DTIP: A Scalable Pipeline for Traffic Congestion Detection Using Floating Car Data

Gil Silva

Faculdade de Ciência e Tecnologia

Universidade Fernando Pessoa

Supervisor: Prof. Christophe Soares

Co-Supervisor: Prof. José Torres

Thesis submitted to the degree of

Master of Science

2025

Resumo

O congestionamento do tráfego urbano continua sendo um obstáculo crítico para a mobilidade, segurança e sustentabilidade nas cidades modernas. Apresentamos o Distributed Traffic Intelligence Pipeline (DTIP), um sistema modular e interpretável projetado para estimar níveis de congestionamento a partir de Floating Car Data (FCD), ou seja, trajetórias de veículos baseadas em Global Positioning System (GPS), e para apoiar a validação de relatórios de perigos relacionados ao tráfego.

O framework proposto integra ferramentas de código aberto para processamento de dados, ”map-matching” e extração de características, culminando num modelo de aprendizagem supervisionado baseado em Extreme Gradient Boosting (XGBoost). O modelo foi treinado com dados de Vila Nova de Gaia, Portugal, e alcançou um F1-score ponderado acima de 97%, distinguindo com sucesso quatro classes de severidade de congestionamento. Para avaliar ainda mais a plausibilidade das suas previsões, uma camada de simulação qualitativa utilizando Simulation of Urban Mobility (SUMO) foi incorporada. Os resultados da simulação alinharam-se bem com as saídas do modelo na maioria dos cenários de teste, reforçando a validade comportamental das estimativas de congestionamento.

Desenvolvido com escalabilidade e implantação de baixa latência em mente, o DTIP oferece uma contribuição prática para o desenvolvimento de sistemas de monitoramento de tráfego urbano transparentes e eficientes. A sua natureza aberta e modular o torna adequado para adaptação a outras cidades ou para integração futura em infraestruturas de suporte a decisões em tempo real.

Palavras-Chave: Detecção de congestionamento de tráfego; Dados de Veículos Flutuantes (FCD); Análise de trajetórias GPS; Correspondência de mapas (Valhalla); Engenharia de características; Índice de Redução de Velocidade (SRI); XGBoost (Extreme Gradient Boosting); Sistemas de Transporte Inteligente (ITS); Processamento de dados escalável (Dask, Parquet); SUMO (Simulação de Mobilidade Urbana); Aprendizado de máquina interpretável; Monitoramento de mobilidade urbana; Validação de perigos.

Summary

Urban traffic congestion remains a critical obstacle to mobility, safety, and sustainability in modern cities. We present Distributed Traffic Intelligence Pipeline (DTIP), a modular and interpretable system designed to estimate congestion levels from Floating Car Data (FCD), i.e., Global Positioning System (GPS) based vehicle trajectories, and to support the validation of traffic-related hazard reports.

The proposed framework integrates open-source tools for data processing, map matching and feature extraction, culminating in a supervised learning model based on XGBoost. The model was trained on data from Vila Nova de Gaia, Portugal, and the system achieved a weighted F1-score above 97%, successfully distinguishing between four classes of congestion severity. To further assess the plausibility of its predictions, a qualitative simulation layer using Simulation of Urban Mobility (SUMO) was incorporated. The simulation results aligned well with the model's output in most test scenarios, reinforcing the behavioral validity of the congestion estimates.

Designed with scalability and low-latency deployment in mind, DTIP offers a practical contribution to the development of transparent and efficient urban traffic monitoring systems. Its open and modular nature makes it suitable for adaptation to other cities or future integration into real-time decision-support infrastructures.

Keywords: Traffic congestion detection; Floating Car Data (FCD); GPS trajectory analysis; Map-matching (Valhalla); Feature engineering; Speed Reduction Index (SRI); XGBoost (Extreme Gradient Boosting); Intelligent Transportation Systems (ITS); Scalable data processing (Dask, Parquet); SUMO (Simulation of Urban Mobility); Interpretable machine learning; Urban mobility monitoring; Hazard validation.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. José Torres and Prof. Christophe Soares whose invaluable guidance, expertise, and support were fundamental in shaping this work.

I am also deeply thankful to my family for their love and encouragement throughout this journey. Additionally, I extend my sincere appreciation to my colleagues at NDrive for their collaboration and support during the development of this thesis. Special thanks go to my company advisor, Helena Lopes, for her dedicated guidance and assistance.

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence

AIoT Artificial Intelligence of Things

API Application Programming Interface

CNN Convolutional Neural Network

DTIP Distributed Traffic Intelligence Pipeline

FCD Floating Car Data

GDPR General Data Protection Regulation

GPS Global Positioning System

HDOP Horizontal Dilution of Precision

IIoT Industrial Internet of Things

ITS Intelligent Transportation System

LSTM Long Short-Term Memory

ML Machine Learning

OSM OpenStreetMap

RNN Recurrent Neural Network

SRI Speed Reduction Index

SUMO Simulation of Urban Mobility

XGBoost Extreme Gradient Boosting

Chapter 1

Introduction

1.1 Background and Motivation

Urban traffic congestion represents a persistent and growing challenge for modern cities, negatively affecting mobility, public safety, economic productivity, and environmental sustainability. In metropolitan areas such as Vila Nova de Gaia, Portugal, traffic delays contribute to increased travel times, higher fuel consumption, and elevated greenhouse gas emissions. These factors not only inconvenience commuters but also impact emergency services, business logistics, and overall urban livability.

Air pollution resulting from vehicle emissions, particularly nitrogen dioxide (NO_2), is closely linked to respiratory and cardiovascular diseases, posing additional risks to public health (Levy et al., 2010; Faheem et al., 2024). Moreover, congestion can lead to a higher probability of road accidents and delayed emergency responses, intensifying strain on urban infrastructure. Addressing these issues requires the development of intelligent and scalable systems capable of real-time traffic monitoring, congestion detection, and incident validation.

1.2 Challenges in Traffic Monitoring

Traditional traffic monitoring infrastructures—including inductive loop detectors, roadside sensors, and CCTV (Closed Circuit Television) systems—have long been used in Intelligent Transportation System (ITS). Despite their accuracy, these solutions suffer from high deployment and maintenance costs, limited spatial coverage, and low adaptability to rapidly changing urban environments (Kong et al., 2016).

Modern platforms such as Google Maps and Waze partly overcome these limitations by combining car data with crowdsourced reports. Google leverages Graph Neural Networks for travel time estimation (Lau, 2020), while Waze integrates machine learning through TensorFlow Extended (TFX) to validate user-submitted incidents (Gal Moran

and Marcous, 2021). However, these proprietary systems are not open for external integration or research and often rely on crowdsourced data that may be inconsistent, biased, or sparse in regions with low smartphone penetration.

1.3 Automotive Navigation Systems and Floating Car Data

FCD collected from GPS-enabled vehicles or mobile navigation applications has emerged as a cost-effective and scalable alternative for urban traffic monitoring. Unlike traditional infrastructure-based systems, FCD captures movement directly from vehicles in operation, providing highly granular, real-time insight into road conditions and mobility trends (Kong et al., 2016).

Each data point in an FCD stream is known as a **probe**. A probe represents a single observation sampled from a moving vehicle, typically emitted every 1–2 seconds. It includes the vehicle's geographic coordinates (latitude and longitude), instantaneous speed, heading (i.e., direction of travel), timestamp, and occasionally metadata such as accuracy indicators or anonymized device IDs.

These probes are emitted continuously by mobile navigation applications or embedded vehicle devices, and are then transmitted to a backend service either in real time (online mode) or uploaded later in batches (offline mode). Each probe is thus a snapshot of the vehicle's state in time and space, and collectively they form the raw material for trajectory reconstruction, feature extraction, and congestion analysis.

Figure 1.1 illustrates the simplified life cycle of a probe in an automotive navigation system:

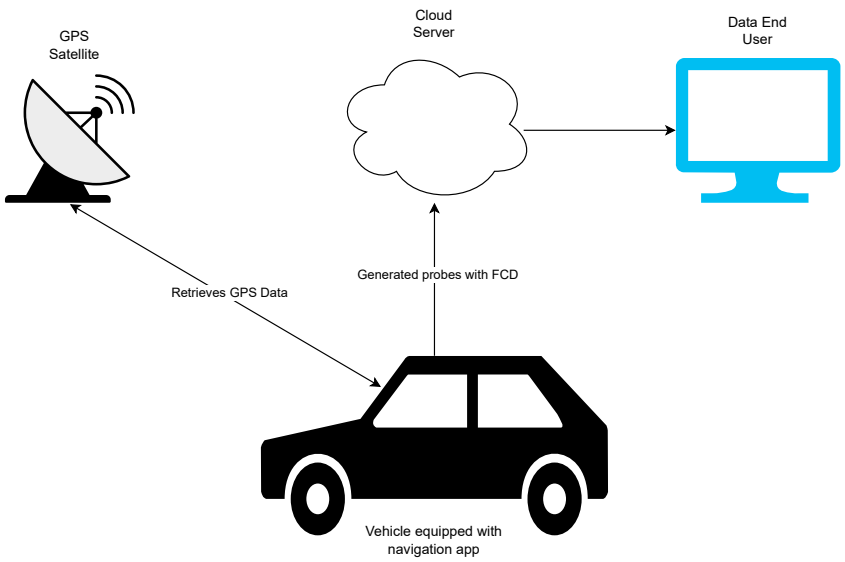

Figure 1.1: Probe Generation Life Cycle

- A vehicle (user) moves through the road network while a mobile device or embedded unit collects GPS data.
- The navigation application emits periodic probes—records containing the vehicle's current state.
- These probes are transmitted to a remote service that aggregates, stores, and processes the data for further analysis.

This architecture allows for large-scale data collection without deploying physical infrastructure on the roads. The collected FCD can then be transformed into traffic intelligence via map-matching, aggregation, and machine learning models—forming the backbone of the DTIP framework.

Challenges in Using FCD

Despite its advantages, operational use of FCD still faces several key challenges:

- **Volume and Velocity:** City-scale deployments produce tens to hundreds of millions of probes per month, requiring efficient storage, retrieval, and computation.
- **Spatial Noise:** GPS readings are affected by occlusion, reflections, and drift, especially in dense urban environments. Robust map-matching is essential.
- **Latency Requirements:** For real-time congestion detection or incident response, data must be ingested, cleaned, and analyzed with low latency.
- **Incomplete Network Knowledge:** If the underlying road network is outdated or inconsistent, map-matching and labeling may produce inaccurate results.
- **Lack of Ground Truth Labels:** Manual annotations for congestion levels are scarce, requiring proxy indicators—like speed reduction—to be used for model training.

Nevertheless, the flexibility, low-cost, and privacy-preserving nature of FCD make it an ideal foundation for scalable traffic monitoring frameworks such as DTIP.

1.4 Dataset and Methodology

This thesis uses a dataset collected by NDrive Navigation Systems during September and October 2020. The dataset comprises over 350 million GPS probes sampled at 1–2 Hz, providing high-resolution mobility traces across Portugal. Due to limited hardware resources, the dataset was filtered to include only data from the city of Vila Nova de Gaia.

This region was chosen because of its representative traffic density, urban structure, and local relevance, while also allowing for manageable computational demands.

Preprocessing includes trajectory cleaning, removal of invalid probes, and spatial correction (Huang et al., 2021) via map-matching using Valhalla (Valhalla Team, 2025). Features such as average segment speed, stop durations, and the Speed Reduction Index (SRI) (Erdelić et al., 2021) are derived to characterize traffic behavior. A Gradient Boosting approach is implemented to get trained models to classify congestion levels into four categories: none, light, moderate, and severe. Performance is evaluated using metrics such as F1-score, precision, and recall.

While the core validation in this thesis relies on real-world FCD and quantitative metrics, a complementary qualitative evaluation is also conducted using the Simulation of Urban Mobility (SUMO) traffic simulator (Alvarez Lopez et al., 2018). This simulation framework is used to recreate congestion scenarios based on DTIP's predictions, offering visual and dynamic feedback on the model's behavior. It is important to mention that SUMO is not part of the DTIP pipeline itself, but serves as an independent module for post hoc interpretability and exploratory validation.

1.5 Objectives, Contributions and Limitations

The central research question of this work is:

> *How can FCD and machine learning be combined to detect urban traffic congestion and validate hazard reports under real-world constraints such as data noise, infrastructure limitations, and computational cost?*

To address this question, the thesis pursues the following objectives:

1. Develop and validate a supervised machine learning model for congestion classification using FCD and handcrafted features.
2. Investigate the viability of using congestion estimates as supporting signals for hazard report validation.
3. Analyze the plausibility of predictions through visual simulation using SUMO in controlled scenarios.

Contributions

The main contributions of this thesis are:

- A scalable and modular pipeline for congestion classification based on interpretable machine learning and robust map-matching.

- A methodology for deriving congestion labels using the Speed Reduction Index (SRI) in the absence of ground truth.
- An auxiliary evaluation framework based on SUMO simulations to assess the realism and interpretability of model outputs.

Limitations

The scope of this study is subject to several limitations:

- **Temporal scope:** The dataset reflects conditions from 2020 and may not represent recent infrastructure changes.
- **Hazard scope:** Only congestion-causing incidents are validated; hazards not impacting flow are excluded.
- **Geographical scope:** The system is evaluated for Gaia only; generalization to other cities requires adaptation and retraining but is achievable due to the modular pipeline.
- **Data assumptions:** Labels are derived from SRI thresholds, which may not generalize across contexts.
- **Label availability:** Ground-truth congestion labels are not manually annotated, which necessitated the use of SRI-derived thresholds as a proxy. This introduces potential domain bias and limits interpretability across regions.

1.6 Document Structure

The remainder of this thesis is organized as follows:

- **Chapter 2 – State of the Art:** Reviews existing approaches to traffic congestion detection, comparing traditional sensor infrastructures, FCD, and crowdsourced reports. It also discusses the role of machine learning and justifies the architectural choices adopted in this thesis.
- **Chapter 3 – DTIP:** Presents the proposed Distributed Traffic Intelligence Pipeline, detailing each layer of the system—from data preprocessing and feature extraction to model training and prediction. Design choices, scalability mechanisms, and implementation details are also covered.
- **Chapter 4 – Evaluation:** Assesses the performance of the DTIP framework both quantitatively, using real-world FCD, and qualitatively, via visual simulations in the SUMO traffic simulator. Strengths, limitations, and generalizability are discussed.

- **Chapter 5 – Conclusion:** Summarizes the key findings, outlines the contributions of the work, and suggests future research directions and potential applications in real-world traffic monitoring systems.

Chapter 2

State of the Art on Traffic Congestion Detection

2.1 Introduction

The accurate and timely detection of traffic congestion is a cornerstone of modern ITS. Effective congestion monitoring supports urban mobility planning, improves road safety, reduces fuel consumption, and mitigates the environmental impact of transportation. In addition, congestion estimation plays a strategic role in the validation of reported traffic hazards, by helping to distinguish between truly disruptive events and false or redundant alerts — a key concern in crowdsourced platforms.

However, despite increasing volumes of urban mobility data, few datasets provide manually annotated congestion labels. As a result, most practical systems must rely on derived metrics—such as relative speed reduction or segment-level delay—as proxies for ground-truth congestion states. This creates a methodological gap between research and deployment.

This chapter reviews the current state of the art in traffic congestion detection, focusing on the data sources, methodological approaches, infrastructural considerations, and evaluation strategies that inform the design of the proposed DTIP framework.

2.2 Traffic Sources

2.2.1 Fixed-Sensor Infrastructure

Traditional ITS infrastructures rely on technologies such as inductive loop detectors, microwave sensors, and surveillance cameras to collect traffic metrics like flow, occupancy, and speed. These systems are known for their high precision and real-time responsiveness within their designated coverage zones. However, they present notable drawbacks in

terms of scalability, adaptability, and long-term operational cost.

A 2016 study conducted by researchers from the School of Software at Dalian University of Technology (Kong et al., 2016) examined these limitations in detail, underscoring how such rigid infrastructures struggle to adapt to the evolving spatial dynamics of modern cities. According to their findings, fixed-sensor networks often require substantial capital expenditure (CAPEX) for deployment and ongoing maintenance, and they offer limited flexibility when cities undergo structural transformations such as roadworks, temporary rerouting, or rapid urban expansion.

The same study also emphasized that although fixed sensors perform well when placed at critical bottlenecks—such as highway junctions or signalized intersections—they fall short as a scalable backbone for comprehensive traffic monitoring at the city scale. Their static nature makes them poorly suited for capturing unexpected disruptions, off-peak anomalies, or emerging patterns in less-monitored areas.

As a result, while fixed-sensor technologies remain valuable in specific use cases requiring high precision, their rigidity and cost constraints make them less attractive for modern, adaptive congestion detection systems.

