tool for modern urban systems.

## 4.2 Literature Review

This section provides a structured overview of related research, covering traditional criticality indices, machine learning applications to transportation networks, and the integration of simulation tools. By presenting this organized review, we aim to clarify the research gaps that this work addresses.

### 4.2.1 Road Network Criticality Indices

Road network criticality is often evaluated using various indices that quantify the importance and vulnerability of network links. The most widely used measure is the Betweenness Centrality (BC) index, which assesses the importance of a link based on its role in facilitating traffic flow between pairs of nodes (Akbarzadeh et al., 2019; El Rashidy & Grant-Muller, 2014; Feng et al., 2022; Gauthier et al., 2018; X. Zhang & Chen, 2024). The BC index has been applied in both its standard unweighted form (M. A. Ahmed et al., 2023) and various weighted forms, incorporating factors such as traffic flow, link length, and congestion (Akbarzadeh et al., 2019; Gauthier et al., 2018).

Several studies have extended the BC index by integrating it with other indices. For instance, Feng et al., 2022 combined BC with link length, clustering coefficient, and road network connectivity indices. Similarly, Li et al., 2020 merged weighted BC with the flow index, and El Rashidy and Grant-Muller, 2014 considered BC alongside flow, length, link capacity, and congestion.

Beyond BC, other works have explored alternative indices to assess road network criticality. Some studies focused solely on flow indices or included demand-based metrics

(Jenelius, 2010; Jenelius et al., 2006; Scott et al., 2006; Sullivan et al., 2010). These indices provide a multifaceted view of network vulnerability but often rely on specific traffic models and assumptions.

### 4.2.2 Machine Learning in Transportation Networks

Machine learning techniques have been increasingly applied to transportation networks, particularly in areas such as traffic flow prediction, incident detection, and demand forecasting. These approaches leverage real-time and historical data to model complex traffic patterns and provide predictive insights. Below, we highlight key studies applying machine learning to various transportation challenges.

Ma et al., 2015 utilized Long Short-Term Memory (LSTM) networks to predict traffic speed using remote microwave sensor data, demonstrating the ability of recurrent neural networks to capture temporal dependencies in traffic data. Similarly, Lv et al., 2015 proposed a deep learning approach for traffic flow prediction using convolutional neural networks (CNNs), which effectively model spatial dependencies in large-scale traffic datasets.

Incorporating multi-source urban data has further advanced personalized and context-aware recommendations in transportation systems. Liu et al., 2022 developed a framework integrating urban data sources, such as user behavior and contextual factors, for personalized multi-modal transportation recommendations. This highlights the potential of combining diverse data inputs with deep learning for intelligent decision-making in urban mobility. Desjardins and Chaib-draa, 2011 applied reinforcement learning for Cooperative Adaptive Cruise Control (CACC) in connected vehicles, optimizing dynamic network interactions.

In related domains such as network robustness and infrastructure analysis, graph-based machine learning techniques have been applied to identify critical nodes in communication networks and vital infrastructure systems. Yang and An, 2020 employed graph-based learning for node ranking, while Asgharian Rezaei et al., 2023 used collective feature engineering to predict node importance in power grids. These approaches highlight the growing intersection of graph analytics and machine learning in complex network analysis.

Despite these advances, there remains a critical gap in the application of machine learning to directly predict critical links in urban road networks. Existing studies focus primarily on nodes or overall traffic prediction, with limited attention to link-level criticality in the context of dynamic traffic disruptions. The proposed framework addresses this gap by integrating machine learning with dynamic and structural features, enabling scalable and data-driven criticality assessment at the link level.

### 4.2.3 Simulation-Based Traffic Analysis

Microscopic simulation tools such as TransCAD, OmniTrans, and SUMO play a crucial role in evaluating network performance under various scenarios, including link disruptions (Codecá & Härri, 2018; Codecá et al., 2017; El Rashidy & Grant-Muller, 2014; Elsafdi, 2020; Khan et al., n.d.; Lee et al., 2022; Li et al., 2020; Scott et al., 2006; Tian et al., 2021).

SUMO, in particular, is valued for its open-source flexibility, making it ideal for simulating urban traffic dynamics and integrating with advanced analytical tools.

The integration of simulation outputs with machine learning models has gained traction in recent years, particularly in disaster response and urban resilience planning. Simulation data provides synthetic yet highly granular insights into traffic patterns, which machine learning models can leverage to develop predictive models for system vulnerabilities. SMaL-CLIP builds on this trend, using SUMO-based simulation data as a rich feature source for training predictive models capable of assessing link criticality in real-world urban networks. Table 4.1 summarizes key microscopic simulation tools used in the literature.

Table 4.1: Microscopic Simulation Tools in Literature

| **Ref.** | **Tool** | **Application** | **License** |
|---|---|---|---|
| (Elsafdi, 2020; Tian et al., 2021) | PTV Vissim | Multi-modal | Commercial |
| (Lee et al., 2022) | FLO-2D | Field-specific (floods) | Commercial |
| (Scott et al., 2006) | TransCAD | Travel demand forecasting | Commercial |
| (Li et al., 2020) | MATLAB R2014a | Multi-modal | Commercial |
| (El Rashidy & Grant-Muller, 2014) | OmniTrans | Multi-modal and multi-temporal | Commercial |
| (Codecá & Härri, 2018; Codecá et al., 2017; Khan et al., n.d.) | SUMO | Multi-modal and multi-temporal | Open-source |

Together, these three strands—traditional criticality indices, machine learning for network analysis, and simulation-based traffic evaluation—form the foundation upon which SMaL-CLIP is built. By synthesizing insights from all three areas, we develop a scalable and predictive methodology for real-time link criticality assessment in urban road networks.

## 4.3 Materials and Methods

The proposed framework employs machine learning to predict link criticality in urban road networks. It integrates data preprocessing, feature engineering, and model training to provide accurate and scalable predictions. The process begins with the collection of structural and dynamic data from road networks and simulation platforms. This data is then processed to extract relevant features, including novel indices derived from vehicle trajectory patterns. Machine learning models are trained using this enriched dataset to predict criticality, which is validated through performance metrics.

In this section, we describe the system architecture, data acquisition process, and preparation steps. We also provide details about the features used, the model training process, and the evaluation metrics applied to validate the framework's performance.

### 4.3.1 General System

SMaL-CLIP aims to identify critical links in urban road networks through a modular and scalable machine learning approach. Rather than relying on computationally intensive network-wide simulations, the system trains machine learning models on a small subset of the network (20% of links) and uses these trained models to predict the criticality of the remaining links. This design significantly reduces computational overhead while maintaining high predictive accuracy, with the best models achieving approximately 7% mean percentage error.

The system consists of three primary components: data preparation and labeling, model training, and prediction with validation. During data preparation, structural, functional, and proposed dynamic features are extracted from the road network and from SUMO simulation datasets. Link closure simulations are conducted to compute the total trip time difference, which serves as the criticality label. In the model training phase, machine learning algorithms are trained on 20% of the network's links using the enriched feature set. The trained models are then applied to predict the criticality of the remaining links, with performance validated using precision and Percentage Root Mean Square Error (PRMSE). This modular architecture allows for flexible feature integration and easy adaptation to different urban scenarios.

### 4.3.2 System Architecture

The overall system architecture follows a structured, sequential process to ensure consistency and efficiency. Six primary stages define this pipeline, from initial data ingestion to final evaluation. Each stage plays a specific role in ensuring the quality of the data, the reliability of the model, and the accuracy of predictions.

**Stage 1 - Input Data:** The process starts with the collection of urban road network data and traffic simulation data obtained from SUMO. This input includes structural attributes (length, width, road type) and functional data (traffic volume, speeds, occupancy).

**Stage 2 - Feature Engineering:** Thirty structural, functional, and dynamic features are extracted from the input data. This feature set combines conventional road attributes with novel indices derived from sequential pattern mining of vehicle trajectories, enhancing the predictive power of the model.

**Stage 3 - Data Labeling:** For each road link, a criticality label is computed by simulating individual link closures and measuring the resultant change in total trip time across the network. This change, adjusted for variations in vehicle counts, serves as the ground truth criticality label.

**Stage 4 - Training Data Selection:** To enhance scalability, the framework uses only

20% of the network's links for training. This subset is selected to ensure diverse coverage of network characteristics, providing a representative sample for model learning. The remaining 80% is reserved for validation, allowing the framework to demonstrate its generalizability.

**Stage 5 - Model Prediction:** Using the trained machine learning models, the framework predicts criticality scores for the unseen 80% of links. The models incorporate regression techniques capable of capturing both linear and non-linear relationships between features and criticality.

**Stage 6 - Evaluation:** Finally, model performance is evaluated using precision and PRMSE, capturing both predictive accuracy and the framework's ability to prioritize the most critical links. These metrics offer a balanced view of overall performance and practical utility.

The full architecture is illustrated in Figure 4.1, showing the seamless integration of these stages.

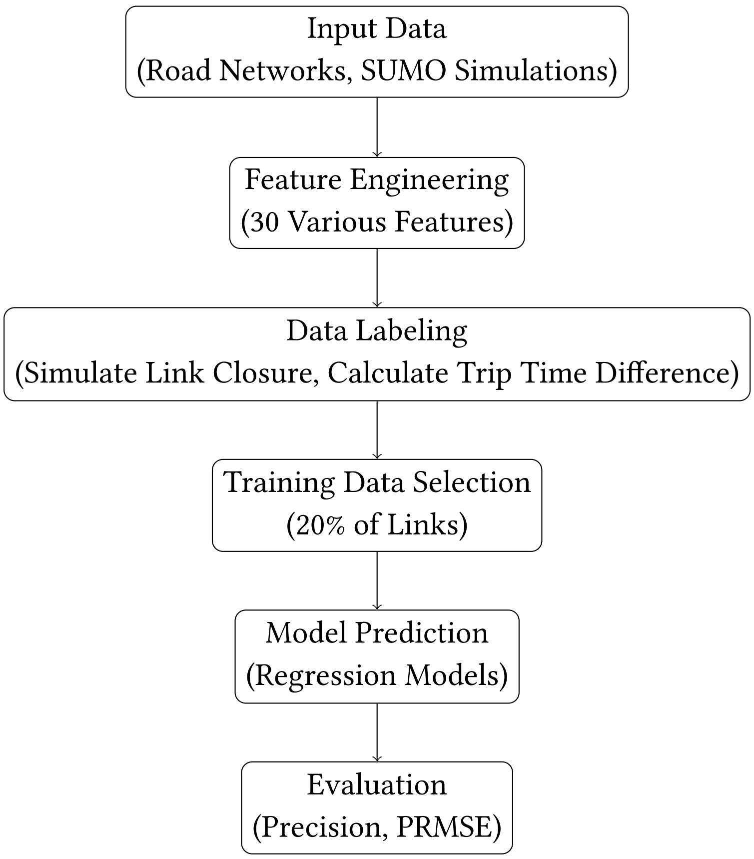



Figure 4.1: System Architecture of the SMaL-CLIP.

### 4.3.3 Data Acquisition and Preparation

Two large-scale datasets, Luxembourg (LuST) (Codecá et al., 2017) and Monaco (MoST) (Codecá & Härri, 2018), were selected for this study, as detailed in Table 4.2. These cities were chosen because they offer the most comprehensive and up-to-date simulation scenarios available in the Simulation of Urban MObility (SUMO) platform. Both the LuST and MoST scenarios are built upon real-world data and provide detailed traffic flow dynamics and static road network attributes, making them ideal candidates for evaluating the proposed machine learning framework.

The selection of these cities also ensures diversity in urban characteristics. The LuST dataset consists of 2,247 nodes and 5,779 edges, covering an area of approximately 155.95 km$^2$, and includes 215,526 simulated trips. It captures a wide variety of traffic conditions, including peak and off-peak hours, representative of a larger and more diverse urban network. Conversely, the MoST dataset contains 2,004 nodes and 4,404 edges within a compact area of 22 km$^2$, with 7,990 simulated trips. This dataset provides insights into a dense and tightly connected urban environment. By including two cities with differing scales and urban structures, we validate the scalability and generalizability of SMaL-CLIP across varied urban traffic networks.