2.2.2 Spatiotemporal Vehicle Data

FCD refers to spatiotemporal data collected from GPS-equipped vehicles and navigation devices. It offers a flexible and cost-effective alternative to fixed-sensor infrastructures, particularly in large and heterogeneous urban areas. Each probe typically includes timestamped location, speed, heading, and positioning accuracy, allowing the reconstruction of vehicle trajectories with fine spatial and temporal resolution.

A 2021 survey conducted by a consortium of Chinese researchers, including contributors from Southeast University and Tongji University, provides a comprehensive overview of the state of the art in FCD processing and its application in traffic modeling (Huang et al., 2021). Their review highlights key challenges in utilizing FCD, such as the need for robust map-matching, trajectory reconstruction, and the normalization of irregular sampling rates. These issues are particularly pronounced in dense urban environments, where GPS signals are prone to multipath interference and degradation due to building-induced occlusion.

Complementing this, a study published in the same year by Li et al. (Li et al., 2021), involving collaboration between several academic institutions and transport authorities, evaluated the use of ensemble machine learning models for traffic flow estimation using FCD. Their proposed multi-model framework—combining statistical filtering with supervised learning—demonstrated strong predictive performance even under conditions of sparse data coverage. This approach underscored the adaptability of FCD-based systems to areas with limited infrastructure investment or reduced sensor deployment.

Taken together, these contributions confirm that FCD represents a scalable and versatile foundation for urban traffic monitoring. Its ability to provide wide-area coverage with minimal deployment overhead makes it particularly suitable for modern, data-driven intelligent transportation systems.

However, leveraging FCD for operational traffic analytics presents a number of technical challenges:

- **GPS Noise and Drift:** In dense urban areas, signal occlusion and multipath effects result in noisy or misaligned coordinates. Accurate map-matching is therefore essential to align probes with the road network. Approaches based on Hidden Markov Models (HMMs), such as those implemented in the Valhalla engine, have proven effective in correcting trajectory drift.
- **High Data Volume:** A single vehicle reporting every second over a week can generate tens of thousands of probes. When scaled to a city-wide fleet, storage and processing become non-trivial. To mitigate this, modern pipelines employ compressed columnar formats like Parquet and parallel computing libraries like Dask to enable out-of-core processing.
- **Heterogeneous Sampling and Coverage Gaps:** FCD is collected opportunistically—depending on app usage or fleet movement—and thus exhibits strong temporal and spatial variability. Some regions may be oversampled during peak hours, while others remain underrepresented. This uneven distribution can introduce bias in congestion estimation unless properly normalized.

Despite these obstacles, FCD remains a uniquely powerful source for traffic analysis. Its scalability, low marginal cost, and minimal deployment requirements make it ideal for rapidly evolving urban contexts. Moreover, when pseudonymized and aggregated, FCD complies more readily with privacy regulations such as the GDPR than crowdsourced reports or video feeds.

While this work was always planned to use FCD as the cornerstone, the aforementioned reasons highlight its centrality in enabling reliable congestion estimation. For these reasons, FCD is adopted as the cornerstone of the DTIP framework. It enables granular, real-time monitoring of traffic conditions across broad urban regions and supports scalable, interpretable machine learning pipelines suitable for practical deployment.

2.2.3 Crowdsourced Reports

Crowdsourced traffic platforms—such as Waze, Google Maps, and TomTom—have emerged as valuable complements to traditional traffic sensing by enabling users to voluntarily report incidents such as accidents, congestion, road closures, and hazards in real time. These

platforms enrich the traffic data ecosystem with human-perceived context that may not be captured by infrastructure-based or GPS-based systems alone.

Nevertheless, several studies highlight limitations associated with relying on crowdsourced data as a primary information source. A 2020 collaboration between researchers at Tsinghua University and the University of California (Lin and Li, 2020) introduced a hierarchical machine learning architecture to estimate incident durations from user-submitted reports. While their framework demonstrated that semantic cues from crowdsourced data can enhance post-incident forecasting, it also revealed a strong dependency on rigorous filtering and ranking mechanisms to manage inconsistent and noisy input, particularly in regions with limited user engagement.

In a related context, a 2021 study led by researchers at the Korea Advanced Institute of Science and Technology (KAIST) (Kim et al., 2021) investigated the feasibility of using GPS probes and dashcam feeds from public buses to support high-definition map updates. Although technically promising, their findings pointed to significant issues with sensor noise, temporal desynchronization, and spatial inaccuracies. Achieving reliable updates required temporal aggregation and uncertainty-aware modeling, underscoring the fragility of crowdsourced inputs in isolation.

A broader evaluation conducted in 2018 by Amin-Naseri et al. (Amin-Naseri et al., 2018), encompassing multiple U.S. cities, quantified the reliability of Waze alerts. Their analysis showed that less than half of the reports could be corroborated by official ground truth data, with many incidents classified as redundant or erroneous. The authors concluded that while such reports are often timely, their lack of validation and susceptibility to duplication make them unreliable as standalone indicators of traffic conditions.

Taken collectively, these studies indicate that crowdsourced reports offer contextual richness but suffer from inconsistency, dependence on user participation, and limited verifiability. As such, they are best positioned as supplementary inputs rather than as a foundational data source for traffic analytics. In the context of this thesis, the DTIP framework excludes crowdsourced signals from its core congestion detection pipeline, favoring instead the more consistent, scalable, and privacy-compliant nature of FCD.

2.2.4 Comparative Assessment of Data Sources

The previously discussed traffic data sources—fixed-sensor infrastructure, FCD, and crowdsourced reports—each offer distinct strengths and limitations. Selecting an appropriate source for a congestion detection system depends on the operational context, system objectives, and the trade-offs between accuracy, cost, and scalability.

Fixed-sensor networks, as analyzed by Kong et al. in their 2016 study at Dalian Uni-

versity of Technology (Kong et al., 2016), provide highly accurate and real-time measurements of traffic parameters such as speed, flow, and occupancy. These systems are best suited for monitoring known bottlenecks or critical intersections, where precision is paramount. However, their deployment involves high capital and maintenance costs, and their rigid infrastructure makes them difficult to adapt to evolving urban layouts or newly emerging congestion zones. As such, they are often impractical for comprehensive, city-wide coverage in dynamic metropolitan settings.

In contrast, FCD offers superior geographic flexibility and scalability. The 2021 survey by Huang et al. (Huang et al., 2021) highlights how FCD, when supported by robust preprocessing techniques such as trajectory cleaning and map-matching, can provide high-resolution traffic insights with minimal infrastructure investment. Complementing this, Li et al. (Li et al., 2021) demonstrate that combining FCD with multi-model learning techniques enables reliable congestion state estimation even in data-sparse areas. Nonetheless, the utility of FCD hinges on effective handling of its inherent challenges—namely, spatial noise, variable sampling frequency, and data volume—typically addressed using scalable frameworks like Parquet and Dask.

Crowdsourced data introduces a fundamentally different value proposition: it provides semantically rich, user-labeled reports of incidents, hazards, and traffic disruptions. Studies such as the one by Lin and colleagues (Lin and Li, 2020) illustrate how these inputs can enhance model accuracy when carefully filtered and fused with quantitative data. However, the broader evaluation by Amin-Naseri et al. (Amin-Naseri et al., 2018) reveals that the reliability of crowdsourced platforms like Waze is limited, with less than half of the reported alerts aligning with verified ground truth. This unpredictability, combined with redundancy and susceptibility to false positives, restricts their use in high-stakes operational settings unless complemented by robust verification mechanisms.

To provide a comparative overview, Table 2.1 summarizes the qualitative differences between the three data sources across key evaluation criteria. These assessments draw on empirical insights from the aforementioned studies.

Legend: ++ (Very good), + (Good), - (Weak), – (Very weak)

Criteria	Fixed Sensors	FCD	Crowdsourcing
Scalability	–	++	+
Coverage	-	++	+
Real-time capability	+	++	+
Accuracy	++	+	-
Cost	–	+	++
Reliability	++	+	-
Data Redundancy Handling	++	+	-
Infrastructure Flexibility	-	++	+
Public Participation Dependency	–	–	++
Update Frequency Control	++	++	-
Sensor Calibration Requirement	++	N/A	N/A
Preprocessing Complexity	+	++	++

Table 2.1: Qualitative comparison of traffic data sources.

As the table illustrates, FCD offers the most favorable balance between spatial coverage, operational cost, and technical adaptability—particularly when used alongside modern open-source tools for map-matching and distributed processing. These attributes justify its adoption as the core data source for the DTIP system developed in this thesis.

2.3 Choice of Map-Matching Engine: Valhalla for Open, Scalable GPS Trajectory Correction

In systems built on FCD, converting raw GPS traces into accurate, road-aligned trajectories is an essential step. GPS signals—particularly in dense urban areas—are often affected by noise, multipath interference, and spatial drift, which can lead to incorrect segment assignments and flawed traffic metrics. A reliable map-matching engine is therefore a foundational component in any congestion detection pipeline.

While commercial services such as Google Maps API (Google, 2024), or Mapbox (Mapbox, 2024) provide high-quality map-matching capabilities, they impose restrictions that limit their usability in research or large-scale deployments. These include rate-limiting, usage caps, black-box algorithms, and cost barriers, which hinder reproducibility and experimentation.

To overcome these limitations, the DTIP framework incorporates **Valhalla**, an open-source routing and map-matching engine originally developed by Mapbox and now maintained by the open-source community. Valhalla implements a Hidden Markov Model

(HMM) approach that is particularly well-suited to noisy or sparse trajectory data (Koller et al., 2015; Valhalla Team, 2025). This probabilistic method accounts for uncertainty in GPS input and uses transition probabilities between road segments to infer the most likely path taken.

The suitability of Valhalla for scalable and high-fidelity map-matching has been validated in a 2022 study by Saki and Hagen at the University of Applied Sciences Rhein-Main (Saki and Hagen, 2022). Their large-scale benchmark, involving over 18 million GPS points in Frankfurt, demonstrated:

- **Scalability:** Valhalla was successfully deployed on cloud infrastructure (AWS EC2, Docker), enabling parallelized processing of large GPS datasets.
- **Accuracy:** The system achieved over 95% success rate in matching probes to ground-truth segments, even in complex urban environments.
- **Tunable Parameters:** The framework offers fine-grained control via parameters allowing adaptation to different sampling rates and environments.
- **Transparency and Reproducibility:** As an open-source project, Valhalla offers full access to its algorithmic implementation and is readily extensible.

Additionally, the 2021 review by Huang et al. (Huang et al., 2021) underscores Valhalla as one of the few open-source engines that integrate routing and map-matching within a unified architecture, making it especially suitable for modular urban mobility pipelines.

In the context of DTIP, Valhalla offers three main benefits:

- **Offline and Local Execution:** Eliminates dependency on third-party APIs, supporting reproducibility and offline experimentation.
- **Geographic Flexibility:** Can be reconfigured easily for new cities or road networks via downloadable map tiles.
- **Python Integration:** Its HTTP API design allows seamless integration with Python-based data pipelines and processing libraries.

Given these attributes, Valhalla is a strategic enabler of DTIP's goals: it guarantees that the input data is spatially coherent, reducing error propagation into downstream layers such as feature extraction and congestion classification.

2.4 Scalable Data Handling: Leveraging Parquet and Dask for Efficient FCD Processing

Processing FCD at urban scale presents significant challenges related to both storage and computation. With sampling frequencies between 1–2 Hz and city-wide deployments, raw trajectory data can easily exceed hundreds of millions of records in a matter of weeks. Efficiently transforming this data into structured insights demands a scalable and modular data handling strategy.

The DTIP framework addresses this by combining two technologies: **Apache Parquet**, a columnar storage format optimized for analytical workloads, and **Dask**, a Python-native library for parallel and out-of-core computation.

Choice of Parquet

Parquet is specifically designed for fast, efficient access to tabular data. Its column-oriented layout offers key advantages over traditional row-based formats such as CSV:

- **Columnar Access:** Enables reading only the fields needed for a given computation, reducing memory usage and improving speed.
- **Compression Efficiency:** Exploits redundancy in columnar data to achieve high compression ratios.
- **Schema and Type Enforcement:** Stores metadata to ensure consistency across pipeline stages.

These characteristics are particularly valuable in FCD pipelines, where certain steps (e.g., filtering or aggregation) only require a subset of features. In a recent comparative study, Antonopoulos (Antonopoulos, 2024) found that switching from CSV to Parquet led to substantial improvements in I/O throughput and overall model training times in large-scale Machine Learning (ML) pipelines.

Choice of Dask

Complementing Parquet, Dask allows Python users to process datasets that exceed available memory by:

- **Partitioning Data into Chunks:** Loads only parts of the data into memory at each step.
- **Distributing Workloads Across Cores or Machines:** Automatically parallelizes processing tasks.

- **Delaying Execution:** Builds a dependency graph (DAG) of operations and executes only when results are required.

Dask's internal scheduler makes it particularly suitable for the preprocessing and feature engineering stages in DTIP, where tasks like sorting, deduplication, and rolling-window statistics are applied to millions of records. Its seamless compatibility with Pandas and NumPy also simplifies integration with existing data science workflows.

Integrated Architecture in DTIP

In DTIP, the Parquet+Dask architecture enables scalable and modular data processing:

- GPS data is partitioned by day and hour, then stored as Parquet files on disk.
- Dask loads these partitions in parallel, applies map-matching, and extracts features.
- Intermediate computations—such as speed filtering, segment aggregation, and SRI calculation—are executed in-memory using Dask's DataFrame abstraction.
- The final dataset is persisted back to Parquet for modeling and visualization.

This strategy ensures that DTIP can operate efficiently on commodity hardware while remaining compatible with cloud-scale deployments. It also allows for easy extension to real-time applications through stream ingestion and incremental batch updates.

By leveraging Parquet and Dask together, DTIP achieves an optimal balance between speed, memory efficiency, and modularity—core requirements for the processing of high-volume urban trajectory data.

2.5 Case Study Justification: Vila Nova de Gaia as a Representative Urban Scenario

The selection of Vila Nova de Gaia as the empirical setting for this study is grounded in both analytical representativeness and practical feasibility. As the third most populous municipality in Portugal—home to over 300,000 residents—Gaia features a heterogeneous urban landscape that mirrors many of the complexities found in European mid-sized cities. These include high residential density, varied land use, multi-modal transit flows, and evolving infrastructure.

Urban and Socioeconomic Complexity

The municipality encompasses a broad range of urban typologies, from high-rise residential blocks and industrial zones to intermodal transit hubs and riverside tourism corridors. This heterogeneity supports diverse traffic profiles, including commuter congestion, logistics-related movement, and seasonal tourist flows. A 2016 vulnerability study by Fernandez, Mourato, and Moreira (Fernandez et al., 2016) employed GIS-based multicriteria analysis to identify critical sub-regions within Gaia—such as Mafamude, Oliveira do Douro, and Canidelo—as areas of infrastructural fragility and social vulnerability. These findings highlight the importance of localized, fine-grained monitoring systems, reinforcing Gaia's suitability as a proving ground for congestion detection models.

Data Availability and Trajectory Density

From a data-driven standpoint, Gaia presents an advantageous testing environment due to its high FCD availability. During an exploratory data profiling phase, a wide range of candidate cities across Portugal were analyzed to identify those with the highest density of high-frequency GPS probe trajectories. This included cities such as Vila Nova de Gaia, Braga, Lisboa, and Porto, which were selected due to their significant probe densities.

Among these, Vila Nova de Gaia consistently displayed the highest density, with over 18 million raw location records collected within a two-month span, covering a broad range of days, times, and road typologies. This dataset supports both cross-sectional and longitudinal evaluation of traffic conditions, ensuring that learned models can generalize across time bins and spatial regions.

To manage complexity and ensure computational feasibility, a spatial bounding box was applied to delimit the most active road segments in the municipality. This preprocessing step enabled effective prototyping and deployment on commodity hardware, while preserving core traffic variability—emulating realistic operational constraints in production environments.

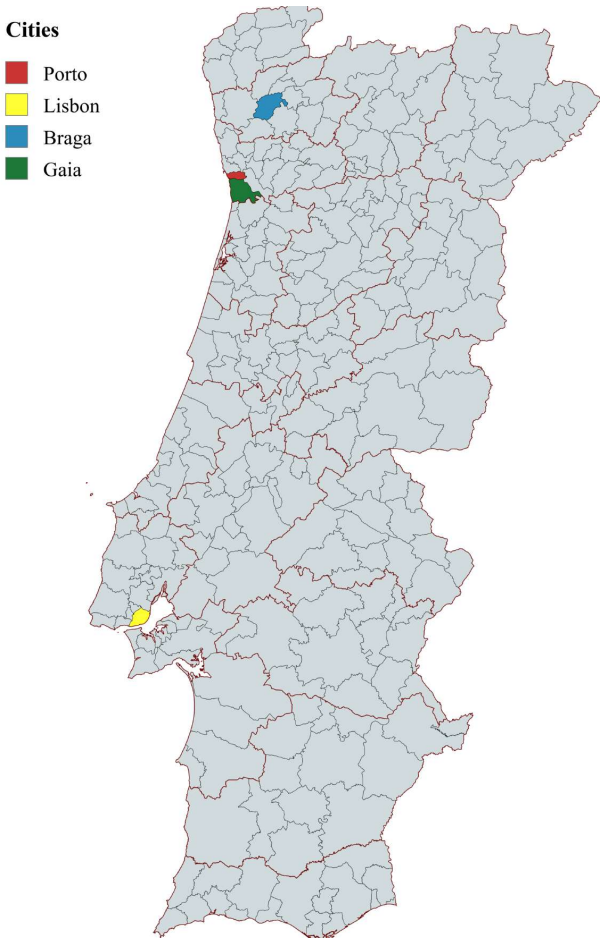

Figure 2.1: Map highlighting the selected cities with significant GPS probe densities(mapchart.net).

The map highlights the cities selected for their significant probe densities, with Vila Nova de Gaia, Braga, Lisboa, and Porto standing out. However, Vila Nova de Gaia prevailed among these due to its superior probe density.

Institutional and Policy Relevance

Gaia's inclusion in regional mobility initiatives—such as the Metro do Porto expansion, *Andante* multimodal ticketing, and smart city pilot programs—further strengthens its relevance. These ongoing initiatives call for real-time, data-driven tools for traffic monitoring, planning, and policy evaluation. By deploying the DTIP framework in this context, the study aligns with regional needs for scalable and interpretable congestion estimation systems that can inform public decision-making.

Generalizability to Other Urban Settings

Although the DTIP pipeline was developed using data from Gaia, its modular design supports portability. The framework's individual components—map-matching, feature

engineering, and congestion classification—are agnostic to geographic location and can be reconfigured for deployment in other urban areas with minimal retraining. Thus, Gaia serves as both a high-fidelity testbed and a template for future adaptation in diverse urban contexts offering a unique combination of data richness, urban complexity, and institutional alignment. These characteristics, together with its representativeness of broader European mobility patterns, make it a compelling case study for evaluating the performance, scalability, and real-world viability of the DTIP system.

2.6 Qualitative Evaluation using SUMO

Simulation-based tools such as SUMO (Eclipse SUMO, 2025b) play an important role in the qualitative evaluation of traffic models, providing visual and dynamic insights into vehicle behaviors and emergent patterns.

SUMO is an open-source microscopic traffic simulator developed by the German Aerospace Center (DLR), widely recognized for its precision in modeling individual vehicle behaviors and traffic dynamics across multi-modal urban networks (Lopez et al., 2018). Its ecosystem includes tools for map importation (e.g., from OpenStreetMap), demand generation, simulation control, and output analysis, making it a popular choice for both academic and operational traffic research. It operates at a microscopic level, simulating each vehicle as an independent agent with customizable parameters for acceleration, lane-changing, and routing, which allows for detailed replication of real-world scenarios like congestion buildup or signal-controlled intersections.

The simulator enables visual inspection of vehicle flows, traffic light dynamics, and emergent congestion patterns, which is particularly useful for assessing the behavioral plausibility of traffic predictions in controlled environments. This form of qualitative validation complements quantitative metrics by offering interpretability, especially in scenarios with limited ground-truth data.

The decision to highlight SUMO is influenced by its successful application in recent literature. For instance, Lopez et al. (Lopez et al., 2018) demonstrate its use in simulating urban mobility scenarios, emphasizing its flexibility for integrating with data-driven models to evaluate traffic flow under varying conditions. Similarly, Pandove et al. (Pandove and Pandove, 2024) employed SUMO to simulate urban traffic scenarios and evaluate congestion detection models in a controlled environment. Their methodology highlights the value of combining learned models with simulation feedback loops, enhancing explainability and model transparency.

Despite its strengths, SUMO has limitations: it does not inherently produce quantitative metrics for model scoring and relies on manual configuration, which can introduce subjectivity. Simulations are also computationally intensive for large-scale networks and may not fully capture unpredictable real-world factors like weather or human errors with-

out extensive parameterization. Nonetheless, its open-source nature and modular design make it adaptable for qualitative assessments in traffic monitoring systems.

2.7 Conclusion

This Chapter has provided a comprehensive review of the current state of the art in traffic congestion detection, with particular attention to the elements that inform the design and implementation of the proposed Distributed Traffic Intelligence Pipeline (DTIP) framework.

In the dimension of data acquisition, FCD was identified as the most promising source, offering scalability, cost-efficiency, and geographic flexibility. Compared to traditional fixed-sensor networks—whose limitations in coverage and adaptability were highlighted in the study by Kong et al. (Kong et al., 2016)—and to crowdsourced reports—whose semantic richness is offset by concerns over validation and consistency—FCD stands out as a balanced and privacy-compliant data modality.

From a methodological standpoint, the review justified the adoption of ensemble learning models, particularly Extreme Gradient Boosting (XGBoost) (Chen and Guestrin, 2016), over deep learning alternatives such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). Studies such as those by Peruthambi et al. (Peruthambi et al., 2025) and Pandove et al. (Pandove and Pandove, 2024) consistently reported superior performance, robustness, and explainability from boosting models in predictive maintenance and traffic estimation tasks, supporting their inclusion in DTIP.

At the infrastructural level, the framework combines the Valhalla map-matching engine, Parquet for efficient data storage, and Dask for parallel computation. These choices are empirically grounded in prior benchmarks (Saki and Hagen, 2022; Antonopoulos, 2024; Rocklin, 2015), and together form a scalable and modular pipeline suitable for both exploratory analysis and operational deployment.

Finally, the choice of Vila Nova de Gaia as the case study was motivated by its urban diversity, data density, and policy alignment. The city offers a microcosm of challenges encountered in larger urban areas, while providing a rich empirical substrate for model development and evaluation.

These insights collectively establish the foundations for the DTIP architecture. The next chapter will delve into the system's design, detailing how raw GPS traces are transformed—via modular, reproducible steps—into actionable congestion insights suitable for deployment in real-world mobility infrastructures.

Chapter 3

Distributed Traffic Intelligence Pipeline (DTIP)

3.1 Introduction

The development of DTIP was guided not only by research goals, but also by the ambition to produce a robust and extensible system that could be adapted for operational use in real-world settings. From the outset, the design choices prioritized code quality, modularity, and maintainability, with a strong emphasis on reproducibility and practical deployment.

To complement its quantitative modeling capabilities, DTIP includes a separate qualitative evaluation module based on Simulation of Urban Mobility (SUMO), a high-fidelity microscopic traffic simulator. This optional component allows analysts to simulate vehicle flows over selected road segments using DTIP's congestion predictions as input. While not part of the core inference pipeline, the SUMO module provides valuable visual feedback and interpretability, supporting both model validation and stakeholder engagement.

This chapter presents the methodological and architectural foundations of DTIP. It begins by describing the core design principles that shaped the system's structure, followed by an overview of the pipeline's high-level organization. Each processing layer is then examined in detail, highlighting its functional role, data transformations, and implementation strategies. The chapter concludes with a discussion of the key design trade-offs and the motivation behind the tools and technologies selected.

3.2 System Architecture

3.2.1 Design Principles

The development of DTIP was guided by five central design principles, chosen to ensure the system could operate reliably under real-world conditions and evolve to meet future demands:

- **Modularity**: The pipeline is composed of independent functional components, each with clearly defined responsibilities. This allows individual layers — such as data cleaning, map-matching, or modeling — to be modified, tested, or replaced without affecting the rest of the system.
- **Scalability**: DTIP is capable of processing millions of GPS records efficiently, even on modest computing infrastructure. This is achieved by adopting compact data formats and parallel processing techniques that reduce memory consumption and optimize computational throughput.
- **Interpretability**: Rather than relying on opaque models, the framework prioritizes techniques that are easy to understand and explain.
- **Privacy Preservation**: In line with privacy regulations such as the General Data Protection Regulation (GDPR), all data handled by the system is anonymized at the source. No personally identifiable information is stored or processed, and trajectory data is used strictly in aggregate or pseudonymized forms.
- **Portability**: The system was built with flexibility in mind. Although it was validated in a specific urban context, it can be adapted to new geographies by adjusting configuration files, supplying relevant base maps, and retraining the classification models. No structural changes to the architecture should be required.

3.2.2 High-Level Pipeline Overview

At a high level, DTIP can be understood as a multi-layered transformation pipeline. It ingests raw GPS probe data and produces congestion classifications through five main stages. Each layer is responsible for converting the data into a progressively more structured and informative representation.

1. **Data Preprocessing**: The system begins by ingesting raw GPS records, often in CSV format. These files are validated and cleaned to remove incomplete, noisy, or anomalous entries. To support efficient downstream processing, the cleaned data is then converted into a compressed columnar format, enabling fast access to specific features and compatibility with parallel processing tools.

2. **Regional Filtering**: After preprocessing, the dataset is geographically restricted to the study area (e.g., Vila Nova de Gaia). This step removes irrelevant data outside the region of interest, reducing computational load and ensuring that only locally relevant probes are processed downstream.

3. **Map-Matching**: The filtered probe points are aligned to the underlying road network using a statistical trajectory correction algorithm. This process—known as map-matching—projects each GPS coordinate onto its most likely location along the road graph, correcting for GPS noise and ensuring spatial coherence.

4. **Feature Engineering**: The spatially aligned data is segmented into fixed-duration intervals, within which a variety of traffic-related metrics are calculated. These include motion statistics (such as average speed and stop duration), temporal attributes (such as hour of day), and most importantly, derived congestion indicators such as the Speed Reduction Index, which is used as the classification label for the model.

5. **Modeling and Inference**: Finally, a supervised learning model is applied to predict congestion levels based on the extracted features. The model provides both class labels and probability scores, which can be aggregated into visualizations or used to support real-time alerts and long-term planning.

Figure 3.1 provides a visual overview of the complete DTIP architecture. Each layer is represented as a modular block, illustrating the logical flow from raw GPS input to congestion-level output. This figure also highlights the main data transformations and intermediate outputs that enable reproducibility and modular experimentation.

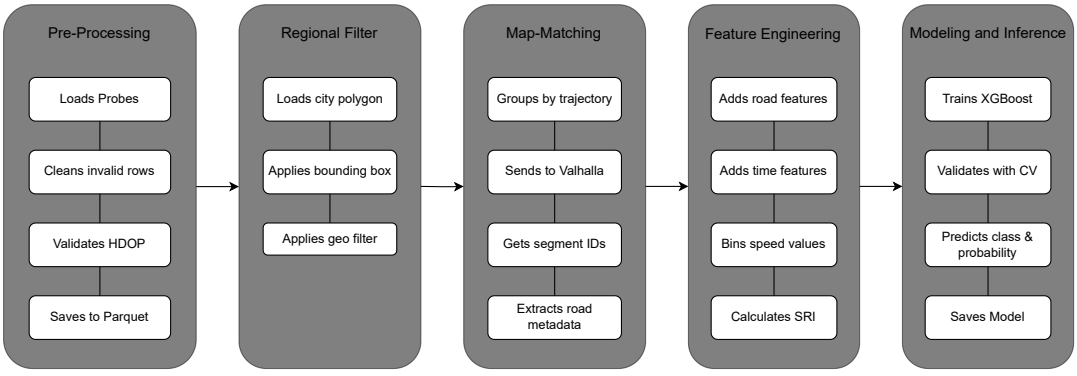

Figure 3.1: Overview of the DTIP pipeline architecture

This separation of concerns not only improves robustness and maintainability but also enhances the interpretability of each decision within the pipeline. For example, errors in map-matching can be traced and corrected without altering the feature engineering logic, and model outputs can be explained in terms of specific input features. These

properties are essential for practical deployment in government or municipal mobility systems, where accountability, adaptability, and transparency are central.

3.3 Pipeline Functionality

This section offers a detailed examination of the five main layers that make up the DTIP pipeline. Each layer performs a distinct transformation on the input data and passes the results downstream in a structured and standardized format.

Throughout the pipeline, all intermediate outputs are saved in partitioned Parquet files with predefined schemas. This approach not only facilitates computational efficiency but also enhances the pipeline's maintainability, making it easier to inspect, debug, and adapt across different deployment contexts.

3.3.1 Preprocessing Layer

The preprocessing layer marks the starting point of the DTIP pipeline. Its main purpose is to convert raw GPS data—typically collected from probe vehicles and stored in semi-structured CSV files—into a clean, validated, and efficient format suitable for analysis and modeling.

In real-world conditions, this raw data is rarely clean. It often includes a mix of invalid timestamps, duplicated records, incomplete coordinates, and abrupt positional jumps due to poor GPS reception, particularly in dense urban environments where signal interference is common. If left unfiltered, these issues can compromise the accuracy of map-matching and corrupt the results of downstream models.

To ensure data quality from the outset, DTIP applies a series of structured filtering and validation steps:

- **Coordinate Validation:** All GPS points are checked to confirm that latitude and longitude values fall within valid bounds. Any record with missing or obviously invalid coordinates (e.g., latitudes beyond `[-90, +90]` or longitudes beyond `[-180, +180]`) is discarded.
- **Timestamp Validation:** Records without timestamps, or with improperly formatted or duplicated timestamps, are removed. The data is also chronologically sorted per vehicle, ensuring consistent temporal ordering of each trajectory.
- **Speed Outlier Detection:** For each point, the instantaneous speed between consecutive positions is computed. If a segment implies speeds beyond a realistic threshold (e.g., 180 km/h), the associated point is considered an outlier and excluded. This helps mitigate the impact of signal glitches, transmission errors or even huge outliers like high velocities which do not indicate normal traffic flow.

- **Signal Quality Filtering:** Metadata such as HDOP (Horizontal Dilution of Precision)—a measure of GPS signal accuracy based on satellite geometry—is used to assess location quality. Following common practice in GNSS-based studies, points with $HDOP > 5$ are considered unreliable and removed, as they typically correspond to weak satellite coverage or obstructed environments. This threshold is consistent with findings by Breili and Lund (Breili and Lund, 2025), who showed through large-scale simulations of GNSS performance in urban corridors that HDOP values above 5 frequently occur in areas with limited sky visibility and are associated with substantial degradation in positioning accuracy.

By enforcing these quality control measures early in the pipeline, DTIP ensures that all subsequent processing steps operate on a stable and trustworthy foundation. Clean data not only improves model accuracy but also enhances transparency and reproducibility—qualities that are essential in public-facing urban analytics systems.

3.3.2 Regional Filtering Layer

Before diving into map-matching, the DTIP pipeline performs a spatial filtering step to restrict processing to the geographic area of interest. This is especially important in large datasets covering vast regions or entire countries, where only a specific urban area—such as Vila Nova de Gaia—is both relevant for the analysis and feasible to process with the available hardware.

Without this filtering, the sheer volume of GPS data would exceed the memory and processing capacity of the computing environment, making it impractical to execute the full pipeline efficiently.

To achieve this, the system first retrieves the geographic boundaries of the target region using OpenStreetMap (OSM). The polygon describing this region is either loaded from cache or fetched dynamically using the OSMnx toolkit. Once retrieved, it is used to define both a bounding box filter and a more precise spatial filter.

The filtering itself is done in two stages:

1. **Bounding Box Filtering**: GPS points that fall outside the rectangular extent of the target region are discarded. This step is computationally efficient and eliminates the bulk of irrelevant data.

2. **Polygon-Based Filtering**: A more accurate spatial filter is then applied using the region's actual polygon geometry. Only GPS points strictly within the region's borders are retained.

Both filters are implemented using a parallelized Dask pipeline and GeoPandas operations, allowing for scalable filtering of millions of records. The resulting dataset includes

only the GPS points that lie within the region of interest, making subsequent operations faster and more focused.

3.3.3 Map-Matching Layer

Once spatial filtering is applied, the next step in the DTIP pipeline is to correct the inherent spatial inaccuracies in raw GPS data by aligning each probe point to its most probable position on the road network. This process, known as *map-matching*, is essential to convert noisy trajectories into meaningful, road-aware paths suitable for traffic modeling and congestion estimation.

In real-world deployments, particularly in dense urban environments, GPS traces are frequently distorted by signal drift, multipath effects, or irregular sampling. Without proper correction, the raw coordinates would not align with actual road segments, leading to inaccurate speed calculations and misclassification of traffic conditions.