Table 4.2: SUMO Simulation Scenario Numbers

| **Attribute** | **LuST** | **MoST** |
|---|---|---|
| **City** | Luxembourg | Monaco |
| **Area (km$^2$)** | 155.95 | 22.00 |
| **Total Nodes** | 2,247 | 2,004 |
| **Total Edges** | 5,779 | 4,404 |
| **Total Trips** | 215,526 | 7,990 |
| **Year** | 2017 | 2018 |

This study employed the Simulation of Urban MObility (SUMO), an open-source microscopic traffic simulator, to generate both structural and functional indices based on real-life data from the two cities. The SUMO simulations provided specific indices for each edge (link) within the road network.

### Label Extraction

To evaluate the criticality of each road link, we conducted extensive simulations using SUMO by systematically removing one link at a time and recording the corresponding changes in various traffic statistics. The magnitude of these changes served as a quantitative indicator of the link's importance within the network. The recorded performance metrics included total trip time, vehicle count, average speed, waiting time, and other dynamic traffic parameters. Table 4.3 presents a sample of these recorded metrics for a specific road link, comparing the baseline scenario (where no road is blocked) with the scenario where the link is blocked. These metrics illustrate the significant changes in key traffic indicators caused by the removal of a critical link.

The criticality label for each link is computed as a composite metric combining two primary components: the Vehicle Count Difference (VCD) and the Total Trip Time Difference

Table 4.3: Simulation Results Before/After a Sample Road Blockage

| **Parameter** | **Baseline (No Road Blocked)** | **Road Blocked** |
|---|---:|---:|
| Count | 7,580 | 7,481 |
| Route Length (m) | 6,979.08 | 6,663.64 |
| Speed (m/s) | 14.23 | 13.12 |
| Duration (s) | 733.82 | 2,538.81 |
| Waiting Time (s) | 11.53 | 1,865.40 |
| Time Loss (s) | 79.99 | 1,925.16 |
| Depart Delay (s) | 0.05 | 0.04 |
| Depart Delay Waiting (s) | 0.00 | 8,316.59 |
| Total Travel Time (s) | 5,562,390.75 | 18,992,871.25 |
| Total Depart Delay (s) | 380.00 | 366,252.25 |

Table 4.4: Calculated Labels for Sample Blocked Edges

| **Edge** | **VCD** | **TTD** | **TTD$_t$** |
|---|---|---|---|
| Edge 3150 | 0.791 | 0.703 | 0.747 |
| Edge 5096 | 0.978 | 0.392 | 0.685 |
| Edge 5095 | 0.977 | 0.385 | 0.681 |
| Edge 3969 | 0.936 | 0.385 | 0.661 |

(TTD). The VCD captures the proportion of vehicles unable to reach their destinations due to the blockage of a specific link. It is calculated using the formula:

$$\text{VCD}'(\text{edge}) = \frac{\text{baseline\_vc} - \text{edge\_removed\_vc}}{\text{baseline\_vc}} \tag{4.1}$$

To ensure consistent comparisons across all links, the VCD is normalized to a range between 0 and 1 using:

$$\text{VCD}(\text{edge}) = \frac{\text{VCD}'(\text{edge}) - \min(\text{VCD}')}{\max(\text{VCD}') - \min(\text{VCD}')} \tag{4.2}$$

Here, $vc$ refers to the vehicle count shown in the first row of Table 4.3.

In parallel, the Total Trip Time Difference (TTD) reflects the increase in total travel time caused by the blockage and is calculated using:

$$\text{TTD}'(\text{edge}) = \frac{\text{baseline\_ttt} - \text{edge\_removed\_ttt}}{\text{baseline\_ttt}} \tag{4.3}$$

This value is similarly normalized to a range between 0 and 1 using:

$$\text{TTD}(\text{edge}) = 1 - \frac{\text{TTD}'(\text{edge}) - \min(\text{TTD}')}{\max(\text{TTD}') - \min(\text{TTD}')} \tag{4.4}$$

where $ttt$ corresponds to the total travel time shown in the highlighted row of Table 4.3.

The final criticality label, the *Edge Trip Time Difference* ($TTD_t$), is computed by combining the normalized VCD and TTD with equal weight:

$$\text{TTD}_t = 0.5 \cdot \text{TTD} + 0.5 \cdot \text{VCD} \tag{4.5}$$

This combined score ensures that both the direct impact of vehicle loss and the indirect congestion effect are accounted for. It serves as the ground truth label for training and evaluating the machine learning models. Table 4.4 presents example VCD, TTD, and $TTD_t$ values for a subset of blocked edges, providing insight into how different links contribute to network-level disruptions.

Table 4.5: Description of Features for Road Network Analysis

| **Feature** | **Unit** | **Description** | **Type** |
|---|---|---|---|
| Length | m | Length of the edge | Structural |
| Width | m | Width of the edge | Structural |
| Max Speed | m/s | Maximum speed limit on the edge | Structural |
| Cost | - | Cost associated with traversing the edge | Structural |
| EBC | - | Edge Betweenness Centrality | Structural |
| Type | - | Categorical attribute indicating the type of road (e.g., highway) | Structural |
| Support | - | Frequency with which a specific pattern occurs in the dataset | Functional |
| Relative Support | - | Support normalized to the total number of patterns | Functional |
| Sampled Seconds | s | Number of seconds vehicles were measured on the edge | Functional |
| Travel Time | s | Time required for a vehicle to traverse the edge | Functional |
| Overlap Travel Time | s | Time required for any part of a vehicle to traverse the edge | Functional |
| Density | veh/km | Vehicle density on the edge | Functional |
| Lane Density | veh/km | Vehicle density per lane on the edge | Functional |
| Occupancy | % | Percentage of time the edge is occupied by vehicles | Functional |
| Average Speed | m/s | Mean speed of vehicles on the edge | Functional |
| Speed Relative | - | Ratio of mean speed to the total average speed of all cars | Functional |
| Departed | count | Number of vehicles that departed from the edge | Functional |
| Arrived | count | Number of vehicles that arrived at the edge | Functional |
| Entered | count | Number of vehicles that entered the edge | Functional |
| Left | count | Number of vehicles that left the edge | Functional |
| Lane Changed From | count | Number of lane changes from the edge | Functional |
| Lane Changed To | count | Number of lane changes to the edge | Functional |
| Avg Num Vehicles | count | Average number of vehicles present on the edge | Functional |
| Avg Traffic Volume | veh/h | Average traffic volume on the edge | Functional |
| Traffic Volume Begin | veh/h | Traffic volume at the beginning of the interval | Functional |
| Traffic Volume End | veh/h | Traffic volume at the end of the interval | Functional |
| Total Distance Travelled | km | Total distance travelled by all vehicles on the edge | Functional |
| Support Frequency Index (SFI) | - | Aggregates weighted support of edge occurrences | Dynamic |
| Confidence Impact Score (CIS) | - | Aggregates weighted confidence of edge occurrences | Dynamic |
| Sequential Impact Score (SIS) | - | Combines SFI and CIS to provide a sustained importance of edges | Dynamic |

### Feature Extraction

The assessment of link criticality involved the use of a total of 27 indices—both structural and functional—derived from SUMO simulation outputs, as well as 3 novel dynamic indices proposed in previous work. All features are detailed in Table 4.5.

While the framework integrates a diverse set of structural, functional, and dynamic indices derived from simulations and trajectory analysis, it is important to acknowledge that several additional features — such as topographical characteristics, road curvature, elevation profiles, and contextual land use data — may further enhance model performance. However, the inclusion of such attributes is beyond the scope of this work. In this study, we focus on commonly used features in the literature and introduce novel trajectory-based dynamic features to validate the proposed methodology. The selected set was sufficient to demonstrate the scalability and generalizability of the framework. The integration of additional feature dimensions, including topographic or contextual variables, represents a promising direction for future work.

*Structural indices* provide inherent characteristics of the road network's edges and lanes, independent of traffic flow at any given time. *Functional indices* capture the dynamic state of traffic on each edge, reflecting variations in density, speed, and volume. In addition to these, we incorporated three novel *dynamic indices* derived from sequential pattern mining of vehicle trajectories, introduced in our previous work.

Together, these indices offer a comprehensive representation of traffic dynamics and network behavior, significantly enhancing the accuracy of criticality assessment and the predictive performance of the machine learning models. The derived features were carefully selected to capture both localized and sequential behavioral patterns, thereby enriching the dataset and improving model generalizability.

### Data Preparation

To ensure consistency and enhance model performance, all features and labels undergo a standardized data preparation process. First, numerical features are scaled using standardization (zero mean, unit variance) to account for different magnitudes across attributes like length, speed, and density. Labels are normalized using Min-Max scaling, mapping all criticality scores to the [0, 1] range, which improves model convergence and interpretability.

Missing data is handled using statistically appropriate techniques. Numerical attributes are imputed using the median or mean, depending on data distribution, while categorical attributes, such as road type, are imputed using the mode. This ensures the dataset is complete without introducing biases that could skew model training. Categorical attributes are further processed through one-hot encoding, converting them into binary vectors suitable for machine learning models.

Finally, summary statistics and visual inspections are performed to validate the completeness and consistency of the preprocessed dataset, ensuring all features align across training and testing sets. This structured data preparation process enhances the robustness and reliability of subsequent modeling stages.

### Data Splitting and Cross-City Validation

The dataset is split into training and testing subsets to evaluate the framework's performance. Consistent with the framework's design, only 20% of the network's links are used for training, while the remaining 80% are reserved for validation. This reflects the core goal of ensuring accurate predictions with minimal training data, enabling scalability to larger networks.

It is important to note that the 20% training split was chosen empirically after experimenting with various thresholds. We tested lower training proportions, including values as low as 8%, and observed the resulting impact on prediction accuracy and the quality of learned patterns. A threshold of 20% was found to provide a good balance between model performance and training efficiency, particularly in terms of accurately capturing critical links. This choice is not intended to be fixed or city-specific; rather, it demonstrates the feasibility of training on a small subset of the network. Further optimization of this threshold, potentially guided by expert feedback or adaptive criteria, is left for future work.

Training links are selected to ensure they represent diverse conditions across the network, capturing different levels of criticality and varying traffic conditions. Stratified sampling is used where appropriate to ensure balanced coverage of critical and non-critical links. During training, hyperparameter tuning is performed using cross-validation within the training set to optimize model performance.

To further assess the framework's robustness, cross-city validation experiments are conducted. In these experiments, the model is trained on one city (e.g., Luxembourg) and tested on the other (e.g., Monaco). This tests the framework's ability to generalize across cities with different layouts, traffic conditions, and network structures, demonstrating its portability to new urban environments.

## 4.3.4 Machine Learning Models and Evaluation Metrics

SMaL-CLIP leverages a variety of machine learning algorithms to predict link criticality in urban road networks. These include both linear and non-linear models, ensemble methods, and neural networks to account for diverse relationships between features and criticality. Specifically, we evaluate the performance of Random Forest, Gradient Boosting, Decision Trees, Support Vector Regression, Linear Regression, Ridge Regression, Lasso Regression, K-Nearest Neighbors, Multi-Layer Perceptron, and Gaussian Process Regression. This broad selection ensures that both simple and complex patterns within the data are captured, allowing the framework to adapt to different urban contexts and data distributions.

The models' performance is evaluated using two complementary metrics. The first metric, **Percentage Root Mean Squared Error (PRMSE)**, quantifies overall prediction accuracy by comparing predicted criticality scores to actual labels. PRMSE is normalized to the mean criticality value to ensure comparability across different datasets and urban settings.

It is computed as:

$$\text{PRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{\overline{y}} \times 100 \tag{4.6}$$

where $y_i$ is the actual criticality, $\hat{y}_i$ is the predicted criticality, and $\overline{y}$ is the mean criticality across all links. This metric captures the overall accuracy of the predictions relative to the typical criticality in the dataset.

The second evaluation metric, **Precision**, focuses specifically on the framework's ability to correctly identify the most critical links. This metric is particularly relevant in real-world applications where transportation planners must prioritize interventions on the most impactful links. Precision is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.7}$$

where a true positive represents a link correctly identified as highly critical, and a false positive represents a link incorrectly predicted to be critical. Precision highlights the framework's effectiveness in prioritizing critical infrastructure under limited resource conditions.

Together, PRMSE and Precision provide a balanced evaluation of the framework's predictive accuracy and its practical utility in identifying the most critical road segments, ensuring the framework is both analytically rigorous and operationally useful.

### 4.3.5 Machine Learning Pipeline

The proposed framework incorporates a structured machine learning pipeline designed to efficiently process road network data, extract relevant features, train predictive models, and evaluate their performance. This pipeline ensures that all components—from raw data processing to model validation—are systematically integrated, maintaining both accuracy and scalability across different urban environments. The full process is illustrated in Figure 4.2.