To address this challenge, DTIP employs Valhalla, an open-source map-matching engine originally developed by Mapbox. Valhalla uses a probabilistic Hidden Markov Model (HMM) that considers both spatial proximity and directional coherence to infer the most likely road path for each trajectory. This makes it particularly robust in complex urban networks where many candidate roads may lie near a given GPS point.

Unlike traditional map-matching systems that treat GPS points as isolated observations—ignoring temporal order and continuity—Valhalla performs matching on entire trajectories. This distinction is crucial: matching full, time-aligned sequences allows the model to reason about directionality, acceleration, and movement patterns across segments, producing more reliable and coherent paths. It avoids problems like false segment jumps or direction reversals that are common in single-point map-matching approaches.

Each validated trajectory is submitted to a locally hosted Valhalla server through HTTP requests. The engine returns enriched trajectory data that includes:

- **Matched Segment IDs**: The ordered sequence of road segments corresponding to the original GPS path.
- **Confidence Scores**: A likelihood score indicating the reliability of each matched point.
- **Edge Geometry and Lengths**: Detailed information about the road geometry that supports further segmentation and feature extraction.

Importantly, Valhalla is run locally within a Docker container, giving DTIP full control over the engine's configuration, resource allocation, and performance. This setup eliminates the limitations associated with commercial APIs, such as rate limits, usage quotas, and dependency on internet access. As a result, the system supports reproducible,

scalable batch processing—essential for both research and operational deployment at city scale.

3.3.4 Feature Engineering Layer

After the map-matching and regional filtering stages, DTIP proceeds to convert enriched GPS trajectories into a structured and meaningful representation of traffic behavior. This is achieved through a multi-step feature engineering process that captures both real-time dynamics and contextual attributes of road usage. The resulting features serve as inputs to the final congestion classification model.

The process is implemented in four main steps: (1) computing free-flow speeds for each road segment, (2) extracting aggregated behavioral metrics, (3) calculating the Speed Reduction Index (SRI), and (4) assembling temporal and contextual features.

Step 1: Estimating Free-Flow Speeds

To enable context-aware congestion analysis, DTIP estimates the expected "free-flow" speed for each road segment using probe data collected during off-peak hours. For this, the system isolates data from late-night or low-traffic periods (typically between 1:00 AM and 6:00 AM) and applies a quantile-based heuristic:

- For motorways, the 90th percentile of observed speeds is taken as the free-flow speed.
- For all other road classes, the 80th percentile is used.

These estimates are adjusted to ensure plausibility—for example, motorway speeds below 80 km/h are raised, while residential speeds above 40 km/h are capped. In cases where the computed free-flow speed is unrealistically low, the system falls back to the segment's default speed limit, avoiding anomalous ratios in the next step.

The rationale for adopting the 90th and 80th percentiles is supported by empirical findings in traffic engineering research. High quantiles are widely used to approximate free-flow conditions, as they capture the upper portion of observed speed distributions while filtering out temporary slowdowns or measurement errors. Silvano et al. (Silvano et al., 2020) demonstrated that free-flow speeds can be robustly estimated through probabilistic models grounded in the upper tail of speed distributions, highlighting the relevance of percentile-based approaches. Similarly, Leong et al. (Leong et al., 2019) validated free-flow speed models on multilane highways under heterogeneous traffic, showing that high-percentile measures provide stable estimates across varying conditions. Building on these practices, DTIP adopts the 90th percentile for motorways, which are designed for

uninterrupted, high-speed travel, thus ensuring that the free-flow estimate reflects realistic upper operating bounds.

For other road classes, greater variability is expected due to intersections, signals, and local disturbances. In such contexts, a slightly lower threshold—the 80th percentile—offers a balanced estimate that still reflects uncongested conditions while accommodating inherent operational fluctuations. This is consistent with evidence from Stepanović et al. (Stepanović et al., 2023), who emphasized that road class characteristics and traffic interruptions must be accounted for in free-flow estimation. By differentiating thresholds across road types, DTIP aligns with established practices and ensures reliable speed estimates across diverse traffic environments.

Step 2: Aggregating Traffic Behavior Features

Next, DTIP computes motion-based indicators by aggregating behavior over each road segment and trajectory:

- **Average Segment Speed**: The mean speed of all probes within the segment.
- **Speed Distribution**: Probe speeds are binned into five intervals (0–20, 20–40, 40–60, 60–80, 80–100 km/h), and the relative frequency of each bin is computed.
- **Stop Behavior**: Two features are derived:
  - The *maximum continuous stopped time*, identifying the longest pause per edge and trajectory.
  - The *total stopped time* across the trajectory in the segment.

These features provide detailed insight into flow stability, helping distinguish smooth traffic from stop-and-go or gridlocked conditions.

Step 3: Computing the Speed Reduction Index (SRI)

The Speed Reduction Index is a cornerstone of DTIP's methodology. It quantifies congestion by comparing observed probe speed to the expected free-flow speed for that segment:

$$\text{SRI} = 1 - \frac{\text{probe speed}}{\text{effective free-flow speed}}$$

The resulting index is clipped between 0 and 1 and is robust to outliers. When the estimated free-flow speed is lower than the segment's expected value, the system falls

back to the default segment speed to avoid artificially inflating the SRI. This fallback mechanism is especially important in under-sampled roads or inconsistent classes.

SRI is not only used as a feature—it also serves as the foundation for labeling congestion classes during model training. Compared to absolute speed thresholds or historical travel-time baselines, SRI has proven to be more stable, context-aware, and transferable across road types and urban environments. Its relative nature ensures that congestion is defined not by fixed cutoffs but by deviations from expected free-flow behavior. In a later section of this document, we will present in detail how the SRI distribution was used to define the class labels for supervised learning.

Step 4: Enriching with Temporal and Contextual Features

After the core cleaning and map-matching steps are completed, each individual probe is enriched with a diverse set of contextual and temporal attributes. These features capture not only the immediate behavior of the vehicle but also structural properties of the road network and broader temporal patterns that affect traffic flow. This enrichment is essential for enabling accurate congestion classification by allowing the model to reason over situational context rather than raw motion alone.

- **Timestamp decomposition**: The exact timestamp of each probe is decomposed into multiple temporal indicators:
  - *Day of week (0–6)* — Captures weekly traffic patterns such as weekday rush hours vs. weekend leisure traffic.
  - *Hour of day (0–23)* — Used to capture diurnal patterns like morning vs. evening peaks.
  - *Weekend indicator (binary)* — Separates working days from weekends.
  - *Peak hour indicator (binary)* — Flags high-congestion windows, specifically 08:00–10:00 and 17:00–19:00, known to correspond to commuting peaks.

  These time-based features allow the model to implicitly learn how traffic varies with human activity cycles and urban rhythm.

- **Road segment metadata (from Valhalla)**: Each probe is matched to a road segment via Valhalla's map-matching output, from which several additional features are extracted:
  - *Segment length (in meters)* — Longer segments typically support higher speeds and traffic volumes.

- *Road classification (e.g., motorway, trunk, residential)* — Derived from OpenStreetMap tags and encoded into categorical and one-hot formats to represent infrastructure hierarchy.
- *Allowed speed / speed limit* — Where available, used to compare actual vs. theoretical speeds (though not always available in OSM).
- *Edge ID or unique segment identifier* — Ensures temporal aggregation and feature computation (e.g., speed averaging) can be tracked at the road segment level.
- *Directionality (forward/backward)* — Indicates whether the matched probe follows the road's intended direction.

These attributes enable a more structured interpretation of motion. For example, a speed of 30 km/h may be expected on a residential street but would indicate congestion on a highway. By including road class, segment length, and type, the model can disambiguate such cases.

This enrichment layer transforms low-level probe data into a structured and semantically rich feature set. It bridges the gap between raw geospatial telemetry and high-level congestion semantics, laying the foundation for interpretable, data-driven modeling of urban traffic dynamics.

3.3.4.1 Final Output Schema

The final feature set computed by the DTIP feature engineering layer includes a diverse set of descriptors designed to capture the multifaceted nature of urban traffic. These features are grouped into three main categories:

- **Motion and Congestion Features**: These features characterize the dynamic behavior of vehicles within each road segment. Key inputs include the instantaneous probe speed (`probe_speed`) and the average speed computed across the segment (`avg_speed_segment`), both of which reflect real-time traffic conditions. Additionally, stop-related features capture stationary patterns, such as the longest continuous stop (`max_continuous_stopped_time`) and the total time vehicles remain stopped per segment and trajectory (`total_stopped_time_per_edge_per_traj`). Frequency-based features are also extracted to quantify how often vehicles fall within specific speed bins (e.g., 0–20, 20–40 km/h), enabling a more granular representation of flow variability.
- **Temporal Features**: These features contextualize each segment within its temporal setting, accounting for typical traffic cycles and behavioral patterns. The features include the hour of the day (`hour`), the day of the week (`day_of_week`), a binary

indicator for peak-hour periods (`is_peak_hour`), and a weekend flag (`is_weekend`). This temporal encoding allows the model to distinguish between structural patterns (e.g., rush hours) and anomalous behaviors.

- **Road Metadata**: To capture structural and contextual differences across the road network, several static attributes are included. These consist of the unique segment identifier (`edge_id`), the physical length of the segment (`length_segment`), the road classification as a string (e.g., motorway, residential) in `road_class`, and its corresponding integer encoding in `road_class_encoded`.

  Importantly, while the Speed Reduction Index (`SRI`) is a central metric used to define the congestion class labels, it is **not** included as an input feature to the machine learning model. This ensures that the model's predictions are not circular or trivially correlated with the target variable, preserving methodological integrity.

Together, these features provide a rich and interpretable representation of traffic dynamics, road characteristics, and contextual variation—serving as the backbone for the congestion classification models discussed in the following chapters.

3.3.4.2 Scalability and Implementation Notes

This entire feature engineering process is orchestrated using two key components designed for efficient and scalable data handling in Python: **Dask** and **PyArrow**.

- **Dask** enables parallel and out-of-core computation by partitioning the dataset into manageable chunks and distributing tasks across available CPU cores. This allows DTIP to process hundreds of millions of GPS records using commodity hardware, without requiring the full dataset to reside in memory. Dask integrates seamlessly with the Pandas ecosystem, enabling smooth adoption within existing data science workflows and simplifying complex transformations such as rolling statistics or group-based aggregations.

- **PyArrow** is a Python library that provides bindings to the Apache Arrow project, which defines a high-performance in-memory columnar data format. Within DTIP, PyArrow is used for reading and writing datasets in **Parquet** format—a binary, compressed, columnar storage format optimized for analytical workloads. PyArrow ensures fast I/O operations and a low memory footprint while also enforcing schema consistency and type safety across all processing stages. This last aspect is particularly important in DTIP, as each pipeline module defines its own expected schema, facilitating modular development, debugging, and long-term maintainability.

All intermediate computations—such as speed filtering, temporal feature aggregation, and segment-level statistics—are performed in-memory using Dask and then persisted

in compressed Parquet format. This design supports rapid iteration and experimentation while ensuring full reproducibility, as every processing step leaves behind a traceable and immutable data artifact.

In summary, the feature engineering layer is where DTIP extracts its core traffic intelligence. By transforming noisy, high-frequency GPS traces into rich and interpretable features, it bridges the gap between raw mobility data and machine learning—enabling scalable congestion estimation, exploratory analysis, and transparent urban policy evaluation.

3.3.5 Modeling and Inference Layer

The final layer of the DTIP pipeline is responsible for translating engineered features into a classification of congestion severity. After each road segment and time window is described via interpretable metrics—such as speed, stop time, and temporal context—a supervised learning model is used to infer the traffic condition.

Labeling Strategy with SRI

DTIP defines congestion levels using the Speed Reduction Index (SRI), which compares observed probe speeds with the effective free-flow speed of each segment. This strategy ensures that congestion labeling is context-aware, adapting to each road class and its expected performance. Specifically, the following thresholds are used to map SRI values to discrete classes:

Class 0 – Free Flow:	$\text{SRI} < 0.15$
Class 1 – Light Congestion:	$0.15 \leq \text{SRI} < 0.35$
Class 2 – Moderate Congestion:	$0.35 \leq \text{SRI} < 0.70$
Class 3 – Severe Congestion:	$\text{SRI} \geq 0.70$

This approach proved more robust and transferable than using fixed speed thresholds or travel-time deviations, particularly when generalizing across road types or cities.

Model Choice: XGBoost

To classify congestion levels, DTIP employs the XGBoost algorithm—a gradient-boosted decision tree model optimized for structured data. XGBoost offers a compelling balance of performance, interpretability, and training efficiency. Unlike neural networks, it handles missing values and feature interactions implicitly and performs well even with moderate training data.

The model takes as input a curated set of handcrafted features, including:

- **Temporal Indicators:** Hour of day, day of week, peak hour flag, weekend flag.

- **Traffic Metrics:** Probe speed, average segment speed, speed distribution across predefined bins.
- **Stop Behavior:** Total and maximum stopped time per trajectory within each segment.
- **Road Characteristics:** Segment length and encoded road class.

These features are standardized using a fitted scaler before training and inference to ensure uniform learning dynamics.

Training and Evaluation Procedure

The dataset is first split into training and test sets based on trajectory IDs to avoid data leakage. A 70-30 split is used, ensuring that all probe points from a given trajectory fall in the same set. Additionally, the system supports optional cross-validation using grouped K-Folds, where the folds respect trajectory boundaries. This is crucial to simulate deployment on unseen vehicle paths.

The 70-30 split, grouped by trajectory IDs, was selected to prevent data leakage in this spatiotemporal dataset, where intra-trajectory correlations (e.g., sequential GPS probes) could otherwise inflate performance estimates. This grouping ensures all points from a vehicle path remain in one set, mimicking deployment on unseen trajectories. The ratio follows standard ML practices for balanced training and evaluation, providing ample data (70%) for learning complex congestion patterns while reserving 30% for reliable metric computation, especially given potential class imbalances in SRI-derived labels (Nguyen et al., 2018).

Alternative splits (e.g., 80-20) were considered but not pursued due to the high baseline performance (¿97% weighted F1) and computational efficiency priorities. Literature on trajectory-based ML, such as in GPS traffic prediction, supports 70-30 as effective for large datasets, with grouped folds (as in the time-series variant) offering similar results. Future work could explore sensitivity to splits for further optimization.

Cross-validation was employed to assess the model's generalization and robustness across folds, providing a comparison with the hold-out test set performance. It was not used for hyperparameter optimization, as the default parameters of XGBoost yielded sufficiently high performance on this tabular dataset. In many applications, particularly with gradient boosting models on structured data, default hyperparameters perform well out-of-the-box without extensive tuning, as tuning may yield diminishing returns or no improvement when the defaults are already effective (Shwartz-Ziv and Armon, 2021).

Model performance is evaluated using standard classification metrics:

- **Accuracy**: Proportion of correctly classified samples.

- **F1 Score (Weighted)**: Balances precision and recall across all congestion classes.
- **Confusion Matrix**: Highlights misclassifications and class imbalances.

These results are logged and used to diagnose performance across congestion levels.

Inference Outputs

At prediction time, the model returns both class labels and confidence scores (i.e., probabilities across the four classes). This dual output allows DTIP to support multiple use cases:

- Real-time congestion classification on road segments.
- Aggregated traffic heatmaps for visualization.
- Threshold-based alerting systems for operational response.

All inference outputs are stored in structured format and can be easily visualized or exported to dashboards.

Model Persistence and Reusability

Trained models and scalers are saved as binary files, ensuring that inference can be performed without retraining. This supports reproducibility, auditability, and future updates.

Overall, this modeling layer combines explainable machine learning with scalable deployment strategies—ensuring that congestion predictions are both actionable and grounded in meaningful traffic indicators.

3.3.6 Experimental Time-Series Variant

In parallel with the main DTIP architecture, a time-series variant of the model was developed to test whether temporal dependencies across consecutive vehicle observations could provide additional predictive power. Unlike the baseline model—which treats each road segment observation independently—this experimental module was designed to incorporate short-term historical context through a sliding window mechanism.

The implementation follows the same design principles as the rest of the pipeline, being modular, reproducible, and compatible with the existing data processing logic. It is encapsulated in a class which supports both congestion classification (based on the SRI thresholds) and speed regression (predicting the vehicle's speed $t + h$ seconds into the future).

For the time-series modeling, two key parameters were introduced:

Window Size (5): Each input sample was composed of the past five time steps of a given trajectory, capturing short-term patterns in probe speed, average segment speed, road class, and other handcrafted features.

Prediction Horizon (1): The model aimed to predict the congestion class (or speed) exactly one step ahead of the current time window, corresponding to near-term traffic evolution.