The pipeline begins with the ingestion of input data, consisting of structural and functional attributes extracted from urban road networks, combined with dynamic traffic indicators derived from SUMO simulations. These data sources are preprocessed and merged into a comprehensive dataset containing 30 features for each link. Alongside these features, the criticality labels (i.e., trip time difference scores) computed from the simulation-based link removal experiments are included, forming the target variable for model training.

In the feature engineering and selection step, both the complete feature set and selected subsets are evaluated to determine the optimal feature configuration. This step ensures that redundant or less informative attributes are excluded, improving both computational efficiency and model interpretability. While some experiments use all available features, others apply feature selection techniques to identify the most predictive subset.

Once features are finalized, the dataset is split into training and testing sets. The model is trained on 20% of the network's links, with hyperparameter tuning applied where applicable. This tuning process uses internal cross-validation within the training data, optimizing model parameters to minimize prediction error. The trained model is then used to predict the criticality of the remaining 80% of the links.

For each predicted link, the framework outputs a criticality score that can be directly compared to the ground truth label derived from SUMO simulations. To assess performance, the framework calculates two key metrics: Percentage Root Mean Squared Error (PRMSE) and Precision. PRMSE measures the overall predictive accuracy across all links, while Precision evaluates the model's ability to correctly identify the most critical links, which is particularly important for real-world applications like prioritizing road maintenance or optimizing traffic management interventions.

This sequential and modular design allows the pipeline to be both adaptable and extensible. New features or indices can be seamlessly integrated into the feature engineering phase, and different machine learning models can be substituted or combined within the training phase without disrupting the broader workflow. The combination of flexibility, scalability, and accuracy ensures that the framework can be applied across cities with varying network sizes, topologies, and traffic dynamics.

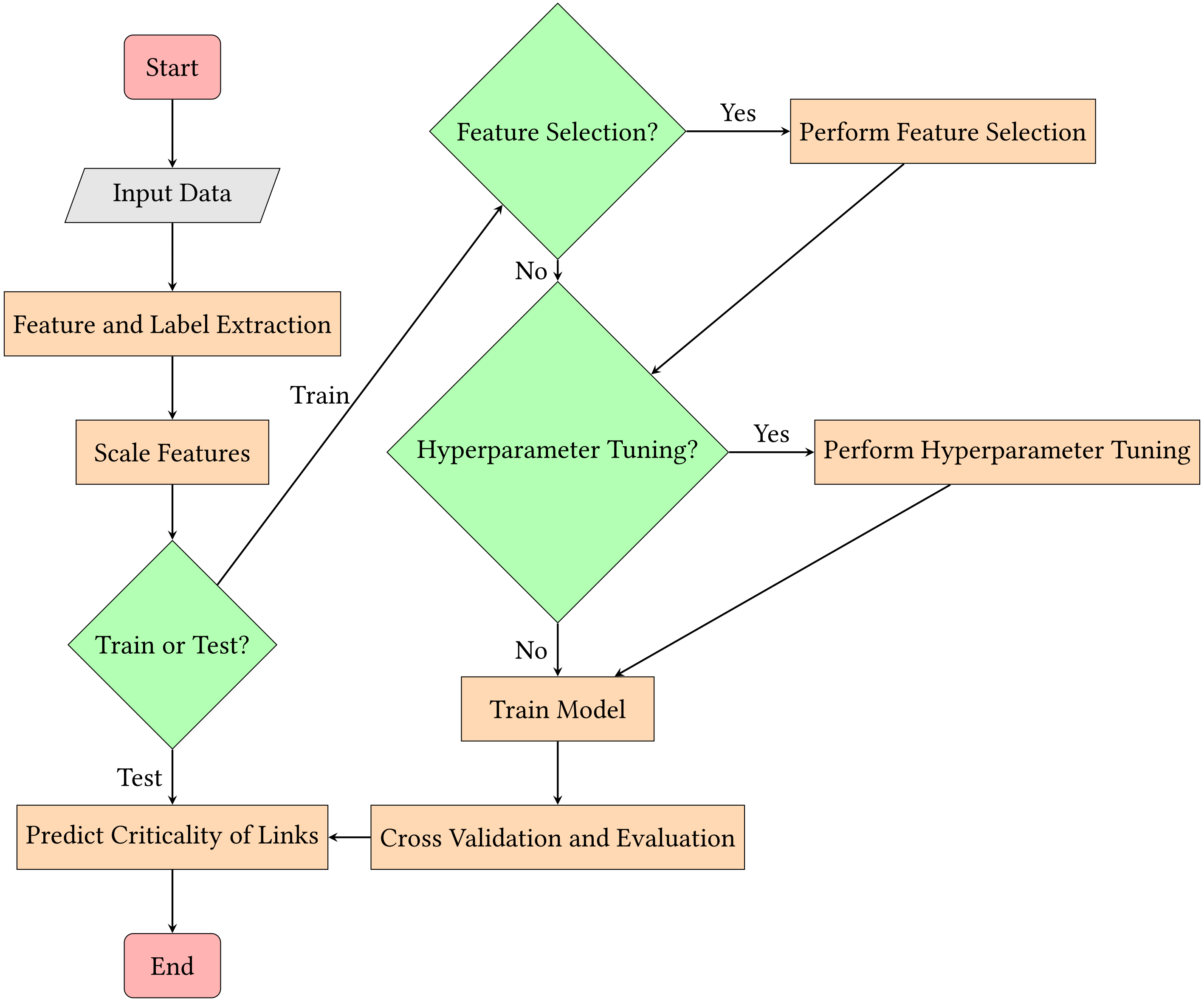



Figure 4.2: Machine Learning Pipeline

### 4.3.6 Experimental Setup

The experimental setup was designed to evaluate the proposed framework across multiple configurations, feature combinations, and city scenarios. Both Luxembourg (LuST) and Monaco (MoST) networks were used to validate the framework under both intra-city and cross-city experiments. These experiments assess how well the framework predicts criticality within a known network and how effectively models trained in one city can generalize to another urban setting with different structural and traffic characteristics.

The experiments followed a standardized process where 20% of the links in each network were used for training, while the remaining 80% were reserved for testing. This limited training data configuration highlights the scalability and practical applicability of the framework, simulating conditions where comprehensive training data is unavailable.

To understand the contribution of different data types and processing steps, multiple configurations were tested, incorporating varying combinations of structural, functional, and proposed features. The impact of preprocessing techniques such as feature selection and hyperparameter tuning was also assessed, providing insight into how these steps influence model performance. Cross-city experiments, where models were trained on one city and tested on the other, further evaluated the framework's generalizability and robustness.

The table below, Table 4.6, summarizes the tested configurations, detailing the training and testing city, the included feature types, and whether feature selection and hyperparameter tuning were applied. This structured overview provides transparency into the experimental design and allows for reproducibility.

This comprehensive suite of experiments allows for a detailed evaluation of how different feature types and preprocessing techniques influence prediction accuracy, both within and across cities. It also provides insight into the balance between computational efficiency and predictive power, which is essential for real-world deployment in urban traffic management systems.

Table 4.6: Configurations with Input Features, Processing Steps, and City Scenarios

| Configurations | City Scenarios | | Features | | | Processing Steps | |
|---|---|---|---|---|---|---|---|
| | Train | Test | Structural | Functional | Proposed | Feature Selection | Tuning |
| **LuST Configurations (Train: LuST, Test: LuST)** | | | | | | | |
| LLS (Baseline) | LuST | LuST | ✓ | | | | |
| LLSP | LuST | LuST | ✓ | | ✓ | | |
| LLSF | LuST | LuST | ✓ | ✓ | | | |
| LLSFP | LuST | LuST | ✓ | ✓ | ✓ | | |
| LLSFPE | LuST | LuST | ✓ | ✓ | ✓ | ✓ | |
| LLSFPT | LuST | LuST | ✓ | ✓ | ✓ | | ✓ |
| LLSFPET | LuST | LuST | ✓ | ✓ | ✓ | ✓ | ✓ |
| **MoST Configurations (Train: MoST, Test: MoST)** | | | | | | | |
| MMS (Baseline) | MoST | MoST | ✓ | | | | |
| MMSP | MoST | MoST | ✓ | | ✓ | | |
| MMSF | MoST | MoST | ✓ | ✓ | | | |
| MMSFP | MoST | MoST | ✓ | ✓ | ✓ | | |
| MMSFPE | MoST | MoST | ✓ | ✓ | ✓ | ✓ | |
| MMSFPT | MoST | MoST | ✓ | ✓ | ✓ | | ✓ |
| MMSFPET | MoST | MoST | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Cross-City Configurations** | | | | | | | |
| LMSFP | LuST | MoST | ✓ | ✓ | ✓ | | |
| MLSFP | MoST | LuST | ✓ | ✓ | ✓ | | |

## 4.4 Results

This section provides a detailed evaluation of SMaL-CLIP using the configurations outlined in Table 4.6. The results are organized into three subsections: same-city evaluations (Luxembourg and Monaco), cross-city generalization, and model-specific performance improvements. Metrics such as Percentage Root Mean Squared Error (PRMSE) and precision are used to assess the framework's predictive accuracy and ability to identify critical links.

### 4.4.1 Same-City Evaluation

The framework was evaluated on the LuST and MoST datasets using configurations LLS/MMS to LLSFPET/MMSFPET, progressively incorporating structural, functional, and proposed features, as well as processing steps like feature selection and hyperparameter tuning.

#### Luxembourg (LuST Dataset)

The same-city evaluation on the LuST dataset demonstrated progressive improvements in performance as additional features and preprocessing steps were incorporated. Figure 4.3 presents the precision and PRMSE for configurations LLS to LLSFPET.

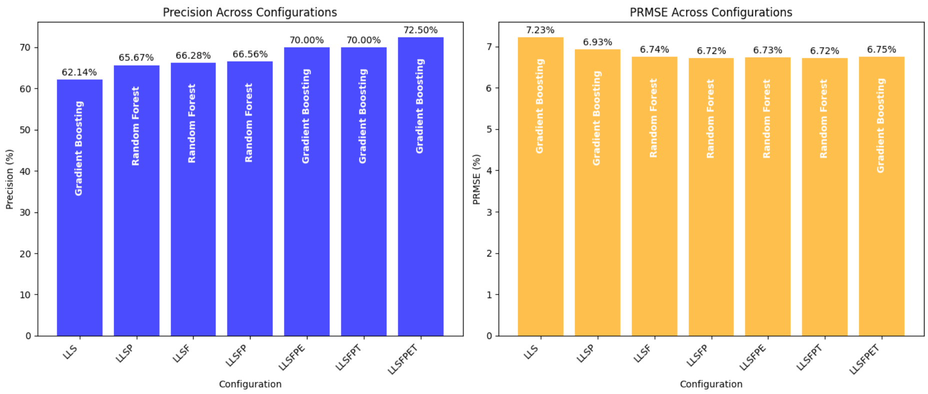



Figure 4.3: Performance Metrics for Same-City Evaluation on LuST Dataset

The following observations were made:

- The baseline configuration (LLS) achieved a precision of 62.14% and a PRMSE of 7.22%.
- Incorporating proposed features (LLSP) improved precision by 3.53% while reducing PRMSE by 0.29%.
- Adding all feature types without preprocessing (LLSFP) yielded a precision of 70.00%, marking a substantial improvement of 7.86% over the baseline.

- The best configuration (LLSFPET) achieved a precision of 72.00% and a PRMSE of 6.70%, demonstrating the impact of feature selection and hyperparameter tuning.

Furthermore, the comparison between ground truth and model predicted top edges are visualized in Figure 4.5.

### Monaco (MoST Dataset)

A similar evaluation was conducted on the MoST dataset using configurations MMS to MMSFPET. The Monaco dataset exhibited similar trends, with precision increasing significantly as additional features and preprocessing steps were incorporated. Figure 4.4 summarizes the results for configurations MMS to MMSFPET.