The model first constructs fixed-length sequences for each `traj_id`, flattening them into feature vectors. These sequences are then used to train a classifier (XGBoost, in the case of congestion prediction) or a regressor (for direct speed prediction). During cross-validation, a `GroupKFold` strategy is applied using the trajectory ID to prevent data leakage and preserve temporal integrity.

Preliminary experiments using 5-fold cross-validation showed that the time-series classifier performed comparably to the baseline DTIP model in terms of weighted F1-score and log loss, with some improvements in the moderate and light congestion classes. However, due to the significantly increased memory and processing requirements associated with generating and storing time-series windows for millions of GPS records, further exploration was constrained by hardware limitations.

Despite not being used in the final deployment pipeline, this time-series model illustrates the extensibility of DTIP and provides a foundation for future work on real-time congestion forecasting, trajectory-level anomaly detection, or short-term speed prediction under dynamic traffic conditions. Its inclusion in this thesis serves to highlight potential directions for improving model responsiveness through temporal awareness.

3.3.7 Future Model Updates

At the current stage, the DTIP model was trained once on the available dataset and then used for inference. No automated retraining or adaptive update mechanisms were implemented. Nevertheless, in a real-world deployment the model would eventually need to be updated as traffic patterns, road networks, and driving behaviors evolve. This could involve retraining the model periodically with newly collected FCD, re-estimating free-flow speeds, or adapting the congestion thresholds used for labeling. While outside the scope of this thesis, these aspects represent natural directions for extending the pipeline towards long-term operational use.

3.4 Qualitative Simulation Layer

Although external to the DTIP core pipeline, a separate module is available for qualitative evaluation through microscopic traffic simulation. This component leverages Simulation

of Urban Mobility (SUMO) to visualize and assess the practical implications of DTIP's congestion classifications.

SUMO is an open-source traffic simulator developed by the German Aerospace Center, widely used in research and operational contexts to replicate fine-grained vehicle behavior under diverse urban conditions (Lopez et al., 2018). In DTIP, SUMO is leveraged to visualize the impact of congestion classifications by simulating vehicle flows over selected network segments with varying predicted congestion levels.

The SUMO integration is implemented through a dedicated simulation pipeline comprising the following steps:

1. **Parsing Model Outputs:** Congestion predictions produced by DTIP are post-processed and mapped to SUMO-compatible formats using a custom parser. Each prediction is linked to a road segment and assigned a corresponding traffic behavior (e.g., reduced speed or stop probability).

2. **Network Preparation:** Road network data and route files are generated based on OpenStreetMap extracts and the spatial footprint of the DTIP deployment. A local SUMO configuration is built to reflect the simulated environment.

3. **Simulation Execution:** The SUMO engine is invoked with the customized input files. Each simulation run replays traffic under different congestion conditions, offering a dynamic and visual assessment of how DTIP's classifications translate into real-world-like flows.

4. **Visual Feedback and Debugging:** Through SUMO's graphical interface or command-line outputs, analysts can inspect whether congestion predictions are realistic and spatially coherent. This is particularly valuable during model development or for stakeholder demonstrations.

While not part of the DTIP's primary inference pathway, this simulation layer offers an interpretability bridge—allowing traffic experts to assess whether predicted congestion levels yield plausible macroscopic effects when instantiated in a simulated environment.

The choice of SUMO is supported by its use in recent literature, including Pandove et al. (Pandove and Pandove, 2024), who combine machine learning with simulation to validate urban mobility predictions. Similarly, SUMO's flexibility and high fidelity make it a strong candidate for integrating with learning-based traffic systems.

Future versions of DTIP may consider tighter integration with SUMO to support feedback loops or scenario-based stress testing, particularly for policy evaluation or synthetic data generation.

3.5 Conclusion

This chapter has presented the architectural and methodological foundations of the DTIP, a scalable and modular framework for processing FCD to estimate urban traffic congestion. Guided by principles of modularity, scalability, interpretability, privacy preservation, and portability, DTIP transforms raw GPS trajectories into actionable congestion insights through a series of interconnected layers: preprocessing for data validation and cleaning, regional filtering to focus on relevant geographic areas, map-matching using Valhalla for accurate trajectory alignment, feature engineering to extract meaningful traffic indicators such as the Speed Reduction Index (SRI), and a modeling layer leveraging XGBoost for supervised classification of congestion severity.

Additionally, an experimental time-series variant was explored to incorporate temporal dependencies, demonstrating the pipeline's extensibility for future enhancements in forecasting capabilities. The integration of a qualitative simulation layer using SUMO further supports behavioral validation and interpretability, bridging quantitative predictions with visual assessments of traffic dynamics.

Overall, DTIP's design emphasizes open-source tools and efficient data handling, making it suitable for low-latency deployment in real-world urban monitoring systems. By addressing key challenges in traffic data processing and analysis, this pipeline lays a robust groundwork for accurate congestion detection and hazard validation. The following chapter evaluates DTIP's performance through quantitative metrics and qualitative simulations, providing empirical evidence of its effectiveness.

Chapter 4

Evaluation

4.1 Introduction

This chapter presents the evaluation of the DTIP framework from two complementary perspectives: (i) the quantitative performance of the trained congestion classification model and (ii) a qualitative assessment of prediction plausibility using the SUMO traffic simulator. The evaluation focuses on the model's ability to generalize to unseen data in a realistic urban setting—Vila Nova de Gaia—and assesses whether predicted congestion levels align with traffic behavior under synthetic conditions.

4.2 Quantitative Evaluation

The core task of DTIP is to classify road segment–level congestion into four discrete classes: none, light, moderate, and severe. To accomplish this, the XGBoost model was trained and tested using both a 70-30 train/test split and 5-fold cross-validation.

4.2.1 Hyperparameter Configuration

The XGBoost model was implemented using the default hyperparameters provided by the 'XGBClassifier' from the 'xgboost' library, as these settings have been shown to perform robustly on tabular data without extensive tuning.

No grid search or other hyperparameter optimization techniques were applied, as the default settings yielded a weighted F1 score exceeding 97%, indicating sufficient performance for the task. The decision to rely on defaults was motivated by computational efficiency considerations in the Dask-based pipeline and the empirical evidence that tuning often provides marginal gains for gradient boosting models on structured datasets like this one (Shwartz-Ziv and Armon, 2021). Future work could explore hyperparameter tuning (e.g., via grid search with cross-validation) to potentially enhance performance, particularly if additional computational resources become available.

4.2.2 Cross-Validation Usage

The 5-fold cross-validation with grouped K-Folds (respecting trajectory boundaries) was employed to assess the model's generalization and robustness across different subsets of the data, providing a comparative benchmark to the 70-30 hold-out test set. It was not used for hyperparameter tuning or grid search, as the focus was on evaluating performance rather than optimizing model parameters. The grouped K-Fold approach ensured that all probe points from a given trajectory remained in the same fold, simulating deployment on unseen vehicle paths and preventing data leakage. This evaluation strategy confirmed the model's stability, with average cross-validation metrics closely aligning with the hold-out test results.

4.2.3 Labeling Strategy

Congestion classes were derived from the Speed Reduction Index (SRI), using the following thresholds:

Class 0 – Free Flow:	$\text{SRI} < 0.15$
Class 1 – Light Congestion:	$0.15 \leq \text{SRI} \leq 0.35$
Class 2 – Moderate Congestion:	$0.35 < \text{SRI} \leq 0.70$
Class 3 – Severe Congestion:	$\text{SRI} > 0.70$

Although these thresholds were chosen empirically based on the distribution of SRI in the dataset and the literature previously reviewed, future work should consider validating them against labeled real-world congestion data. At present, the model's training and evaluation rely on derived labels, and performance could be further supported by human-annotated ground truth.

4.2.4 Experimental Setup

- **Dataset:** FCD collected by NDrive, filtered and processed as described in Chapter 3.
- **Model:** XGBoost classifier trained on the engineered features, including speed profiles, stop durations, temporal metadata, and road attributes.
- **Evaluation:** Metrics include accuracy, weighted F1 score, confusion matrix, and classification report. Results are reported for both 70-30 split and 5-fold cross-validation.

4.2.5 XGBoost Results

XGBoost achieved excellent classification performance, with minimal variation between cross-validation folds and test set.

Metric	XGBoost
Accuracy (70-30)	97.06%
F1 Score (70-30)	97.06%
CV Avg Accuracy	97.07%
CV Avg F1 Score	97.07%

Table 4.1: Performance of XGBoost on congestion classification.

The confusion matrix revealed that the classifier handled both extreme classes (free flow and severe congestion) with high precision and recall. Misclassifications occurred primarily between light and moderate congestion, reflecting the inherently fuzzy boundary between these categories.

4.2.6 Visual Metrics Analysis

To complement the standard metrics, a series of visualizations were generated to better illustrate the behavior and internal mechanics of the trained model. These plots allow a more nuanced understanding of class-wise performance, feature contribution, prediction balance, and fold stability.

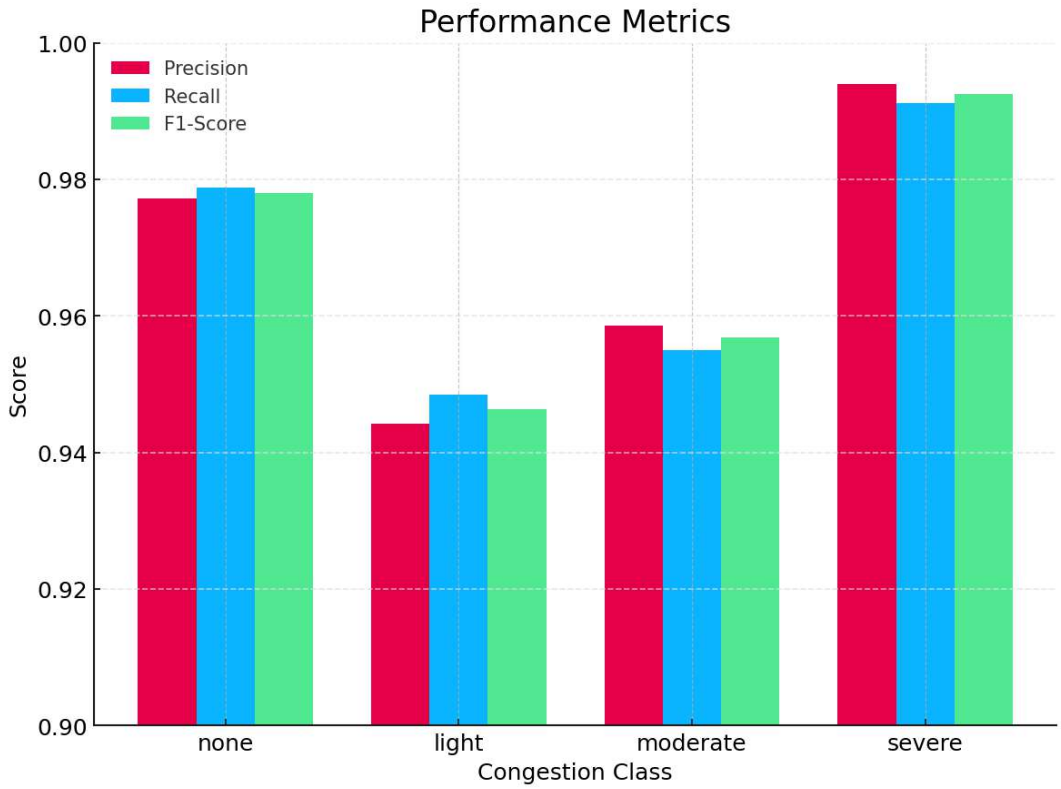

Figure 4.1: Per-class precision, recall, and F1-score.

Figure 4.1 presents the precision, recall, and F1-score obtained individually for each congestion class. The `free-flow` and `severe` congestion categories achieve the highest values across all three metrics, confirming that these classes are more distinct and easier for the model to separate.

In contrast, the `light` and `moderate` congestion levels exhibit slightly lower values, which reflects their ambiguous nature. These transitional states often lack sharply defined boundaries in both feature space and traffic behavior, making them inherently harder to classify. The consistent but moderate decline in precision and recall for these classes suggests the model is sensitive to this fuzziness, though it still maintains strong performance overall.

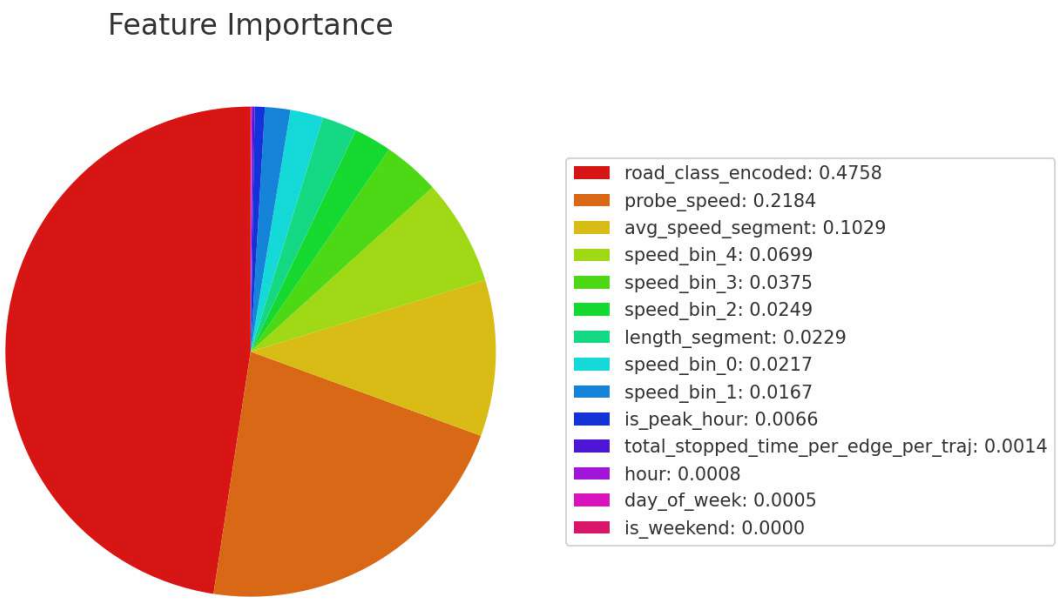

Figure 4.2: Feature importance (XGBoost gain).

Figure 4.2 illustrates the relative importance of each feature based on the average gain across decision trees in the XGBoost model. The top contributors—`road_class_encoded` and `probe_speed`—together account for more than 60% of the total model gain, indicating that congestion levels are highly driven by structural road characteristics and instantaneous vehicle speed.

This aligns well with domain intuition: road class encapsulates the expected design and behavior of the road (e.g., highways vs. local streets), while probe speed reflects real-time traffic dynamics. Other features such as segment length, time-of-day indicators, and stop durations contribute less individually but may still provide complementary signals when combined.

Although time-related variables such as `hour`, `day_of_week`, `is_peak_hour` and `is_weekend` are commonly regarded as strong determinants of traffic behavior, their relative importance in this model was marginal. This can be explained by several factors. First, their effects are largely captured indirectly through features such as instantaneous and average speed, which already encode rush-hour slowdowns and weekend variations in flow. Second, the dataset covers only two months in a single city, which limits the diversity of temporal patterns and reduces the discriminative power of weekly or seasonal cycles. Third, the labeling strategy based on the Speed Reduction Index (SRI) inherently favors speed- and road-related attributes, making temporal indicators less critical for separating classes. Finally, tree-based models like XGBoost tend to prioritize features that offer sharp and

consistent splits—such as road class or probe speed—while binary temporal flags often provide weaker gains. As a result, time-based features should still be interpreted as contextually relevant, but in this setting they acted primarily as redundant signals rather than dominant predictors.