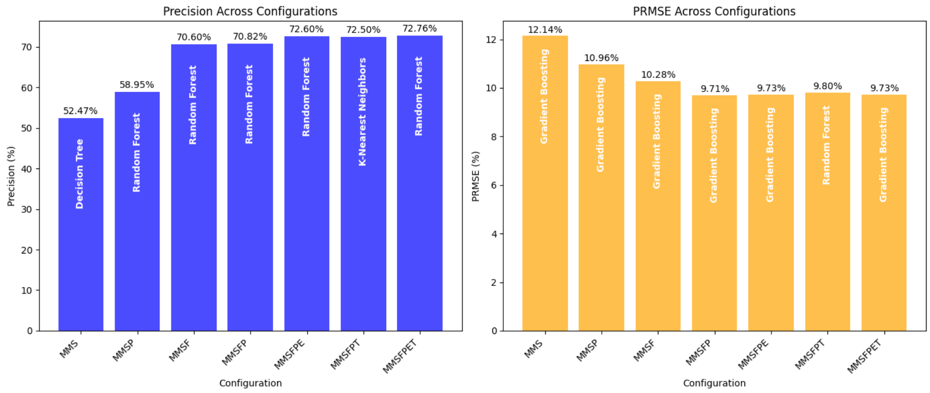
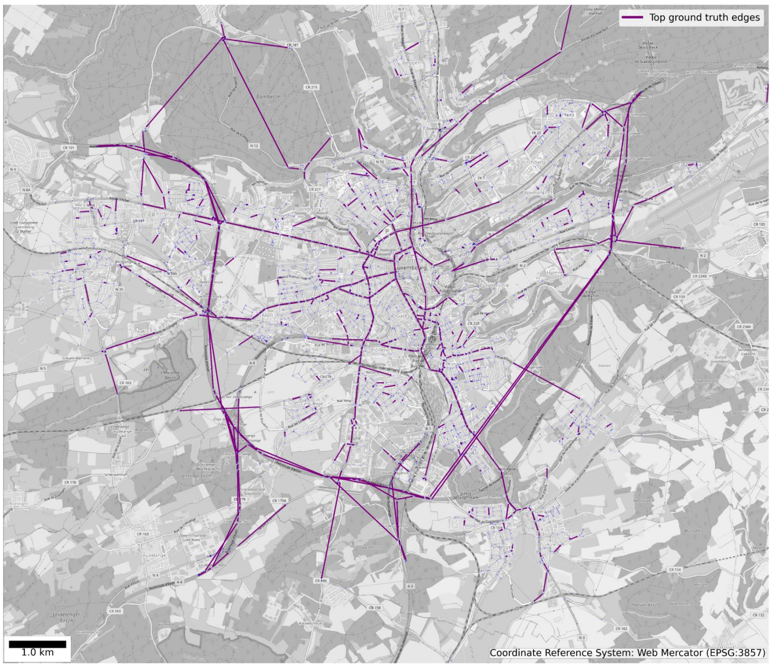



Figure 4.4: Performance Metrics for Same-City Evaluation on MoST Dataset

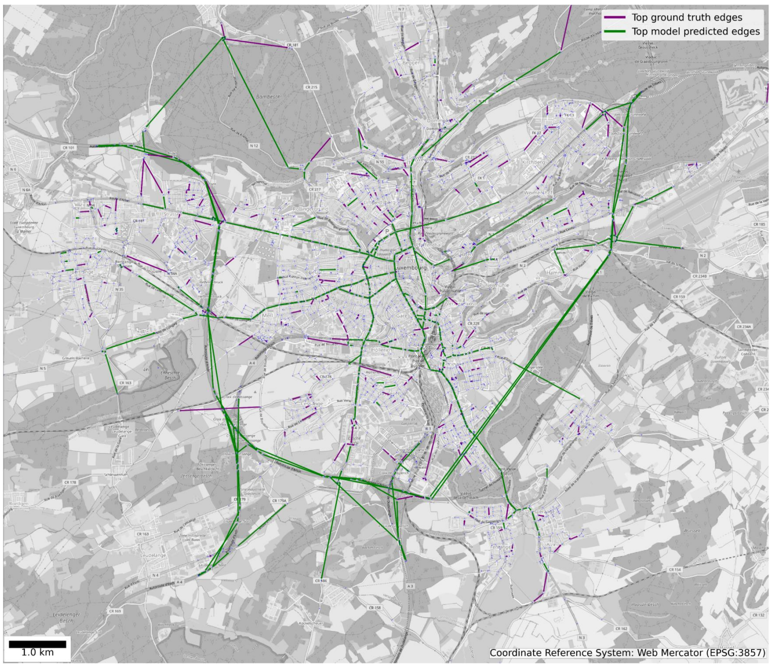
The following observations were made:

- The baseline configuration (MMS) had a precision of 52.47% and a PRMSE of 12.14%.
- Adding functional features (MMSF) increased precision by 18.13%, highlighting the importance of dynamic traffic metrics.
- The full-feature configuration (MMSFP) achieved a precision of 71.10%, indicating a 35.42% improvement over the baseline.
- Advanced preprocessing (MMSFPET) further improved precision to 73.00%, with a modest reduction in PRMSE.

Similar to LuST scenario, the comparison of the top edges in ground truth versus best model predicted results are shown in Figure 4.6 and zoomed in Figure 4.7 for clarity.

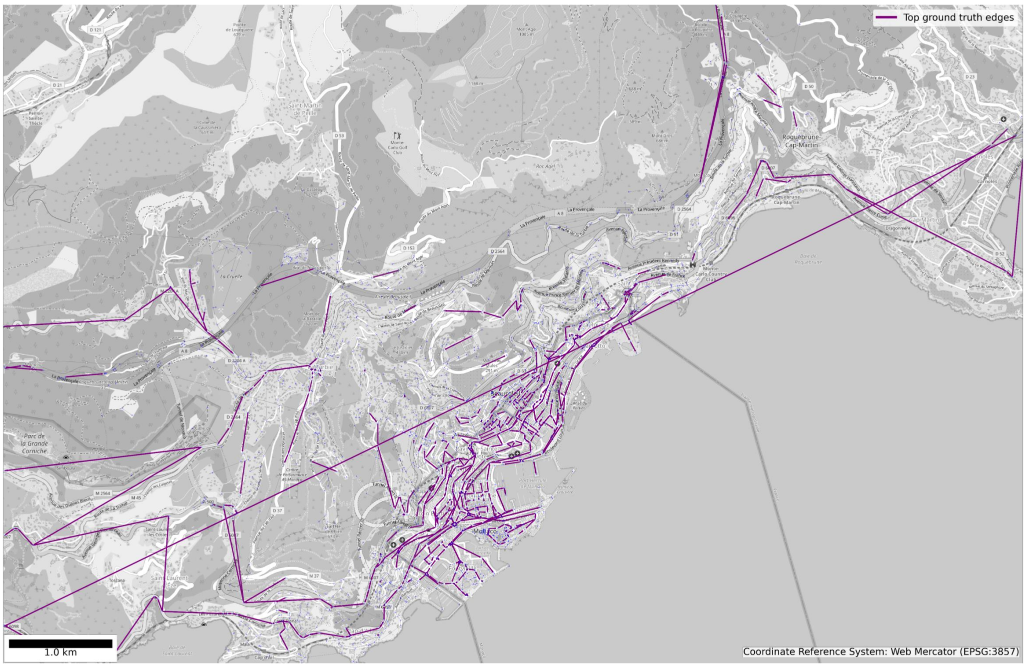



(a) Ground Truth Top Ranked Edges

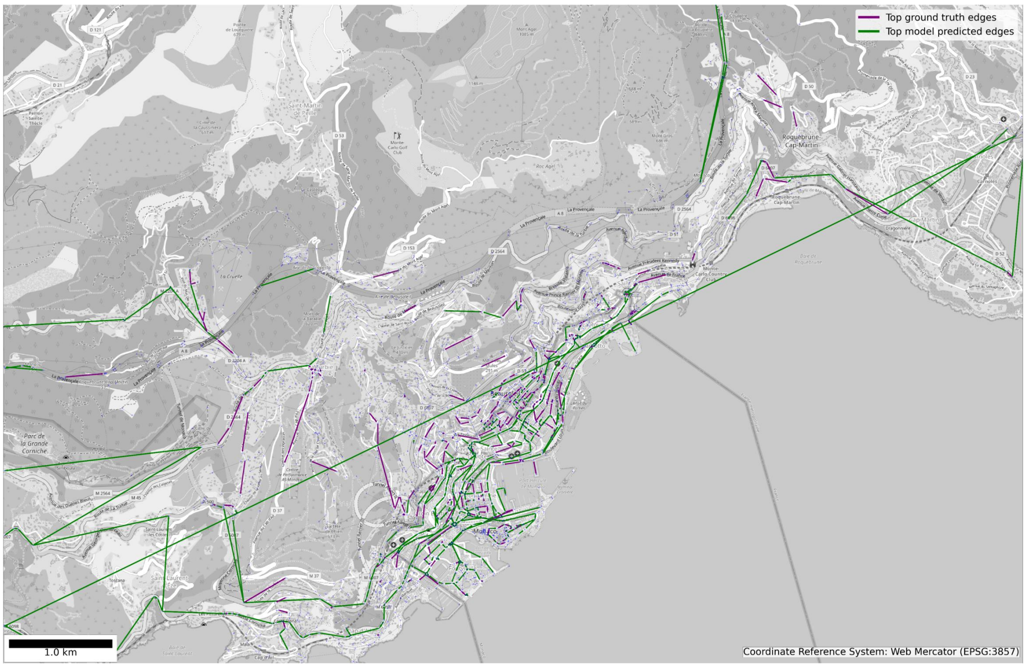



(b) Best Model Predicted Top Ranked Edges

Figure 4.5: Comparison of Ground Truth and Best Model Predicted Top Ranked Edges on LuST Dataset.




(a) Ground Truth Top Ranked Edges




(b) Best Model Predicted Top Ranked Edges

Figure 4.6: Comparison of Ground Truth and Best Model Predicted Top Ranked Edges on MoST Dataset.


(a) Ground Truth Top Ranked Edges


(b) Best Model Predicted Top Ranked Edges

Figure 4.7: Comparison of Ground Truth and Best Model Predicted Top Ranked Edges on MoST Dataset (Zoomed in for clarity).

### 4.4.2 Cross-City Generalization

To further evaluate the scalability and robustness of SMaL-CLIP, cross-city generalization experiments were conducted. These tests assess the framework's ability to predict link criticality in a city on which it was not trained, an important consideration for applications in cities with limited data availability.

As shown in Table 4.7, the model trained on Luxembourg and tested on Monaco (LMSFP) achieved a precision of 70.70%, demonstrating its ability to generalize across different network structures and traffic conditions. Similarly, training on Monaco and testing on Luxembourg (MLSFP) resulted in a precision of 66.61%, a modest drop reflecting the larger and more complex structure of Luxembourg's network. These results underscore the framework's adaptability compared to traditional approaches, which would typically require recalculation of centrality metrics for each new network, limiting portability.

Table 4.7: Performance Metrics for Cross-City Generalization

| **Configuration** | **Precision (%)** | **PRMSE (%)** |
|---|---|---|
| LMSFP | 70.70 | 6.73 |
| MLSFP | 66.61 | 9.73 |

### 4.4.3 Model Performance Across Configurations

The performances of the top models (Random Forest, Gradient Boosting, Decision Tree) were analyzed across configurations. Table 4.8 shows the consistent improvement in precision and PRMSE.

Table 4.8: Improvement in Model Performance Across Configurations

| **Model** | **LLS / MMS** | **LLSFPET / MMSFPET** | **Precision (%)** | **PRMSE (%)** |
|---|---|---|---|---|
| Random Forest | 61.80 / 50.62 | 68.57 / **72.75** | +6.77 / +22.13 | -0.51 / -2.39 |
| Gradient Boosting | **62.14** / 47.06 | **72.50** / 70.00 | +10.36 / +22.94 | -0.46 / -2.24 |
| Decision Tree | 60.89 / **52.47** | 70.00 / 68.36 | +9.11 / +15.89 | -0.25 / -1.96 |

The consistent performance gains across configurations, particularly when functional and dynamic features are incorporated, emphasize the advantage of integrating machine learning into criticality prediction. Traditional centrality measures, while effective for static network analysis, lack the ability to adapt to evolving traffic conditions. This adaptability is crucial in smart city contexts where real-time responses to network disruptions are needed.

## 4.5 Discussion

This section provides an in-depth discussion of the results, focusing on the framework's performance across different configurations, the impact of features and preprocessing, model-specific insights, and the broader implications for urban traffic management and future research. The discussion also highlights how SMaL-CLIP advances the field by introducing machine learning into critical link prediction, contrasting it with traditional approaches that rely on static indices and heuristic rules. It finally explains the limitations of the approach and future directions.