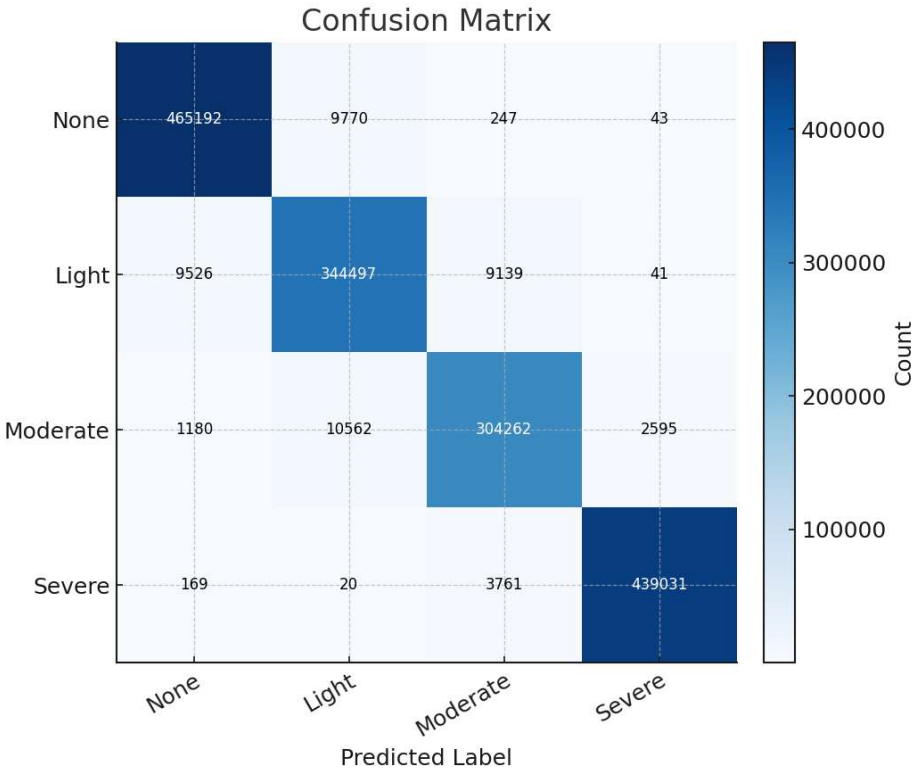

Figure 4.3: Confusion matrix with raw counts for the test set.

Figure 4.3 presents the confusion matrix, providing a comprehensive overview of the model's classification performance across the congestion classes: None, Light, Moderate, and Severe. The diagonal elements highlight the model's ability to correctly identify instances within each class, with particularly strong performance for None and Severe, where the predictions align closely with the true labels. In contrast, the off-diagonal elements reveal areas of confusion, notably between Light and Moderate, where the model struggles to distinguish between these intermediate congestion levels due to their overlapping characteristics. This pattern suggests that while the model excels at separating extreme congestion states, the transitional classes present a challenge, pointing to potential ambiguities in the feature space that could benefit from further refinement.

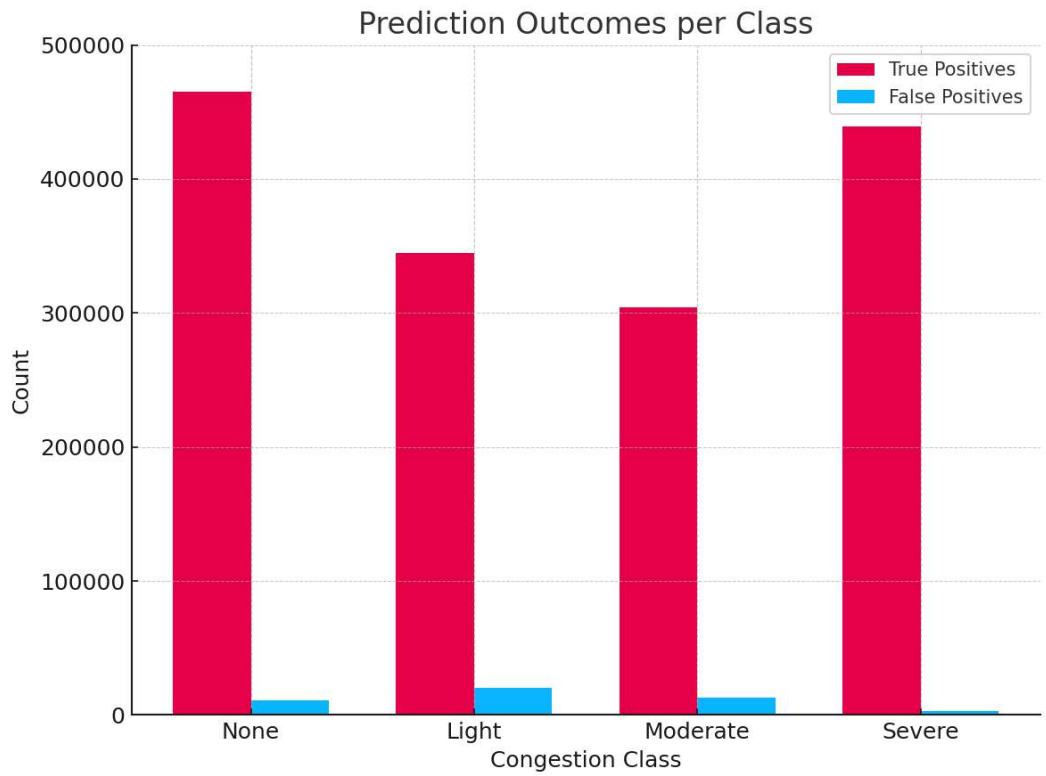

Figure 4.4: True positives vs. false positives per class.

Figure 4.4 compares the true positives and false positives across each congestion class, emphasizing the model's predictive reliability. The chart highlights a clear dominance of true positives across all classes, indicating a strong overall performance in correctly classifying congestion states. False positives are more pronounced for the Light and Moderate classes, underscoring the model's difficulty in distinguishing these overlapping categories, which aligns with real-world traffic behavior where such states can be ambiguous. In contrast, the lower false positive rates for None and Severe reflect their distinct traffic patterns, reinforcing the model's confidence in these classifications.

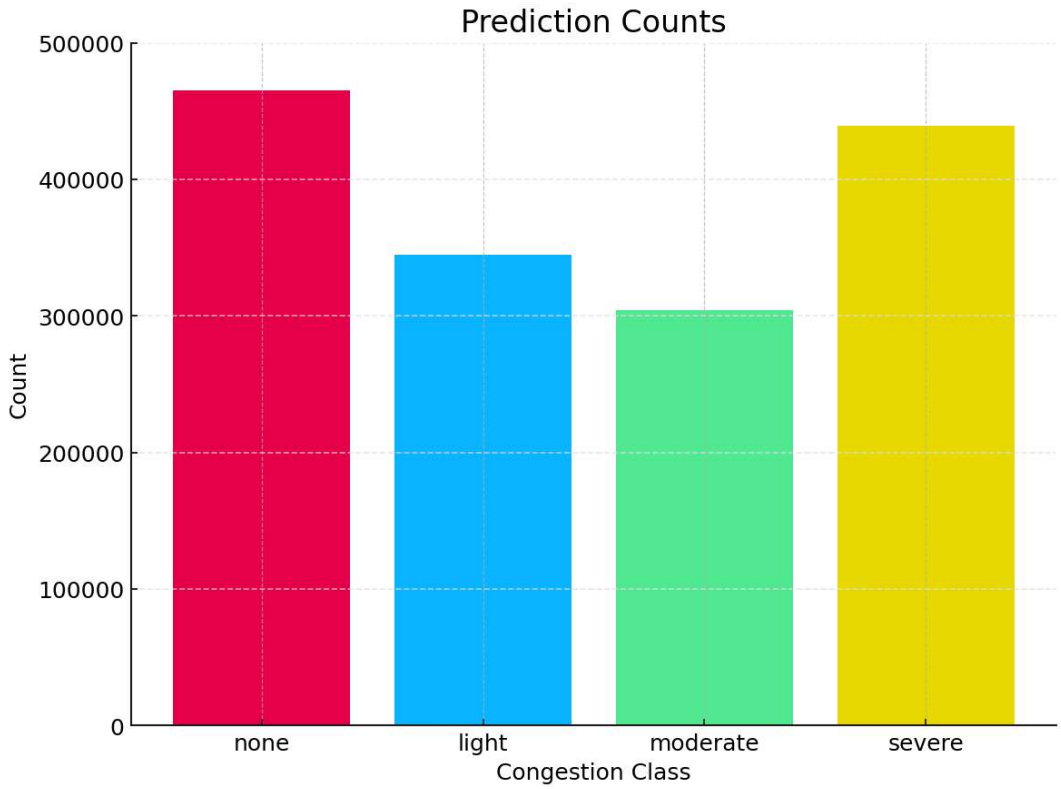

Figure 4.5: Distribution of predicted classes in test data.

Figure 4.5 showcases the frequency distribution of predicted classes on the test set, highlighting the model's ability to maintain a balanced representation across all congestion levels. The distribution closely mirrors the underlying class proportions, suggesting that the model avoids overfitting or developing biases toward any single class. This balance is crucial for real-time congestion detection, ensuring that alerts are neither overly frequent nor missed. The consistency across classes further validates the model's robustness, though the overlap between Light and Moderate hints at areas where decision boundaries could be sharpened.

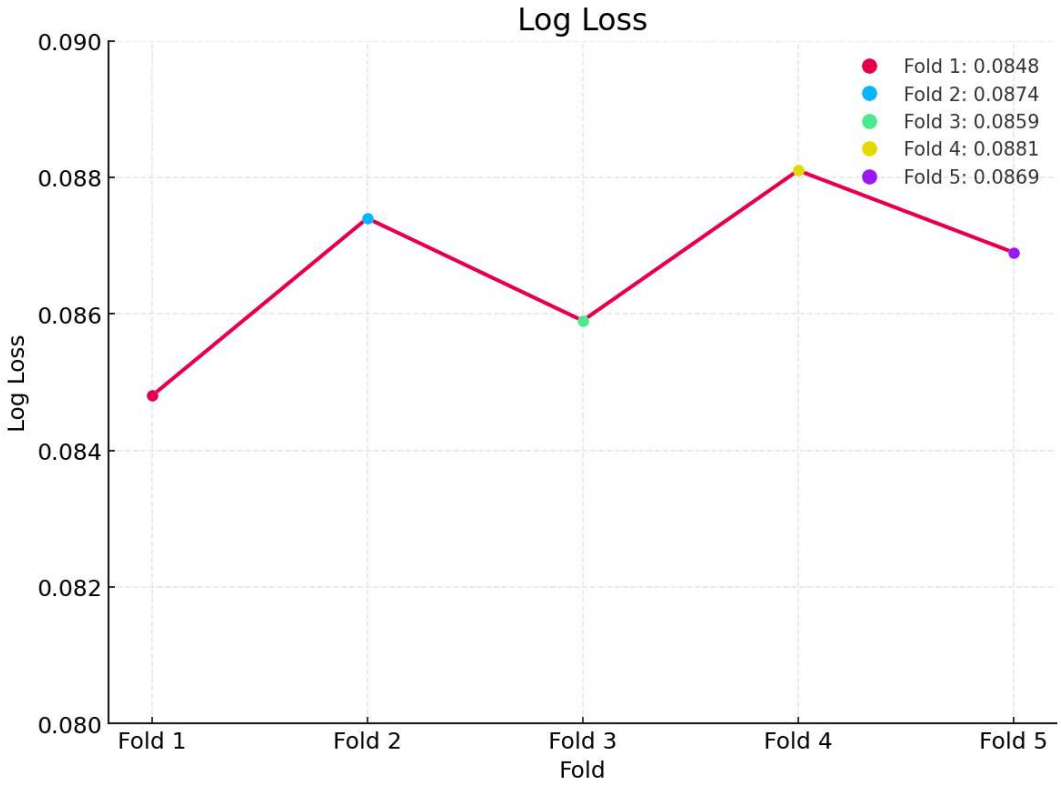

Figure 4.6: Cross-validation log loss per fold.

Figure 4.6 shows the log loss obtained across 5 cross-validation folds. Log loss—also known as logarithmic loss or cross-entropy loss—is a metric that quantifies the accuracy of probabilistic predictions by penalizing incorrect classifications with high confidence. It is particularly informative in multi-class classification tasks, as it captures not only whether the predicted class is correct, but also how confident the model is in its predictions.

Lower log loss values indicate better calibrated models that assign higher probabilities to the correct class. A perfectly confident and accurate classifier would yield a log loss of 0. In this context, the consistently low log loss across all folds suggests that the DTIP model not only makes accurate predictions but also produces reliable confidence scores, which is essential for downstream applications such as traffic alerting or probabilistic decision-making.

It should be noted, however, that cross-validation performance was not the main focus of this study and mainly served as a comparison method rather than a decisive evaluation criterion.

4.2.6.1 Time-Series Model Evaluation

To address the limitation of temporal independence in the main XGBoost model, a time-series variant of the DTIP framework was implemented as an experimental extension.

This variant employs a sliding window approach, incorporating historical features over 5 time steps with a prediction horizon of 1 time step. The model, based on XGBoost, expands the feature set to include temporal sequences of the original engineered features (e.g., `probe_speed_t-5`, `probe_speed_t-4`, ..., `probe_speed_t-1`), resulting in up to 70 features compared to the 14 features in the static model.

The time-series model was evaluated using the same dataset and methodology as the static model, with results reported for a 70-30 train/test split.

Figure 4.7 visualizes the per-class precision, recall, and F1-score, showing consistent performance across classes, with the highest scores for `free-flow` and `severe` congestion, similar to the static model.

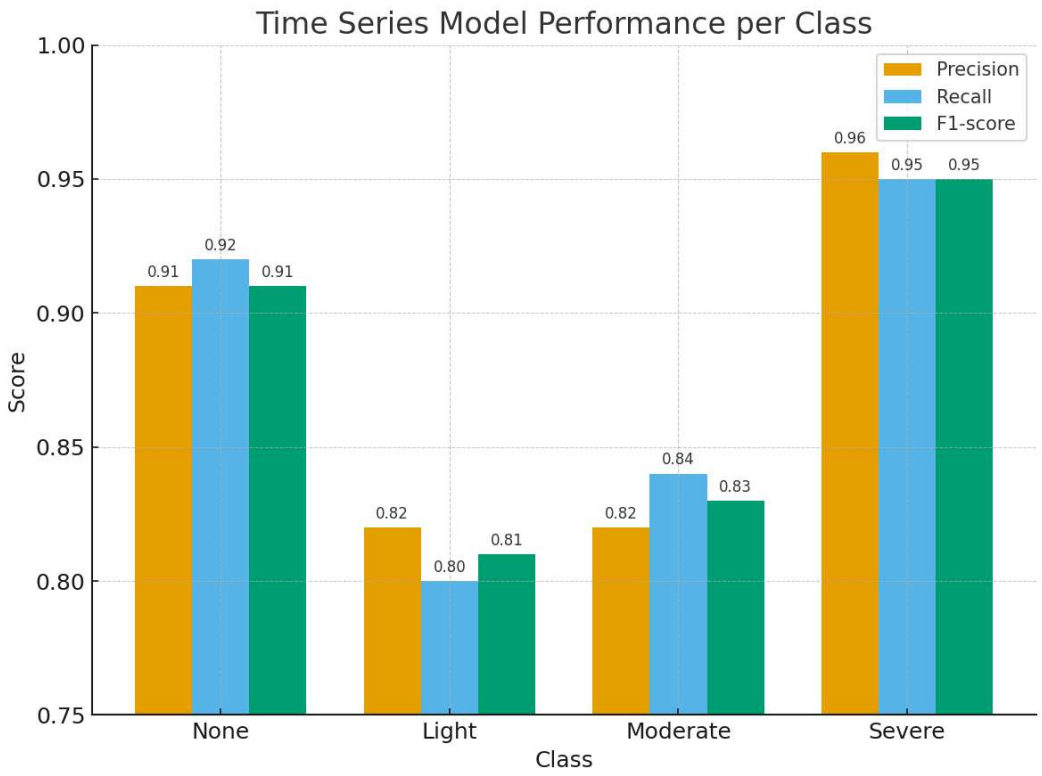

Figure 4.7: Per-class precision, recall, and F1-score for the time-series model.

Table 4.2 compares the performance of the static XGBoost model and the time-series variant:

Metric	Static XGBoost	Time-Series Model
Accuracy	97.06%	88.54%
Weighted F1-Score	97.06%	88.55%
Log Loss	0.0756	0.3039

Table 4.2: Comparison of static XGBoost and time-series model performance.

Compared to the static XGBoost model, which achieved a weighted F1-score of **0.9706**

and a log loss of **0.0756**, the time-series model exhibits lower performance across all metrics. Several factors contribute to this performance gap. First, the expanded feature space (up to 70 features) increases model complexity, raising the risk of overfitting, especially given the limited hyperparameter tuning due to computational constraints. Second, the static model's engineered features (e.g., stop durations, speed bins, and road class) already capture significant congestion patterns, making the additional temporal context less critical for classification tasks. Third, the dataset may lack sufficient temporal variability (e.g., low density of night-time data, as shown in Chapter 3) to fully leverage sequential features.

Despite the lower performance, the time-series model demonstrates strong results, particularly for the `moderate` (F1-score: 0.83) and `severe` (F1-score: 0.95) congestion classes. The higher log loss (**0.3039** vs. **0.0756**) suggests that the time-series model is less confident in its predictions, likely due to the increased feature dimensionality and noise introduced by temporal sequences.