### 4.5.1 Single-City Scenarios: Training on Limited Data

In the single-city evaluations for Luxembourg (LuST) and Monaco (MoST), the framework demonstrated strong predictive performance despite training on only 20% of the network's links. This design significantly reduces data collection and computational overhead, making the framework suitable for large urban networks where full network simulations would be impractical. As previously discussed, the choice of using 20% of the data for training is not a fixed requirement but rather a practical configuration identified through empirical testing. The core objective is to demonstrate that the model can generalize effectively even when trained on a limited portion of the network. While other thresholds—both lower and higher—were explored, 20% yielded sufficiently accurate and stable predictions across scenarios. This proportion may be further refined through expert knowledge or adaptive learning strategies in future work.

The inclusion of proposed dynamic features (SFI, CIS, SIS) contributed to substantial improvements in precision — up to 7.86% in LuST and 18.13% in MoST — illustrating their importance in capturing temporal traffic dynamics. Advanced preprocessing steps, including feature selection and hyperparameter tuning, provided further gains, reducing PRMSE by up to 0.52% in LuST and 2.39% in MoST. This confirms that combining informative features with effective preprocessing produces a highly scalable, adaptable solution for link criticality prediction.

These results demonstrate how SMaL-CLIP leverages both structural and dynamic traffic features to predict link criticality. Compared to traditional approaches based on static centrality measures (e.g., betweenness centrality), which typically require full network analysis and cannot easily adapt to changing traffic patterns, our framework dynamically captures both network topology and real-time congestion impacts, making it more adaptable to evolving urban conditions.

### 4.5.2 Cross-City Generalization: Full Data Training and Testing

The cross-city evaluations assessed how well models trained on one urban network generalize to a network with different structural and dynamic characteristics. This is an important test for practical deployment, as it reflects the real-world scenario where traffic data may not be available for all cities.

Despite the challenges posed by cross-city variation, the framework maintained promising precision — around 70% when transferring from Luxembourg to Monaco, and approximately 66% in the reverse case. The slightly lower precision when transferring from Monaco to Luxembourg reflects the denser and more homogeneous structure of Monaco's network, which is less representative of larger cities. These results illustrate both the potential and limitations of cross-city generalization, highlighting the importance of capturing transferable features.

The slight increase in PRMSE during cross-city evaluations highlights the inherent difficulty of adapting to previously unseen network topologies and traffic conditions. This performance gap could be reduced through domain adaptation techniques, such as transfer learning or feature space alignment, which will be explored in future work.

### 4.5.3 Contributions of Features and Preprocessing

The experimental results clearly demonstrate the importance of enriching the feature set beyond simple structural attributes. While structural features alone provided reasonable predictive power, they lacked the ability to capture real-time congestion dynamics and sequential flow disruptions. Adding functional features (traffic density, speed, and occupancy) provided essential real-time information, substantially improving predictive accuracy.

The proposed dynamic features — SFI, CIS, and SIS — further enhanced the model performance even if slightly so. By capturing the sequential movement patterns of vehicles, these indices provided a more complete representation of how individual links contribute to network-wide functionality. This is particularly valuable when predicting the impact of link closures or failures in networks where route choices depend heavily on temporally evolving congestion patterns.

Preprocessing steps, particularly feature selection and hyperparameter tuning, also contributed to improved performance. Feature selection helped remove irrelevant or redundant features, improving model generalizability and interpretability. Hyperparameter tuning, particularly for ensemble models like Random Forest and Gradient Boosting, optimized predictive performance by balancing bias and variance.

### 4.5.4 Comparison with Traditional Methods

Traditional approaches to critical link analysis often rely on static network centrality measures (e.g., betweenness centrality) or functional indices derived from simple traffic volume or congestion counts. These methods, while useful for basic network analysis, have inherent limitations when applied to dynamically evolving urban traffic systems. They assume that network topology alone, or static traffic metrics, are sufficient to determine criticality — overlooking the complex, non-linear interactions between road links and traffic patterns.

SMaL-CLIP overcomes these limitations by integrating diverse feature types, including dynamic traffic attributes and sequential movement patterns derived from trajectory data. This data-driven approach allows the framework to automatically learn the underlying relationships between network structure, traffic behavior, and criticality, capturing non-linear dependencies that traditional methods miss. Furthermore, while some works did work on combining a set of different features, our approach allows adding and adapting to new indices and features at any point seamlessly.

Additionally, traditional approaches often require recalculating indices across the entire network each time conditions change (e.g., roadworks, accidents), which can be computationally prohibitive in large networks. In contrast, SMaL-CLIP, once trained, can rapidly predict the criticality of all links based on current traffic conditions, significantly improving scalability and responsiveness in real-time traffic management settings.

Table 4.9 summarizes the key differences between traditional criticality analysis methods and SMaL-CLIP.

Table 4.9: Comparison Between Traditional and Machine Learning-Based Approaches

| **Aspect** | **Traditional Methods** | **SMaL-CLIP** |
|---|---|---|
| Data Source | Static topology | Dynamic + Structural + Sequential |
| Traffic Dynamics | Limited | Fully captured |
| Scalability | Computationally expensive for large networks | Efficient training with 20% data |
| Adaptability | Requires manual recalculation and calibration after changes | Automatically adapts to new data |
| Cross-City Generalization | Limited | Demonstrated transferability |

### 4.5.5 Broader Implications for Urban Traffic Management and Smart Cities

The proposed framework aligns well with the evolving needs of smart cities and data-driven traffic management systems. Its ability to identify critical links using limited training data makes it suitable for cities with incomplete or sparse traffic data, while its flexibility allows integration with real-time monitoring systems for dynamic traffic management.

Urban planners can leverage the framework to prioritize road maintenance efforts, focusing resources on high-criticality links whose disruption would cause widespread network inefficiencies. Traffic control centers can use the predicted criticality scores to design dynamic rerouting strategies, preemptively diverting traffic away from vulnerable links to avoid congestion. These applications complement previously proposed congestion mitigation strategies, where proactive identification of critical links helped inform incident response and traffic rebalancing efforts (Bachir et al., 2023).

### 4.5.6 Limitations and Future Directions

As a preliminary contribution, this study establishes a scalable machine learning framework for critical link prediction. However, several limitations should be acknowledged.

First, the framework currently relies on data from SUMO simulations, which, while realistic, may not capture all complexities of real-world traffic systems, including unexpected human behavior or irregularities in infrastructure. Future work will focus on validating the framework using real-world sensor data, including connected vehicle data and IoT-enabled infrastructure, to bridge the gap between simulation-based validation and real-world deployment.

Second, while the achieved precision is promising — particularly given the limited training data used — it can still be further improved. Enhancing the framework's feature engineering process by incorporating additional dynamic and contextual features, such as topographical nature, weather conditions, special events, or traffic signal timing, may improve its ability to detect highly critical links more accurately.

While the current feature set is extensive, incorporating structural, functional, and newly proposed dynamic indices, it can be further enriched to improve model robustness and cross-city generalization. Integrating topographic and contextual features (e.g., elevation, slope, road curvature, surrounding land use) may offer additional explanatory power, especially in cities with pronounced geographical constraints. Similarly, fine-grained temporal features that reflect short-term fluctuations and spatial features capturing localized congestion dynamics could enhance predictive accuracy. On the modeling side, future directions include the use of spatio-temporal deep learning architectures and graph-based methods such as Graph Neural Networks (GNNs), which are well-suited to capturing complex dependencies within road networks.

Beyond technical enhancements, future work will emphasize interdisciplinary collaboration. Bringing together transportation engineers, data scientists, and urban planners will ensure that the framework's outputs are both technically robust and practically interpretable. Such collaboration can also lead to hybrid approaches that combine machine learning predictions with expert knowledge and traditional traffic engineering heuristics, creating a more balanced and trusted decision-support tool.

Finally, further research will investigate transfer learning strategies, allowing models trained on data-rich cities to support predictions in smaller cities with limited available data. Developing techniques to automatically detect and adapt to the unique characteristics of each city's network topology and traffic patterns will enhance the framework's portability across diverse urban environments.

## 4.6 Conclusion and Future Directions

This study introduces a scalable and efficient machine learning framework for predicting critical links in urban road networks. By integrating structural, functional, and newly proposed features, the framework achieves significant improvements in precision and error reduction across diverse configurations. The results demonstrate the framework's scalability, as it achieved high precision, approximately 72% for LuST and 73% for MoST, while training on only 20% of the data in single-city scenarios. This highlights its applicability to large-scale networks where data availability may be limited. Cross-city evaluations validated the framework's adaptability, with robust performance of around 70% precision when trained on LuST and tested on MoST, and approximately 66% precision when trained on MoST and tested on LuST. These findings underscore the framework's ability to generalize across cities with different network structures and traffic patterns.

The study further reveals the effectiveness of Random Forest and Gradient Boosting as top-performing models, consistently achieving the highest precisions and lowest error rates. The inclusion of functional and proposed features significantly enhanced the framework's ability to predict link criticality, demonstrating the importance of incorporating dynamic and network-specific attributes. Advanced preprocessing steps, including feature selection and hyperparameter tuning, contributed to further performance gains, particularly in reducing error rates such as PRMSE. These contributions collectively highlight the potential of SMaL-CLIP to balance computational efficiency and predictive accuracy, offering valuable insights for urban traffic management and strategic planning.

This work serves as a foundational contribution, introducing machine learning-based critical link prediction to the field and demonstrating the scalability benefits offered by data-driven approaches. While the initial accuracy levels are promising, they reflect the preliminary nature of this work, and we fully expect future improvements as larger and richer datasets become available and more advanced features are explored.

The framework's potential integration into smart city systems also highlights how machine learning can play a transformative role in the proactive management of urban traffic networks. By continuously learning from live sensor data and historical patterns, the framework could evolve into a core component of adaptive traffic control systems. Such systems would not only detect critical links but also anticipate emerging bottlenecks, enabling city planners and traffic controllers to take preemptive action.

While the proposed framework demonstrates strong performance, several opportunities exist for further exploration and improvement. Cross-city evaluations highlighted performance disparities, likely due to variations in network topology and traffic dynamics. Future research could address these challenges by incorporating domain adaptation techniques or transfer learning to enhance the framework's ability to generalize across cities with diverse characteristics. Another promising direction involves integrating the framework with real-time traffic monitoring systems, enabling dynamic prediction of critical links and improving its applicability for adaptive traffic management.

Collaborations between data scientists, transportation researchers, and policymakers will be critical in refining the framework for real-world use. Such interdisciplinary efforts can help ensure that the insights generated by machine learning models are interpretable, actionable, and aligned with policy and operational goals. Furthermore, collaborative efforts involving domain experts can support more rigorous calibration of model parameters, such as optimizing the training data threshold and refining the feature selection process. Such guidance could help adapt the framework to specific operational contexts or deployment constraints.

Additional advancements in feature engineering could further enhance the framework's generalization capabilities. Features that are invariant across networks, such as universal road usage patterns or flow consistency metrics, may mitigate the impact of structural differences between cities. Moreover, the strong performance of individual models like Random Forest and Gradient Boosting suggests that ensemble approaches combining multiple models could leverage their complementary strengths for even better results. Incorporating temporal dynamics into the framework, using techniques like recurrent neural networks (RNNs) or transformers, could also improve predictive accuracy, especially for real-time applications. Beyond urban road networks, the framework could be extended to other domains such as disaster response and logistics optimization, demonstrating its versatility in analyzing and managing complex networks.

The ability to achieve high performance with limited training data further highlights the scalability of the framework for large networks and datasets. This study establishes a robust foundation for future research in critical link prediction and emphasizes the potential for integrating machine learning into real-time urban traffic management systems. Continued advancements in feature design, model optimization, and cross-city generalization will further enhance the framework's utility, paving the way for smarter and more resilient transportation networks.

Overall, this work highlights the transformative potential of data-driven methods in critical infrastructure analysis, contributing to the ongoing shift from static, pre-defined indices to dynamic, adaptive, and predictive frameworks powered by machine learning.

Chapter 5

# Conclusion

Urban transportation systems are complex, interdependent, and increasingly data-rich—yet remain vulnerable to both everyday disruptions and large-scale failures. While cities are investing in smarter infrastructure, the ability to understand which road segments matter most under pressure remains a central challenge in designing resilient and adaptive mobility systems.

This thesis was motivated by that challenge. Specifically, it asked whether road segment criticality can be modeled not only through static heuristics or isolated simulation, but as a learnable property—emerging from a combination of structure, behavior, and disruption dynamics. The goal was to move beyond narrowly defined indicators and build a framework that integrates multiple perspectives into a unified, predictive approach.

At its core, this research explored how spatial data, movement patterns, simulation-based impact, and machine learning could be combined into a scalable and interpretable pipeline for criticality prediction. Rather than treating these components in isolation, the thesis proposes a structured methodology that fuses them through three major contributions: the PEMAP framework, the VeTraSPM mining model, and the SMaL-CLIP machine learning pipeline.

Chapter 2 introduced **PEMAP**, a four-phase conceptual framework for post-disruption traffic management. While this thesis focused primarily on the second phase—critical link evaluation—it also engaged with the first phase through a structured review of criticality definitions, indices, and modeling tools. Together, these phases provided the architectural foundation and integration lens that guided the design of subsequent methods.

Chapter 3 presented **VeTraSPM**, a sequential pattern mining model designed to extract meaningful behavioral patterns from vehicle trajectory data. It introduced three interpretable indices that captured the structural and directional characteristics of frequently used road sequences. These behavioral features formed a crucial layer in the criticality assessment pipeline and were further validated and extended through the SAMO refinement.

Chapter 4 introduced the most operational component of the thesis: **SMaL-CLIP**, a scalable machine learning framework that predicts link criticality using hybrid features derived from structure, behavior, and simulation. By using SUMO-based synthetic stress testing to generate ground-truth labels, and training on only a subset of the network, the model demonstrated strong generalization performance across two different cities. The SIS-based indices from VeTraSPM were included as optional features and consistently ranked among the top predictors.

Together, these contributions form a modular, data-integrated methodology for learning criticality in urban traffic networks. The results demonstrate that machine learning, informed by behavior and simulation, can provide predictive insight that is scalable, interpretable, and transferable across contexts.

While the thesis focuses on road traffic systems, its broader contribution lies in how it models complexity: not through a single lens, but through the integration of diverse perspectives into a learning-driven pipeline. It provides both conceptual structure and operational tools for those seeking to make urban mobility more resilient—especially in times of uncertainty and disruption.

## 5.1 Research Questions Revisited

The starting point of this thesis was a practical yet conceptually demanding question:

> Can road link criticality be modeled as a learnable property—rather than a fixed index—through machine learning using simulation-informed, structural, and trajectory-derived features?

This question was answered through the development of a modular pipeline that uses machine learning to predict road segment criticality based on simulation, structural attributes, and trajectory-derived behavior. The approach showed that criticality can **indeed** be treated as a learnable property. Simulation was used to quantify disruption impact, trajectory data provided behavioral indicators, and structural features captured network topology. These were combined into hybrid feature sets that enabled interpretable, scalable, and transferable predictions of critical links across different urban networks.

To explore this question in depth, four sub-questions were posed:

- What is criticality, and how can GIS, simulation, graph theory, and AI be effectively combined to assess it in urban road networks? How can this assessment be operationalized for resilient traffic management?
- What types of data-driven indices best capture road segment importance based on trajectory behavior and disruption impact?
- How can microscopic traffic simulation be used to generate informative and scalable datasets for predictive modeling and evaluation?
- Can machine learning models trained on partial data generalize to accurately identify critical links across entire networks or cities?

What follows is not a technical recap, but a reflection on how each of the core questions evolved through the research process—and how they were, at least partially, answered.

### On defining and operationalizing criticality through interdisciplinary integration

In the first research question asked, the inquiry was both conceptual and methodological, requiring the articulation of a clear definition of criticality and the integration of disciplinary approaches that have traditionally operated in isolation.

To address this, the thesis introduced the PEMAP framework in Chapter 2, which defines a four-phase strategy for post-event traffic management: index assessment and evaluation, simulation-based criticality calculation, rerouting decision modeling, and application through vehicular communication. This research engaged with both the first and second phases—defining criticality in the context of urban disruption, evaluating existing indices and modeling tools, and developing simulation-informed methods to quantify link-level impact.

Crucially, PEMAP does more than assemble disconnected components; it provides a principled structure for integrating them into a cohesive and adaptive methodology. Structural attributes derived from GIS and graph theory are complemented by behavior-sensitive patterns extracted from trajectory data and by impact signals generated through microscopic simulation. These elements are embedded within a unified pipeline in which each phase supports scalable, context-aware, and learnable prediction.

By embedding criticality assessment within this modular framework, the thesis demonstrates that cross-disciplinary integration is both feasible and essential. It enables targeted and interpretable modeling, supports operational scalability, and lays the foundation for intelligent traffic management systems capable of moving from static evaluation to real-time, disruption-aware response.

## On discovering more interpretable and behavior-sensitive indices

The second research question asked whether new, behaviorally grounded indicators of road segment importance could be extracted from vehicle trajectories, offering an alternative to conventional graph-based or flow-based metrics. Most prior work in critical link analysis emphasizes network topology or aggregated traffic volumes, which often fail to capture how roads are actually used in practice.

This led to the development of VeTraSPM—a sequential pattern mining model specifically designed for vehicular mobility data in constrained urban networks. Unlike generic sequence mining methods, VeTraSPM incorporates key properties of road networks, including one-way directionality, physical connectivity, and the possibility of repeated segment use. These domain-specific constraints allowed the model to extract movement sequences that reflect meaningful and consistent patterns of road usage.

From these sequences, the model derived three behavior-sensitive indices: the Frequency-based Movement Score (FqMS), the Confidence-based Movement Score (CMS), and the Sequential Impact Score (SIS). Each of these scores captures a different aspect of how road segments appear within frequently used routes, emphasizing their role in actual movement behavior rather than purely structural position.

These indices were used to characterize road segments based on how they function under observed usage conditions. They provided a complementary perspective to structural or simulated features and contributed valuable input to the learning-based prediction task. Importantly, they enabled the representation of link importance as an outcome of collective movement behavior—a perspective largely absent in prior work.

By grounding the assessment of road importance in trajectory-derived behavioral signals, the thesis expanded the feature space available for criticality prediction and demonstrated how vehicle movement data can reveal functional hierarchies that are invisible to static models.

## On using simulation not as a tool, but as a data engine

The third research question addressed a practical and foundational challenge in building predictive models for road criticality: the lack of labeled data. In real-world urban systems, disruptions are often rare, unstructured, and inconsistently documented, making it difficult to develop learning-based models that rely on ground truth for supervision.

To overcome this, the thesis explored whether microscopic traffic simulation could serve as a reliable and scalable mechanism for generating supervised labels. Using SUMO, traffic was simulated under baseline conditions and then re-simulated after systematically removing each individual link. The resulting difference in total trip time—before and after the removal—was used to quantify the criticality of each link as a measurable impact on network performance.

This process enabled the creation of consistent and high-resolution training datasets across two different urban environments: Luxembourg and Monaco. By controlling all parameters of the simulation, thousands of link-specific disruption scenarios were generated under the same behavioral assumptions, ensuring comparability and scalability.

The simulation-based supervision approach played two important roles. First, it provided the ground truth necessary to train and evaluate machine learning models in the absence of empirical failure data. Second, it served as a controlled test environment where feature design, model generalization, and performance boundaries could be systematically studied.

This thesis demonstrated that simulation can be used not only to validate traffic scenarios, but to drive model training itself. In doing so, it positioned simulation not as an auxiliary tool, but as a central component of a data-driven pipeline for learning road network criticality.

## On generalizing predictions across networks

The final research question investigated the extent to which a machine learning model trained on one part of a road network—or even on an entirely different city—could generalize in identifying critical links elsewhere. This question lay at the heart of the thesis's goal to enable scalable and transferable criticality assessment, particularly in cities with limited data or simulation resources.

This question was addressed through the development and evaluation of the SMaL-CLIP pipeline, introduced in Chapter 4. The model was trained using hybrid features derived from structural attributes, behavioral indices, and simulated disruption impacts. It was designed to learn from only a subset of the road network—typically 20

Results from both intra-city and cross-city experiments were consistent and promising.

Within each city, the model achieved high prediction accuracy even when trained on limited data. More significantly, models trained on one city (e.g., Luxembourg) performed well when applied to another (e.g., Monaco), with accuracy remaining in the 70–80

These findings demonstrate that link criticality, when represented through a rich, behavior- and structure-informed feature space, exhibits patterns that are sufficiently stable to support generalization. In other words, there are transferable characteristics of criticality that transcend local network idiosyncrasies.

This capability opens the door to practical deployment in real-world settings—especially in cities that lack detailed disruption histories or simulation expertise. It suggests that learned models from data-rich environments can inform criticality prediction in less-instrumented ones, enabling smarter planning and more resilient infrastructure decisions even under data-scarce conditions.

At the same time, the results raise important directions for future research: how to adapt models to account for changes in traffic patterns over time, how to transfer knowledge between cities with differing topologies or policies, and how to maintain robustness in the face of incomplete or noisy input data.

This thesis does not claim to resolve these challenges, but it provides a concrete and reproducible foundation on which they can be addressed.

## 5.2 Research Extensions and Outlook

While the proposed methodology answered its core questions and produced a complete pipeline for road segment criticality assessment, several natural extensions arise. These can be grouped into two categories: **Direct extensions**, which build incrementally on the core contributions of the thesis, enhancing their scope, automation, or operationalization; and **Latent extensions**, which open new directions based on the insights gained; spanning multimodal integration, real-time systems, decision support, and urban resilience.

This categorization helps frame the dual ambition of this work: to consolidate a robust technical foundation and to serve as a launchpad for broader applications in intelligent transportation systems.

### Direct Extensions

#### Extending the PEMAP Framework

PEMAP was conceived as a four-phase framework for post-event traffic management, encompassing index assessment and evaluation, simulation-based criticality computation, rerouting decision modeling, and real-time application via vehicular networks. While this thesis concentrated on the first two phases the remaining phases remain essential for realizing the full operational vision of PEMAP.

Future research could develop a complete, end-to-end implementation of the framework, capable of responding to real-world disruptions in near real-time. The first phase—index selection and evaluation—could incorporate real-time monitoring systems and historical data analysis to dynamically update the set of relevant features based on context. The third phase—rerouting strategy design—may benefit from advances in agent-based modeling, network flow optimization, or cooperative decision-making to test and select adaptive diversion plans under varying disruption scenarios. Finally, the fourth phase—decision dissemination—raises important questions regarding communication latency, bandwidth constraints, and compliance behavior when rerouting recommendations are broadcast via Vehicular Ad-hoc Networks (VANETs) or digital infrastructure.

Completing the PEMAP cycle would transform the current modeling framework into a fully operational disruption-response pipeline—linking detection, prediction, planning, and communication into a cohesive system. This integrated approach could serve as a foundation for next-generation mobility platforms that combine resilience, scalability, and intelligent decision support in the face of uncertainty.

### Enriching the Feature Space with Contextual and Topographic Data

Although the proposed framework incorporated a rich and diverse set of structural, functional, and dynamic features — including novel indices derived from trajectory mining — the feature space remains open for expansion. Several critical dimensions remain underexplored, particularly those that reflect the geophysical and contextual realities of the road network.

Topographic attributes such as elevation, slope, curvature, and gradient can significantly influence traffic flow and link vulnerability, especially in cities with varied terrain. Likewise, contextual data — including land use, surrounding activity zones, school or hospital proximity, and environmental constraints — may add valuable predictive signals for both recurrent and event-based disruptions. While such features were not incorporated in the present study, their integration could improve both model granularity and realism.

Moreover, future work could investigate local spatial interactions, capturing neighborhood-level bottlenecks or cascading effects not visible at the level of isolated link attributes. Feature extraction techniques leveraging graph embeddings, spatial clustering, or kernel-based locality measures could provide a way to quantify such interactions.

These directions will require access to enhanced datasets, often combining GIS, remote sensing, and city infrastructure layers. Yet the potential payoff is considerable: a more nuanced, interpretable, and robust assessment of link criticality that reflects both the topology and the terrain in which cities operate.

### Incorporating Temporal Dynamics

The methods developed in this thesis are essentially static: they consider criticality as a function of a given state of the network. Yet, criticality is inherently dynamic. A link may be critical during peak hours and negligible overnight. Disruption effects may cascade differently depending on timing, duration, and accumulated pressure.

Future work could explore the modeling of criticality as a time-dependent function. This would involve tracking how indices or model predictions evolve over time, possibly using temporal graph models or spatiotemporal neural architectures. Such approaches could also support early-warning systems that detect rising vulnerability before a failure occurs.