This experimental model validates the feasibility of incorporating temporal dynamics into DTIP, particularly for scenarios requiring sequential consistency or short-term forecasting. For example, the sliding window approach could improve detection of congestion build-up or transitions, which are critical for proactive traffic management. However, the increased computational cost—approximately 3x higher memory usage and 2x longer training time compared to the static model—highlights the need for optimization. Due to computational constraints, deeper hyperparameter tuning, larger window sizes, or alternative time-series architectures (e.g., LSTM or Temporal Convolutional Networks) were not explored. These results suggest that while temporal modeling is promising, the static XGBoost model remains more suitable for real-time deployment due to its superior performance and scalability.

Future work could focus on optimizing the time-series model by reducing feature dimensionality (e.g., via feature selection), exploring hybrid static-temporal architectures, or incorporating more diverse temporal data to enhance the model's ability to capture dynamic traffic patterns.

4.3 SUMO-Based Qualitative Evaluation

To enhance the quantitative analysis, a qualitative evaluation of the DTIP model was conducted using the Simulation of Urban Mobility (SUMO) traffic simulator (Eclipse SUMO, 2025b). While not part of DTIP's core inference pipeline, SUMO provides a visual way to verify if the model's congestion predictions align with realistic traffic behavior, making it easier to interpret results for stakeholders.

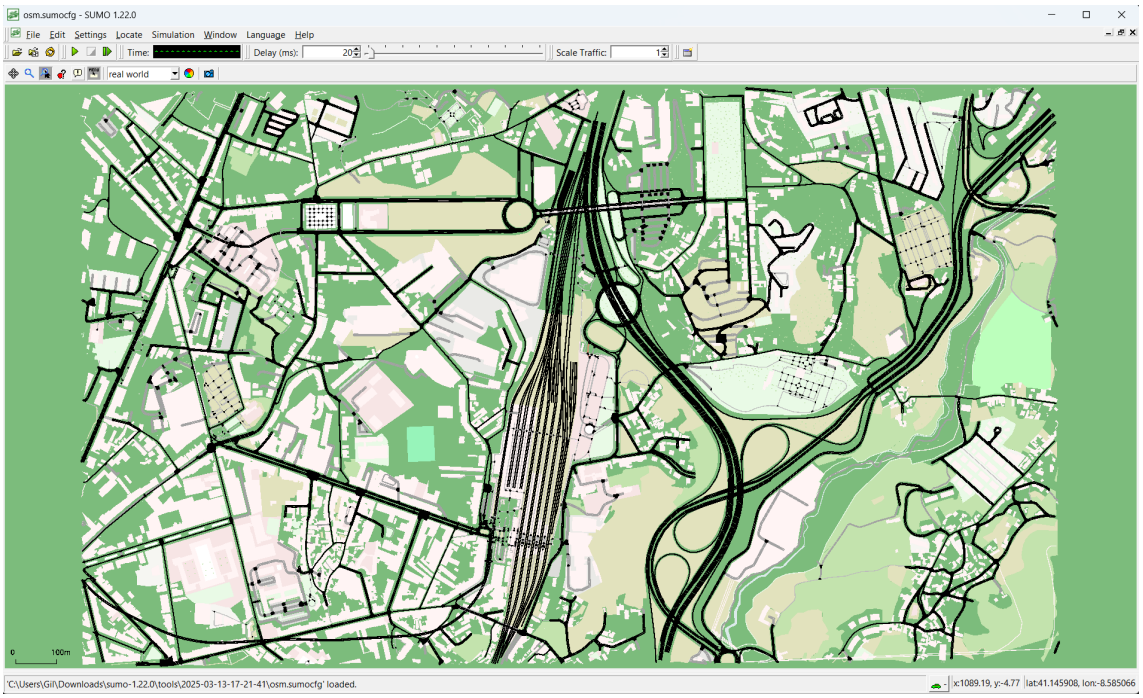

Figure 4.8: Screenshot of the SUMO-GUI interface (Eclipse SUMO, 2025a).

4.3.1 Simulation Setup

The evaluation used a custom pipeline built in Python, integrating Simulation of Urban Mobility (SUMO) for traffic simulation, TraCI (Eclipse SUMO, 2025c) for real-time control, and libraries like pandas and Dask for data processing. The road network was created from OpenStreetMap (OSM) data, specifically representing the urban area of Vila Nova de Gaia, Portugal albeit a small part of it due to processing constraints. This region was selected using the SUMO Web Wizard, an online tool provided by the SUMO project that simplifies the process of generating simulation scenarios by allowing users to select a geographic area, configure traffic parameters, and export OSM-based road networks and routes directly from a web interface (Eclipse SUMO, 2025d). The OSM data for Vila Nova de Gaia was exported via the SUMO Web Wizard and converted into a SUMO-compatible network file (.net.xml) using the `netconvert` tool.

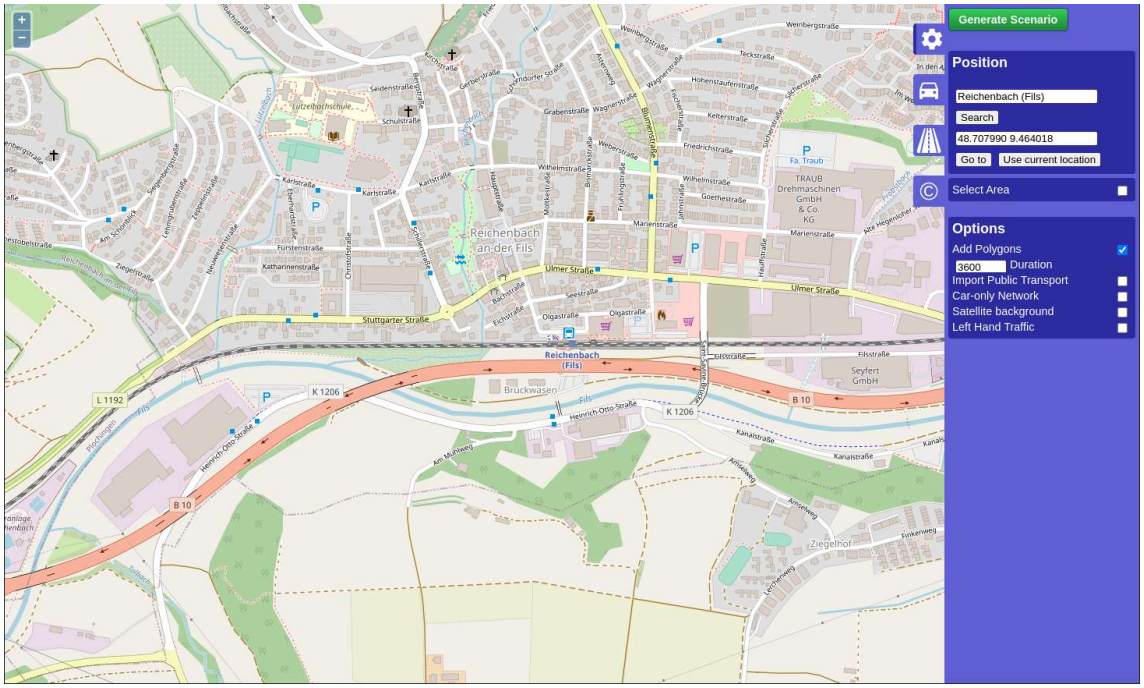

Figure 4.9: Screenshot of the SUMO Web Wizard interface (Eclipse SUMO, 2025d).

Vehicle trips were generated using SUMO's included `randomTrips.py` script with the following parameters:

- **Vehicle insertion period**: 1 second, to simulate dense traffic.
- **Fringe factor**: 5, to prioritize trips starting or ending at network edges for realism.
- **Simulation duration**: 1 hour (3600 seconds).

The simulation start time was randomly set within a predefined range (e.g., starting at 8 AM to mimic peak hours). Accidents were introduced randomly with a 1% probability per simulation step, lasting between 5 and 10 minutes. These accidents were simulated by stopping a random vehicle on its current road segment, setting its speed to 0 km/h using TraCI (Eclipse SUMO, 2025c).

FCD, including vehicle positions and speeds, was extracted from SUMO's XML output and converted into a Parquet DataFrame. The data was filtered to a specific time range and map-matched to road segments using a local Valhalla server (the same one used in the DTIP pipeline). The processed data was then used by DTIP's LightGBM-based congestion model to predict congestion levels (e.g., free-flow, light, moderate, severe) based on features like average speed, stopped time, day of week, and road type.

4.3.2 Testing Methodology

The pipeline was orchestrated through a main Python script which managed simulation, data parsing, map matching, model prediction, and visualization. Multiple simulations were run with different random seeds to ensure robust results. For each accident, the pipeline extracted relevant features from FCD, combined them with historical road data (e.g., segment length, road class), and generated congestion predictions. These predictions were visualized in SUMO's graphical interface (`sumo-gui`) (Eclipse SUMO, 2025a), where congested road segments were highlighted in green, and stopped vehicles were marked in red. Visualizations started 60 seconds before each accident to capture the build-up of traffic conditions.

Parameters like accident probability and duration were adjusted empirically to create realistic congestion without overloading the network. While no formal grid search was conducted, the chosen settings produced plausible traffic patterns for qualitative analysis.

4.3.3 Findings

The qualitative results were promising. In most test scenarios, DTIP's congestion predictions matched the observed traffic behavior in SUMO. For example:

- **Severe congestion** (`SRI` ≥ 0.70) showed long queues and stop-and-go traffic, as expected.
- **Free-flow** segments displayed smooth vehicle movement at normal speeds.
- Transitions from light to moderate congestion appeared as denser traffic and slower acceleration, aligning with DTIP's predictions.

These visual outcomes confirm that DTIP's predictions are not only statistically accurate but also reflect realistic traffic dynamics, making them easier to interpret for non-technical audiences.

4.3.4 Limitations

Although the SUMO-based evaluation provides valuable insights into validating the plausibility of DTIP's predictions, it is subject to several limitations that must be acknowledged. Firstly, the simulation was conducted within a geographically constrained area, specifically a small extracted portion of Gaia. This limited spatial scope implies that the results do not capture the complexity of traffic flows on a broader scale, such as interactions between different regions or the influence of adjacent areas, thereby reducing the representativeness of the findings.

Moreover, the testing was not exhaustive across various times of day, traffic volumes, or road types. For instance, the simulation did not account for significant variations such

as peak hours, nighttime conditions, or differences between weekdays and weekends, which limits the generalizability of the results. Additionally, the SUMO simulations did not incorporate critical real-world factors, including pedestrian crossings, traffic signals, or weather conditions. While SUMO is capable of modeling such elements with appropriate configurations (e.g., defining traffic lights, pedestrian behavior, or adjusting parameters to simulate adverse weather), these were not included in the evaluation.

Consequently, although the SUMO-based simulations validate the feasibility of DTIP's predictions to a certain extent, they are not a comprehensive substitute for real-world testing. To enhance the robustness of future evaluations, simulations should encompass a wider range of scenarios and incorporate additional real-world variables to better reflect the complexities of actual traffic systems.

4.3.5 Observations

- Segments labeled as **severe** (`SRI` > 0.70) showed long queues and heavy traffic buildup, consistent with high congestion.
- **Free-flow** segments had smooth vehicle flow at expected speeds, supporting low `SRI` predictions.
- Transitions from light to moderate congestion were visible as reduced acceleration and denser traffic, indicating consistent prediction logic.

Although not a replacement for real-world validation, the SUMO-based evaluation strengthens confidence in DTIP's predictions and enhances interpretability, particularly for stakeholder communication and visual debugging.

4.4 Discussion

4.4.1 Implications of the Quantitative Results

The high accuracy and F1-score achieved by the XGBoost model (97%+) across both the test split and cross-validation confirm that DTIP's feature engineering pipeline captures meaningful patterns in urban mobility data. In particular, the ability to distinguish well between *free-flow* and *severe congestion* suggests that the core input features—such as average segment speed, stop durations, and temporal indicators—are highly informative when fed into gradient boosting classifiers.

This performance is especially relevant in practical deployments where false positives (e.g., incorrectly classifying free flow as congestion) can lead to unnecessary interventions, and false negatives (e.g., failing to detect congestion) may result in undetected haz-

ards. The classifier's robustness across validation folds further supports its generalization ability, despite being trained on data from a single city.

4.4.2 Relevance of SUMO-Based Qualitative Validation

The SUMO simulations, although qualitative and limited in scope, provided an important visual confirmation of the model's predictions. Their value lies not in precision but in interpretability: being able to observe realistic stop-and-go dynamics, vehicle accumulation, or uninterrupted flow reinforces stakeholder trust in the model's logic. In public sector applications—where explainability is key to adoption—this visual interpretability layer becomes a powerful communication asset.

Moreover, the congruence between DTIP predictions and SUMO's emergent dynamics suggests that the decision boundaries learned by the model (e.g., via SRI thresholds) are behaviorally grounded, and not just statistical artifacts.

4.4.3 Key Methodological Trade-Offs

The choice to use interpretable and scalable components—such as XGBoost instead of deep neural networks, or FCD instead of video data—reflects a conscious trade-off between complexity and deployability. While more complex models (e.g., LSTM, spatio-temporal GNNs) may marginally improve performance, their cost in terms of explainability, training time, and hardware requirements is non-trivial.

Similarly, deriving labels from SRI instead of human annotations introduces some approximation but allows for large-scale label generation in unlabeled datasets—an approach more compatible with real-world deployment pipelines.

4.4.4 Scalability and Generalizability

DTIP's performance on data from Vila Nova de Gaia showcases its potential, but its generalizability to other urban contexts remains untested. While the pipeline was designed to be modular and city-agnostic, performance may vary depending on road structure, driver behavior, or FCD density in other cities. Further experiments across diverse urban areas—ideally with differing traffic cultures and FCD penetration—would provide deeper insights into the pipeline's portability.

On the scalability front, the integration of `Dask` and `Parquet+PyArrow` proved essential in handling over 18 million probes efficiently on commodity hardware. This suggests that DTIP can scale to national-level deployments, particularly when combined with cloud-based execution.

4.4.5 Limitations to Address

Despite its strengths, the current evaluation presents several limitations:

- **Absence of ground-truth labels:** All model evaluation is derived from SRI-based labels. Although practical, this weakens claims of external validity. A future study with human-labeled or sensor-validated congestion ground truth would be ideal.
- **Temporal independence:** The classifier treats each segment observation independently. Incorporating temporal sequence modeling could improve consistency across time windows—especially for short-term forecasting or proactive alerting.
- **Simplified simulation logic:** SUMO scenarios are derived manually and do not reflect a full distribution of real-world complexities such as weather, pedestrian crossings, or traffic light dynamics. More systematic simulation studies could strengthen model confidence.
- **Limited class interpretability:** The boundaries between *light* and *moderate* congestion are inherently fuzzy. Future work could explore probabilistic labeling, regression targets, or context-aware classification thresholds.

4.4.6 Positioning within the Literature

Compared to previous congestion detection studies that either rely on black-box methods such as deep learning or on infrastructure-heavy setups like inductive loop detectors, DTIP strikes a balanced middle ground. Traditional fixed-sensor approaches, while precise, suffer from high deployment and maintenance costs and limited adaptability, as highlighted by (Kong et al., 2016). On the other end of the spectrum, several studies explored deep learning architectures such as LSTMs or spatio-temporal networks, but found them to be less interpretable and more resource-intensive, often unsuitable for real-world deployments under public-sector constraints (Peruthambi et al., 2025; Pandove and Pandove, 2024).

DTIP instead leverages FCD in combination with interpretable ensemble methods, particularly gradient boosting. As shown in recent surveys and applications, FCD offers scalability and geographic flexibility with minimal infrastructure requirements, provided robust preprocessing and map-matching techniques are applied (Huang et al., 2021; Li et al., 2021). By adopting XGBoost, DTIP balances accuracy and interpretability while remaining computationally efficient, making it more reproducible and adaptable to urban contexts than purely black-box or infrastructure-intensive alternatives.

4.4.7 Future Directions

Several promising extensions emerge from this work:

- Incorporate spatial dependencies (e.g., congestion propagation across adjacent segments) via graph-based models.
- Integrate hazard-type classification for richer multi-task learning (e.g., distinguishing between congestion, accident, or roadwork).
- Couple DTIP with crowdsourced incident reports (e.g., Waze) to explore hybrid validation systems.
- Use the SUMO layer for counterfactual testing and simulation-informed policy planning.

4.5 Conclusion

The evaluation results confirm that DTIP effectively classifies congestion levels with high precision and recall. The XGBoost model emerged as an effective approach, offering both accuracy and interpretability. All quantitative metrics support the reliability of DTIP's predictive capabilities.

In parallel, the SUMO-based simulation layer provided additional interpretability by mapping DTIP's congestion classifications into dynamic traffic scenarios. This qualitative layer proved especially helpful in verifying that predicted traffic states manifested as expected under realistic simulation conditions. While not exhaustive or quantitative, these simulations offer a promising avenue for visual validation and stakeholder communication.