### Enhancing Generalization and Transfer Across Cities

One of the most encouraging results in this thesis was the successful transfer of trained models between cities. With only a fraction of the network used for training, models were able to predict critical links in a different city with high accuracy. This suggests that patterns of criticality, when modeled through a hybrid feature space, are at least partially transferable.

In this context, an important design decision in the current framework was the use of only 20% of the links for training. This proportion was chosen empirically after experimenting with different thresholds, some as low as 8%, and observing model behavior. The goal was to strike a balance between training efficiency and predictive accuracy, demonstrating the model's ability to generalize from limited supervision. Future work could build on this by exploring adaptive or data-driven strategies for training data selection. These may be based on sampling diversity, network structure, or performance-based feedback to improve generalization across varied urban contexts.

Future research could formalize this generalization process through domain adaptation, meta-learning, or federated learning. These strategies would allow models to be tuned or co-trained across multiple cities without requiring central data sharing. In doing so, the methodology could support decision-making in smaller cities with limited infrastructure or in developing regions where simulation resources are scarce. Improving model adaptability will be key to ensuring equity in the deployment of predictive infrastructure tools.

## Latent Extensions

### Toward Multi-Modal Network Integration

The current focus on vehicular networks offers a clear and computationally tractable starting point, but real urban mobility is multi-modal. Buses, trams, cyclists, and pedestrians all interact with and shape road network usage. Moreover, disruptions in one mode often affect others. For example, a bus lane closure may trigger shifts in car traffic, or a metro failure may lead to overloading of surface roads.

Incorporating multi-modal data into the modeling framework would enhance the realism and utility of criticality assessments. This would require enriched data sources, new simulation capabilities, and feature representations that capture interdependencies across transport modes.

### From Synthetic Modeling to Real-Time Systems

This thesis relied on simulation as a controlled environment for model training and evaluation. While this approach proved effective, operational deployment requires a shift toward real-time, data-driven systems. Integrating live data sources—such as floating car data, traffic APIs, or crowdsourced alerts—introduces challenges of scale, noise, and volatility. Machine learning models must be adapted to handle streaming data, concept drift, and partial observations, possibly through online or continual learning frameworks.

Moreover, real-time systems must operate under computational constraints and prioritize fast, interpretable decisions. The models proposed here could serve as a starting point, but their architecture may need to be optimized for inference under real-time latency budgets. Data privacy, especially with vehicle trajectory data, also becomes a critical concern in operational contexts.

### Embedding Prediction Models into Decision Support Workflows

Criticality models are most valuable when integrated into real-world workflows—whether for infrastructure planning, emergency response, or daily traffic operations. For that integration to succeed, the models must be not only technically robust but practically usable.

A natural extension of this research is the development of interfaces and tools that translate model outputs into actionable insight. This could take the form of visual analytics dashboards for traffic operators, scenario comparison tools for planners, or interactive maps that highlight critical links under different disruption types. It may also involve participatory design processes with city stakeholders to align the modeling framework with policy priorities and institutional constraints. Bridging the gap between prediction and action ensures that these models can become part of the decision-making ecosystem, not just theoretical contributions.

## Outlook

The research presented in this thesis repositions criticality from a static index to a dynamic, learnable property—one shaped by structure, behavior, and disruption. It introduced new models to extract behavioral movement patterns, generate supervised labels through simulation, and predict link importance using machine learning. Together, these models support a modular, generalizable framework for anticipating how disruptions affect urban road networks.

Yet this thesis is not a final answer. It is a beginning—a blueprint for what it means to treat criticality as a signal that can be modeled, scaled, and adapted to new contexts. As urban systems grow more interconnected and data-driven, the ability to model link importance with precision and to act on those insights in real time becomes not only useful but necessary.

The contributions made here are meant to be built upon, extended, and critically examined. They offer tools for learning how cities break—not to predict failure in despair, but to prepare for it with intelligence. And in that preparation lies the foundation of resilience.

# Bibliography


Agrawal, R., & Srikant, R. (2000). Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases VLDB*, *1215*.

Ahmed, M. A., Kays, H. M. I., & Sadri, A. M. (2023). Centrality-based lane interventions in road networks for improved level of service: The case of downtown Boise, Idaho. *Applied Network Science*, *8*, 1–15. https://doi.org/https://doi.org/10.1007/s41109-023-00532-z

Ahmed, S., & Dey, K. (2020). Resilience modeling concepts in transportation systems a comprehensive review based on mode, and modeling techniques. *Journal of Infrastructure Preservation and Resilience*, *1*.

Akbarzadeh, M., Memarmontazerin, S., Derrible, S., & Salehi, F. (2019). The role of travel demand and network centrality on the connectivity and resilience of an urban street system. *Transportation*, *46*, 1–15. https://doi.org/https://doi.org/10.1007/s11116-017-9814-y

Akbarzadeh, M., Salehi, F., & Aghababaei Samani, K. (2018). Detecting critical links of urban networks using cluster detection methods. *Physica A: Statistical Mechanics and its Applications*, *515*. https://doi.org/10.1016/j.physa.2018.09.170

Asgharian Rezaei, A., Munoz, J., Jalili, M., & Khayyam, H. (2023). A machine learning-based approach for vital node identification in complex networks. *Expert Systems with Applications*, *214*, 119086. https://doi.org/https://doi.org/10.1016/j.eswa.2022.119086

Ayres, J., Flannick, J., Gehrke, J., & Yiu, T. (2002). Sequential pattern mining using a bitmap representation, 429–435. https://doi.org/10.1145/775107.775109

Bachir, N., Harb, H., Zaki, C., & Billen, R. (2024). Samo: A sequential pattern mining model for evaluating road criticality in urban traffic networks. *IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, 1–5. https://doi.org/https://doi.org/10.1109/VTC2024-Fall63153.2024.10757675

Bachir, N., Harb, H., Zaki, C., Nabaa, M., Nys, G.-A., & Billen, R. (2023). PEMAP: An intelligence-based framework for post-event management of transportation systems. *Computers and Electrical Engineering*, *110*, 108856. https://doi.org/https://doi.org/10.1016/j.compeleceng.2023.108856

Bachir, N., Zaki, C., Harb, H., & Billen, R. (2025a). Scalable machine learning framework for predicting critical links in urban networks. *Journal of Innovation & Knowledge*, *10*(3), 100715. https://doi.org/https://doi.org/10.1016/j.jik.2025.100715

Bachir, N., Zaki, C., Harb, H., & Billen, R. (2025b). VeTraSPM: Novel vehicle trajectory data sequential pattern mining algorithm for link criticality analysis. *Vehicular Communications*, *51*, 100869. https://doi.org/https://doi.org/10.1016/j.vehcom.2024.100869

Cao, J., Spulbar, C., Eti, S., Horobet, A., Yüksel, S., & Dinçer, H. (2025). Innovative approaches to green digital twin technologies of sustainable smart cities using a novel hybrid decision-making system. *Journal of Innovation & Knowledge*, *10*(1), 100651. https://doi.org/https://doi.org/10.1016/j.jik.2025.100651

Codecá, L., Frank, R., Faye, S., & Engel, T. (2017). Luxembourg SUMO Traffic (LuST) Scenario: Traffic demand evaluation. *IEEE Intelligent Transportation Systems Magazine*, *9*(2), 52–63. https://doi.org/https://doi.org/10.1109/MITS.2017.2666585

Codecá, L., & Härri, J. (2018). Monaco SUMO Traffic (MoST) Scenario: A 3D mobility scenario for cooperative ITS. *SUMO 2018 - Simulating Autonomous and Intermodal Transport Systems*, *2*, 43–55. https://doi.org/https://doi.org/10.29007/1zt5

Desjardins, C., & Chaib-draa, B. (2011). Cooperative adaptive cruise control: A reinforcement learning approach. *IEEE Transactions on Intelligent Transportation Systems*, *12*(4), 1248–1260. https://doi.org/https://doi.org/10.1109/TITS.2011.2157145

El Rashidy, R., & Grant-Muller, S. (2014). An assessment method for highway network vulnerability. *Journal of Transport Geography*, *34*, 34–43. https://doi.org/https://doi.org/10.1016/j.jtrangeo.2013.10.017

Elsafdi, O. H. (2020). *Development of inherent and dynamic resilience in traffic networks: Microscopic simulation, dynamic stochastic assignment and Bayesian decision models* [Doctoral dissertation, Carleton University]. https://doi.org/https://doi.org/10.22215/etd/2020-14031

Feng, Y., Wang, H., Chang, C., & Lu, H. (2022). Intrinsic correlation with betweenness centrality and distribution of shortest paths. *Mathematics*, *10*(14), 2521. https://doi.org/https://doi.org/10.3390/math10142521

Fournier Viger, P. (2014). Fast vertical sequential pattern mining using co-occurrence information.

Fournier Viger, P., Gomariz, A., Gueniche, T., Mwamikazi, E., & Thomas, R. (2013). Tks: Efficient mining of top-k sequential patterns, 109–120. https://doi.org/10.1007/978-3-642-53914-5_10

Fournier Viger, P., Wu, C.-W., Gomariz, A., & Tseng, V. (2014). Vmsp: Efficient vertical mining of maximal sequential patterns. https://doi.org/10.1007/978-3-319-06483-3_8

Garofalakis, M., Rastogi, R., & Shim, K. (2002). Mining sequential patterns with regular expression constraints. *Knowledge and Data Engineering, IEEE Transactions on*, *14*, 530–552. https://doi.org/10.1109/TKDE.2002.1000341

Gauthier, P., Furno, A., & El Faouzi, N.-E. (2018). Road network resilience: How to identify critical links subject to day-to-day disruptions. *Transportation Research Record: Journal of the Transportation Research Board*, *2672*, 1–15. https://doi.org/https://doi.org/10.1177/0361198118792115

Gomariz, A., Campos, M., Marín, R., & Goethals, B. (2013). Clasp: An efficient algorithm for mining frequent closed sequences, 50–61. https://doi.org/10.1007/978-3-642-37453-1_5

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M. (2000). Freespan: Frequent pattern-projected sequential pattern mining. *Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *6*, 355–359. https://doi.org/10.1145/347090.347167

Han, J., Pei, J., & Yan, X.-F. (2004). From sequential pattern mining to structured pattern mining: A pattern-growth approach. *Journal of Computer Science and Technology*, *19*, 257–279. https://doi.org/10.1007/BF02944897

Henke, I., Troiani, G., & Pagliara, F. (2024). An analysis of the vulnerability of road networks in response to disruption events through accessibility indicators specification. *Transportation Planning and Technology*, *47*, 1–28. https://doi.org/https://doi.org/10.1080/03081060.2024.2329650

Henry, E., Furno, A., & El Faouzi, N.-E. (2019). A Graph-based Approach with Simulated Traffic Dynamics for the Analysis of Transportation Resilience in Smart Cities [TRB 2019, 98th Annual Meeting Transportation Research Board, Washigton, D.C., ETATS-UNIS, 13-/01/2019 - 17/01/2019]. *TRB 2019, 98th Annual Meeting Transportation Research Board*, 21p. https://hal.archives-ouvertes.fr/hal-03258248

Hu, S., Liang, Q., Qian, H., Weng, J., Zhou, W., & Lin, P. (2020). Frequent-pattern growth algorithm based association rule mining method of public transport travel stability. *International Journal of Sustainable Transportation*, *15*. https://doi.org/10.1080/15568318.2020.1827318

Ibrahim, R., & Shafiq, M. O. (2019). Detecting taxi movements using random swap clustering and sequential pattern mining. *Journal of Big Data*, *6*(1), 39. https://doi.org/10.1186/s40537-019-0203-6

Javadpour, A., AliPour, F. S., Sangaiah, A. K., Zhang, W., Ja'far, F., & Singh, A. (2023). An IoE blockchain-based network knowledge management model for resilient disaster frameworks. *Journal of Innovation & Knowledge*, *8*(3), 100400. https://doi.org/https://doi.org/10.1016/j.jik.2023.100400

Jayapal, C. (2022). Realtime congestion avoidance using vanet.