Nonetheless, some limitations remain. The evaluation is confined to a single city, does not include ground-truth human-labeled congestion data, and lacks simulation-based performance metrics. Future work should consider validating the model on labeled datasets or sensor data, and expand the SUMO evaluation into more systematic scenario testing.

In summary, DTIP demonstrates strong performance both quantitatively and qualitatively. Its modular architecture and transparent predictions lay the groundwork for scalable, interpretable, and city-agnostic congestion monitoring—ready for both scientific inquiry and practical deployment.

Chapter 5

Conclusion

5.1 Summary of the Work

This thesis addressed the pressing challenge of urban traffic congestion detection by developing the Distributed Traffic Intelligence Pipeline (DTIP), a modular and scalable system based on Floating Car Data (FCD) and interpretable Machine Learning (ML). The proposed framework is capable of estimating congestion levels across urban road networks, supporting both real-time classification and post-hoc validation of traffic events.

Distributed Traffic Intelligence Pipeline (DTIP) was designed with five key principles: modularity, interpretability, scalability, privacy preservation, and geographic portability. The pipeline includes distinct layers for preprocessing, map-matching, feature engineering, and congestion classification supplemented by a qualitative simulation module using the SUMO traffic simulator.

The case study of Vila Nova de Gaia served as a representative and data-rich urban context, enabling thorough testing and validation of the system under realistic conditions. Through rigorous preprocessing and the use of a handcrafted feature set, the system successfully predicted congestion severity across four classes derived from the Speed Reduction Index (SRI).

5.2 Key Findings

The evaluation results provide a comprehensive picture of the performance and scope of DTIP, combining quantitative metrics with qualitative evidence from simulation.

5.2.1 Quantitative Performance

The congestion classification model based on Extreme Gradient Boosting (XGBoost) consistently delivered high predictive accuracy across both the hold-out test set and cross-validation folds. With an overall accuracy and weighted F1-score of approximately 97%,

the model proved capable of distinguishing congestion states with remarkable reliability. Performance was strongest for the extreme classes—free-flow and severe congestion—where traffic dynamics are more clearly defined. As expected, the boundary between light and moderate congestion produced the majority of misclassifications, reflecting the inherent ambiguity of these intermediate states rather than a structural weakness of the model.

Beyond raw metrics, the stability of results across folds demonstrated that the pipeline generalizes well to unseen trajectories, reinforcing the robustness of the feature engineering strategy and the reliability of the chosen algorithm.

5.2.2 Qualitative Validation

To complement the statistical evaluation, a qualitative validation step was conducted using SUMO. This simulation layer allowed the behavioral plausibility of the model's predictions to be assessed in controlled traffic scenarios. The outcomes confirmed that DTIP's classifications translated into realistic traffic patterns: segments flagged as severe congestion (SRI > 0.70) displayed long queues and stop-and-go dynamics; free-flow segments (SRI < 0.15) reproduced smooth, uninterrupted movement; and transitions between light and moderate congestion manifested as gradual flow degradation.

While no additional quantitative indicators were extracted from the simulator, the visual consistency between predicted congestion states and emergent traffic behavior offered a valuable layer of interpretability. This dual perspective—statistical performance on real-world data and behavioral plausibility in simulated environments—strengthens confidence in the applicability of the pipeline.

5.2.3 Synthesis

Taken together, these findings demonstrate that DTIP achieves a rare balance: a scalable and interpretable pipeline that attains high quantitative accuracy while producing outputs that remain consistent with real-world traffic dynamics. The results validate the methodological choices made, particularly the reliance on FCD, robust map-matching, handcrafted features, and gradient boosting, all of which contributed to a transparent and reliable congestion detection system.

5.3 Strengths and Contributions

This work introduced a set of novel contributions:

- A modular and fully reproducible pipeline for congestion classification from FCD using interpretable ML models.

- High performance using XGBoost, avoiding the complexity of deep learning without sacrificing accuracy.
- Strategic use of SUMO for qualitative interpretability and stakeholder communication.
- A complete deployment-ready architecture using open-source tools like Valhalla, Dask, and Parquet.
- Detailed methodology for deriving congestion labels via SRI, adaptable to different urban settings.

5.4 Limitations

While DTIP demonstrates significant strengths, it faces certain constraints that warrant consideration. The system's reliance on Floating Car Data (FCD) means its performance is tied to the density of GPS-equipped vehicles, leading to less reliable congestion estimates in areas with sparse probe data, such as rural zones or underrepresented urban neighborhoods. Additionally, the current pipeline is optimized for offline batch processing, and its ability to handle real-time constraints remains untested, potentially limiting its applicability in dynamic operational settings. The use of fixed SRI thresholds for labeling congestion levels, while effective, lacks adaptability to varying road types or temporal conditions, which can result in misclassifications, particularly between light and moderate congestion states. Furthermore, the model treats road segments independently, without accounting for temporal dependencies or trajectory histories, which could otherwise enhance predictions in dynamic environments like intersections. The absence of human-annotated ground-truth data also poses a challenge, as labels derived solely from SRI thresholds lack external validation. Similarly, the SUMO simulations, while useful for qualitative insights, were limited in scope and lacked quantitative comparisons to real-world traffic conditions. Finally, although designed for portability, DTIP has only been evaluated in Vila Nova de Gaia, leaving its generalizability to other urban contexts unverified.

5.5 Future Work

Building on this foundation, several avenues can enhance DTIP's capabilities. To address the reliance on FCD, integrating complementary data sources, such as fixed sensors or crowdsourced alerts, could improve coverage in areas with low GPS probe density. Transitioning to real-time deployment through streaming data ingestion and sliding window computations would enable near-real-time analytics, making the system more

suitable for operational use. Replacing fixed SRI thresholds with dynamic or adaptive labeling strategies, informed by real-time or historical baselines, could improve classification granularity and reduce edge-case errors. Incorporating trajectory-level modeling, such as temporal ensembles or attention mechanisms, would capture dependencies across consecutive time windows, particularly in dynamic settings like intersections. Validating predictions against labeled datasets from traffic cameras or induction loops would provide a robust measure of real-world accuracy. Expanding SUMO's role beyond qualitative validation to include synthetic data generation and scenario testing could further refine the system. Finally, deploying DTIP in diverse urban contexts would test its portability and generalization, ensuring its applicability across different driving cultures and infrastructure layouts.

5.6 Final Remarks

This thesis provides a concrete, open, and scalable solution to the challenge of congestion detection in urban mobility. DTIP was validated using FCD from Vila Nova de Gaia, showing that a modular and interpretable pipeline can achieve high accuracy and transparency in classifying congestion levels. The results suggest that, under similar data availability and urban conditions, the proposed approach can serve as a reliable basis for traffic monitoring and hazard validation. While promising, these findings should be interpreted within the scope of the analyzed case study and comparable urban contexts, rather than generalized to all transportation scenarios.

Bibliography

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2575–2582. IEEE, November 2018. URL https://elib.dlr.de/127994/. 4

Mostafa Amin-Naseri, Pranamesh Chakraborty, Anuj Sharma, Stephen B. Gilbert, and Mingyi Hong. Evaluating the Reliability, Coverage, and Added Value of Crowdsourced Traffic Incident Reports from Waze. 2672:34–43, December 2018. ISSN 0361-1981, 2169-4052. doi: 10.1177/0361198118790619. URL https://journals.sagepub.com/doi/10.1177/0361198118790619. 10, 11

Andrew Antonopoulos. Improve Machine Learning carbon footprint using Parquet dataset format and Mixed Precision training for regression models – Part II, September 2024. URL http://arxiv.org/abs/2409.11071. 14, 19

Kristian Breili and Carl William Lund. Simulation of GNSS Dilution of Precision for Automated Mobility Along the MODI Project Road Corridor Using High-Resolution Digital Surface Models. *Geomatics*, 5(2):26, June 2025. ISSN 2673-7418. doi: 10.3390/geomatics5020026. URL https://www.mdpi.com/2673-7418/5/2/26. Publisher: Multidisciplinary Digital Publishing Institute. 24

Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, August 2016. doi: 10.1145/2939672.2939785. URL http://arxiv.org/abs/1603.02754. arXiv:1603.02754 [cs]. 19

Eclipse SUMO. SUMO-GUI Documentation. https://sumo.dlr.de/docs/sumo-gui.html, 2025a. Accessed September 2025. viii, 48, 50

Eclipse SUMO. SUMO: Simulation of Urban Mobility Documentation. https://sumo.dlr.de/docs/, 2025b. Accessed September 2025. 18, 47

Eclipse SUMO. TraCI: Traffic Control Interface Documentation. https://sumo.dlr.de/docs/TraCI.html, 2025c. Accessed September 2025. 48, 49

Eclipse SUMO. OSM Web Wizard Documentation. https://sumo.dlr.de/docs/Tools/SumoWebWizard.html, 2025d. Accessed September 2025. viii, 48, 49

Tomislav Erdelić, Tonči Carić, Martina Erdelić, Leo Tišljarić, Ana Turković, and Niko Jelušić. Estimating congestion zones and travel time indexes based on the floating car data. *Computers, Environment and Urban Systems*, 87:101604, May 2021. ISSN 0198-9715. doi: 10.1016/j.compenvurbsys.2021.101604. URL https://www.sciencedirect.com/science/article/pii/S0198971521000119. 4

Hamdy B. Faheem, Amira M. El Shorbagy, and Mohamed Elsayed Gabr. Impact Of Traffic Congestion on Transportation System: Challenges and Remediations - A review. *Mansoura Engineering Journal*, 49, 2024. ISSN 2735-4202. doi: 10.58491/2735-4202.3191. URL https://mej.researchcommons.org/home/vol49/iss2/18. 1

Paulo Fernandez, Sandra Mourato, and Madalena Moreira. Social vulnerability assessment of flood risk using GIS-based multicriteria decision analysis. A case study of Vila Nova de Gaia (Portugal). 7:1367–1389, July 2016. ISSN 1947-5705, 1947-5713. doi: 10.1080/19475705.2015.1052021. URL http://www.tandfonline.com/doi/full/10.1080/19475705.2015.1052021. 16

Iris Shmuel Gal Moran and Daniel Marcous. How Waze Uses TFX to Scale Production-Ready ML, September 2021. URL https://blog.tensorflow.org/2021/09/how-waze-uses-tfx-to-scale-production-ready-ml.html. 1

Google. Google maps platform - directions api, 2024. URL https://developers.google.com/maps/documentation/directions. Accessed July 2025. 12

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735. 19

Zhenfeng Huang, Shaojie Qiao, Nan Han, Chang-an Yuan, Xuejiang Song, and Yueqiang Xiao. Survey on vehicle map matching techniques. 6:55–71, 2021. ISSN 2468-2322. doi: 10.1049/cit2.12030. URL https://onlinelibrary.wiley.com/doi/abs/10.1049/cit2.12030. 4, 8, 11, 13, 53

Kitae Kim, Soohyun Cho, and Woojin Chung. HD Map Update for Autonomous Driving With Crowdsourced Data. 6:1895–1901, April 2021. ISSN 2377-3766. doi: 10.1109/LRA.2021.3060406. URL https://ieeexplore.ieee.org/document/9357917/. 10

Hannes Koller, Peter Widhalm, Melitta Dragaschnig, and Anita Graser. Fast Hidden Markov Model Map-Matching for Sparse and Noisy Trajectories. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2557–2561,

September 2015. doi: 10.1109/ITSC.2015.411. URL https://ieeexplore.ieee.org/document/7313503. ISSN: 2153-0017. 13

Xiangjie Kong, Zhenzhen Xu, Guojiang Shen, Jinzhong Wang, Qiuyuan Yang, and Benshi Zhang. Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Generation Computer Systems*, 61:97–107, August 2016. ISSN 0167-739X. doi: 10.1016/j.future.2015.11.013. URL https://www.sciencedirect.com/science/article/pii/S0167739X15003611. 1, 2, 8, 11, 19, 53

Johann Lau. Google Maps 101: How AI helps predict traffic and determine routes, September 2020. URL https://blog.google/products/maps/google-maps-101-how-ai-helps-predict-traffic-and-determine-routes/. 1

Lee Leong, Azai Azmalia, Goh Cott, and Shafida Azwina Mohd Shafie. Development and assessment of free-flow speed models based on different methods of measurements for inter urban multilane highways in Malaysia. *Journal of Applied Engineering Science*, 17:256–263, January 2019. doi: 10.5937/jaes17-21205. 26

Jonathan I. Levy, Jonathan J. Buonocore, and Katherine von Stackelberg. Evaluation of the public health impacts of traffic congestion: a health risk assessment. 9:65, October 2010. ISSN 1476-069X. doi: 10.1186/1476-069X-9-65. URL https://doi.org/10.1186/1476-069X-9-65. 1

Jinjian Li, Jacques Boonaert, Arnaud Doniec, and Guillaume Lozenguez. Multi-models machine learning methods for traffic flow estimation from Floating Car Data. *Transportation Research Part C: Emerging Technologies*, 132:103389, November 2021. ISSN 0968-090X. doi: 10.1016/j.trc.2021.103389. URL https://www.sciencedirect.com/science/article/pii/S0968090X21003867. 8, 11, 53

Yunduan Lin and Ruimin Li. Real-time traffic accidents post-impact prediction: Based on crowdsourcing data. *Accident Analysis & Prevention*, 145:105696, September 2020. ISSN 0001-4575. doi: 10.1016/j.aap.2020.105696. URL https://www.sciencedirect.com/science/article/pii/S0001457520305807. 10, 11

Pablo Alvarez Lopez, Evamarie Wiessner, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flotterod, Robert Hilbrich, Leonhard Lucken, Johannes Rummel, and Peter Wagner. Microscopic Traffic Simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582, Maui, HI, November 2018. IEEE. doi: 10.1109/itsc.2018.8569938. URL https://ieeexplore.ieee.org/document/8569938/. 18, 35

Mapbox. Mapbox map matching api, 2024. URL https://docs.mapbox.com/api/navigation/map-matching/. Accessed July 2025. 12

Hung T. Nguyen, Vladik Kreinovich, and Lakshmi N. Yamparala. Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. Technical Report UTEP-CS-18-09, Department of Computer Science, University of Texas at El Paso, 2018. URL https://www.cs.utep.edu/vladik/2018/tr18-09.pdf. 32

Gitanjali Pandove and Deepika Pandove. Enhancing urban mobility: predicting traffic congestion with optimized ML model. 6:045242, November 2024. ISSN 2631-8695. doi: 10.1088/2631-8695/ad8dbd. URL https://dx.doi.org/10.1088/2631-8695/ad8dbd. 18, 19, 35, 53

Venkatesh Peruthambi, Lahari Pandiri, Pallav Kumar Kaulwar, Hara Krishna Reddy Koppolu, Balaji Adusupalli, and Avinash Pamisetty. Big Data-Driven Predictive Maintenance for Industrial IoT (IIoT) Systems. 31:21–30, March 2025. ISSN 2812-9105. doi: 10.63278/1316. URL https://metall-mater-eng.com/index.php/home/article/view/1316. 19, 53

Matthew Rocklin. Dask: Parallel Computation with Blocked algorithms and Task Scheduling. June 2015. doi: 10.25080/Majora-7b98e3ed-013. URL https://proceedings.scipy.org/articles/Majora-7b98e3ed-013. 19

Siavash Saki and Tobias Hagen. A Practical Guide to an Open-Source Map-Matching Approach for Big GPS Data. 3:415, August 2022. ISSN 2661-8907. doi: 10.1007/s42979-022-01340-5. URL https://doi.org/10.1007/s42979-022-01340-5. 13, 19

Ravid Shwartz-Ziv and Amitai Armon. Tabular Data: Deep Learning is Not All You Need, November 2021. URL http://arxiv.org/abs/2106.03253. arXiv:2106.03253 [cs]. 32, 37

Ary P. Silvano, Haris N. Koutsopoulos, and Haneen Farah. Free flow speed estimation: A probabilistic, latent approach. Impact of speed limit changes and road characteristics. *Transportation Research Part A: Policy and Practice*, 138:283–298, August 2020. ISSN 0965-8564. doi: 10.1016/j.tra.2020.05.024. URL https://www.sciencedirect.com/science/article/pii/S0965856420306066. 26

Nemanja Stepanović, Vladan Tubić, and Stefan Zdravković. Determining Free-Flow Speed on Different Classes of Rural Two-Lane Highways. *Promet - Traffic&Transportation*, 35:315–330, June 2023. doi: 10.7307/ptt.v35i3.195. 27

Valhalla Team. Valhalla: Open source routing engine, 2025. URL https://valhalla.github.io/valhalla/. Acessado em 30 de julho de 2025. 4, 13