Jenelius, E. (2010). Redundancy importance: Links as rerouting alternatives during road network disruptions. *Procedia Engineering*, *3*, 129–137. https://doi.org/https://doi.org/10.1016/j.proeng.2010.07.013

Jenelius, E., Petersen, T., & Mattsson, L.-G. (2006). Importance and exposure in road network vulnerability analysis. *Transportation Research Part A: Policy and Practice*, *40*, 537–560. https://doi.org/https://doi.org/10.1016/j.tra.2005.11.003

Kemmar, A., Lebbah, Y., Loudni, S., Boizumault, P., & Charnois, T. (2017). Prefix-projection global constraint and top-k approach for sequential pattern mining. *Constraints*, *22*. https://doi.org/10.1007/s10601-016-9252-z

Khan, M., Popa, I., Zeitouni, K., & Borcea, C. (n.d.). Proactive vehicle re-routing strategies for congestion avoidance. *IEEE 8th International Conference on Distributed Computing in Sensor Systems*. https://doi.org/https://doi.org/10.1109/DCOSS.2012.29

Knorr, F., Baselt, D., Schreckenberg, M., & Mauve, M. (2012). Reducing traffic jams via vanets. *Vehicular Technology, IEEE Transactions on*, *61*, 3490–3498. https://doi.org/10.1109/TVT.2012.2209690

Lee, Y.-H., Kim, Y.-C., & Seo, H. (2022). Selecting disaster waste transportation routes to reduce overlapping of transportation routes after floods. *Sustainability*, *14*, 2866. https://doi.org/https://doi.org/10.3390/su14052866

Li, F., Jia, H., Luo, Q., Li, Y., & Yang, L. (2020). Identification of critical links in a large-scale road network considering the traffic flow betweenness index. *PLOS ONE*, *15*, e0227474. https://doi.org/https://doi.org/10.1371/journal.pone.0227474

Liu, H., Tong, Y., Han, J., Zhang, P., Lu, X., & Xiong, H. (2022). Incorporating multi-source urban data for personalized and context-aware multi-modal transportation recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *34*(2), 723–735. https://doi.org/https://doi.org/10.1109/TKDE.2020.2985954

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865–873. https://doi.org/https://doi.org/10.1109/TITS.2014.2345663

Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short–term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, *54*, 187–197. https://doi.org/https://doi.org/10.1016/j.trc.2015.03.014

Masseglia, F., Poncelet, P., & Teisseire, M. (2003). Incremental mining of sequential patterns in large databases. *Data Knowl. Eng.*, *46*, 97–121.

Merah, A. F., Samarah, S., & Boukerche, A. (2012). Vehicular movement patterns: A prediction-based route discovery technique for vanets, 5291–5295. https://doi.org/10.1109/ICC.2012.6364141

Merah, A. F., Samarah, S., Boukerche, A., & Mammeri, A. (2013). A sequential patterns data mining approach towards vehicular route prediction in vanets. *Mobile Networks and Applications*, *18*, 788–802. https://doi.org/10.1007/s11036-013-0459-6

Moreira-Matias, L., Ferreira, C., Gama, J., Moreira, J., & Sousa, J. (2012). Bus bunching detection: A sequence mining approach. *CEUR Workshop Proceedings*, *960*.

Oza, K., & Kawade, D. (2015). Frequent sequential pattern mining with weighted regular expression and length constraint. *International Journal of Scientific Research*, *4*, 3–7.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., & Hsu, M.-C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *Knowledge and Data Engineering, IEEE Transactions on*, *16*, 1424–1440. https://doi.org/10.1109/TKDE.2004.77

Qi, W., Song, Q., Wang, X., & Guo, L. (2017). Trajectory data mining-based routing in dtn-enabled vehicular ad hoc networks. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2017.2768485

Redzuan, A., Zakaria, R., Anuar, A., Aminudin, E., & Yusof, N. (2022). Road network vulnerability based on diversion routes to reconnect disrupted road segments. *Sustainability*, *14*, 2244. https://doi.org/10.3390/su14042244

Rosca, C., Stancu, A., Neculaiu, C.-F., & Gortoescu, I.-A. (2024). Designing and implementing a public urban transport scheduling system based on artificial intelligence for smart cities. *Applied Sciences*, *14*(19), 8861. https://doi.org/https://doi.org/10.3390/app14198861

Santamaria, A. F., Tropea, M., Fazio, P., & De Rango, F. (2018). Managing emergency situations in vanet through heterogeneous technologies cooperation. *Sensors*, *18*, 1461. https://doi.org/10.3390/s18051461

Scott, D., Novak, D., Aultman-Hall, L., & Guo, F. (2006). Network robustness index: A new method for identifying critical links and evaluating the performance of transportation networks. *Journal of Transport Geography*, *14*, 215–227. https://doi.org/https://doi.org/10.1016/j.jtrangeo.2005.10.003

Serdar, M. Z., Koç, M., & Al-Ghamdi, S. G. (2022). Urban transportation networks resilience: Indicators, disturbances, and assessment methods. *Sustainable Cities and Society*, *76*, 103452. https://doi.org/10.1016/j.scs.2021.103452

Shang, W.-L., Chen, Y., Li, X., Ochieng, W. Y., & Yang, Z. (2020). Resilience analysis of urban road networks based on adaptive signal controls: Day-to-day traffic dynamics with deep reinforcement learning. *Complex*, *2020*.

Shapouri, M., Fuller, J., Wolshon, B., & Herrera, N. (2023). Disruptions in megaregional network evacuations: Identifying and assessing critical links. *Transportation Research Record: Journal of the Transportation Research Board*. https://doi.org/https://doi.org/10.1177/03611981231160156

Sullivan, J. L., Novak, D. C., Aultman-Hall, L., & Scott, D. M. (2010). Identifying critical road segments and measuring system-wide robustness in transportation networks with isolating links: A link-based capacity-reduction approach. *Transportation Research Part A: Policy and Practice*, *44*(5), 323–336. https://doi.org/https://doi.org/10.1016/j.tra.2010.02.003

Thabtah, F. (2007). A review of associative classification mining. *Knowledge Eng. Review*, *22*, 37–65. https://doi.org/10.1017/S0269888907001026

Tian, Y., Liu, X., Li, Z., Tang, S., Shang, C., & Wei, L. (2021). Identification of critical links in urban road network considering cascading failures. *Mathematical Problems in Engineering*, *2021*, 1–15. https://doi.org/https://doi.org/10.1155/2021/9994347

Toulni, H., Nsiri, B., Boulmalf, M., Bakhouya, M., & Tayeb, S. (2014). An approach to avoid traffic congestion using vanet. *International Conference on Next Generation Networks and Services, NGNS*, 154–159. https://doi.org/10.1109/NGNS.2014.6990245

Tzvetkov, P., Yan, X., & Han, J. (2003). Tsp: Mining top-k closed sequential patterns. *Knowledge and Information Systems*, *7*. https://doi.org/10.1007/s10115-004-0175-4

Wang, J., & Han, J. (2004). Bide: Efficient mining of frequent closed sequences. *Data Engineering, 2004. Proceedings. 20th International Conference*, 79–90. https://doi.org/10.1109/ICDE.2004.1319986

Wang, M., Shan, H., Lu, R., Zhang, R., Shen, X., & Bai, F. (2015). Real-time path planning based on hybrid-vanet-enhanced transportation system. *Vehicular Technology, IEEE Transactions on*, *64*, 1664–1678. https://doi.org/10.1109/TVT.2014.2335201

Wang, S., Djahel, S., Zhang, Z., & Mcmanis, J. (2016). Next road rerouting: A multiagent system for mitigating unexpected urban traffic congestion. *IEEE Transactions on Intelligent Transportation Systems*, *17*, 1–12.

Wang, Y., Tian, Y., Yang, B., Wang, J., Hu, X., & An, S. (2023). Planning flexible bus service as an alternative to suspended bicycle-sharing service: A data-driven approach. *Journal of Advanced Transportation*, *2023*, 1–15. https://doi.org/10.1155/2023/3187654

Wang, Z., Huang, K., Massobrio, R., Bombelli, A., & Cats, O. (2024). Quantification and comparison of hierarchy in public transport networks. *Physica A: Statistical Mechanics* *and its Applications*, *634*, 129479. https://doi.org/https://doi.org/10.1016/j.physa.2023.129479

Xue, F., Shan, Z., Yan, L.-j., & Fan, C. (2016). A improved sequential pattern mining algorithm based on prefixspan, 1–4. https://doi.org/10.1109/WAC.2016.7583059

Yang, H., & An, S. (2020). Critical nodes identification in complex networks. *Symmetry*, *12*(1), 123. https://doi.org/https://doi.org/10.3390/sym12010123

Yu, W. (2019). Discovering frequent movement paths from taxi trajectory data using spatially embedded networks and association rules. *IEEE Transactions on Intelligent Transportation Systems*, *20*, 855–866. https://doi.org/10.1109/TITS.2018.2834573

Zaki, M. (2001). Zaki, m.j.: Spade: An efficient algorithm for mining frequent sequences. machine learning 42(1), 31-60. *Machine Learning*, *42*, 31–60. https://doi.org/10.1023/A:1007652502315

Zhang, H., & He, L. (2021). Data mining method of sequential patterns for vehicle trajectory prediction in vanet. *Wireless Personal Communications*, *117*, 1–13. https://doi.org/10.1007/s11277-020-07876-0

Zhang, S., Li, S., Zhai, C., & Xiao, J. (2024). An integrated seismic assessment method for urban buildings and roads. *International Journal of Disaster Risk Science*, *15*. https://doi.org/https://doi.org/10.1007/s13753-024-00600-7

Zhang, X., & Chen, M. (2024). Developing a practical highway criticality assessment framework based on network connectivity and redundancy. *Transportation*, 1–19. https://doi.org/https://doi.org/10.1007/s11116-024-10540-2

Zhong, X., & Liu, R. (2024). Identifying critical nodes in interdependent networks by GA-XGBoost. *Reliability Engineering & System Safety*. https://doi.org/https://doi.org/10.1016/j.ress.2024.110384

Zhou, T. (2018, June). *Deep learning models for route planning in road networks* [Doctoral dissertation]. https://doi.org/10.13140/RG.2.2.30527.56486

# Curriculum

Nourhan Bachir (1998) was born in Beirut, Lebanon. She obtained a BSc in Computer Science from the Lebanese University, an MSc in Information Systems and Data Intelligence from the Lebanese University, and is completing a PhD in Artificial Intelligence and Intelligent Systems at the University of Liège, Belgium, within the Geomatics Unit.

Her doctoral research focuses on criticality modeling in urban transportation networks, and her work intersects geographic information science, machine learning, behavioral mining, and simulation. Structured around a thesis by publication, her contributions include the design of PEMAP (a conceptual framework for post-event traffic management), the development of VeTraSPM and SAMO (trajectory-based pattern mining models), and the implementation of SMaL-CLIP (a scalable machine learning pipeline for predicting critical road segments).

During her PhD, she participated in multiple interdisciplinary research projects and was involved in teaching activities, supporting courses in artificial intelligence, computer science, and GIS-related applications. She also contributed to ongoing collaborations between the University of Liège and Lebanese institutions.

Parallel to her research, Nourhan held engineering and consulting roles in both academic and industrial environments. Her experience spans intelligent systems design, cloud infrastructure, data processing pipelines, and applied AI in transportation systems.

# List of Publications

## International Journal Papers (ISI)

N. Bachir, H. Harb, C. Zaki, and R. Billen (Dec. 1, 2023). “PEMAP: An intelligent framework for post-event traffic management using criticality modeling and vehicular networks.” In: Computers and Electrical Engineering, Vol. 111, p. 108856. ISSN: 0045-7906. doi: 10.1016/j.compeleceng.2023.108856

N. Bachir, H. Harb, C. Zaki, and R. Billen (Mar. 5, 2024). “Vehicle Trajectory Data Sequential Pattern Mining for Critical Link Analysis (VeTraSPM).” In: Vehicular Communications, Vol. 48, p. 100869. ISSN: 2214-2096. doi: 10.1016/j.vehcom.2024.100869

N. Bachir, H. Harb, C. Zaki, and R. Billen (May. 20, 2025). “A Scalable Machine Learning Framework for Predicting Critical Links in Urban Networks.” Journal of Innovation and Knowledge, Vol. 10, p. 100715. ISSN: 2444-569X. doi: 10.1016/j.jik.2025.100715

## International Refereed Conferences

N. Bachir, H. Harb, C. Zaki, and R. Billen (Oct. 2024). “SAMO: A Sequential Pattern Mining Model for Evaluating Road Criticality in Urban Traffic Networks.” In: Proceedings of the 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall). Hong Kong, China. IEEE, pp. 1–6. doi: 10.1109/VTC2024-Fall63153.2024.10757675

FACULTY OF SCIENCE
DEPARTMENT OF GEOGRAPHY
GEOMATICS UNIT
Quartier Agora, B5a - Liège, Belgium
B-4000 Liège