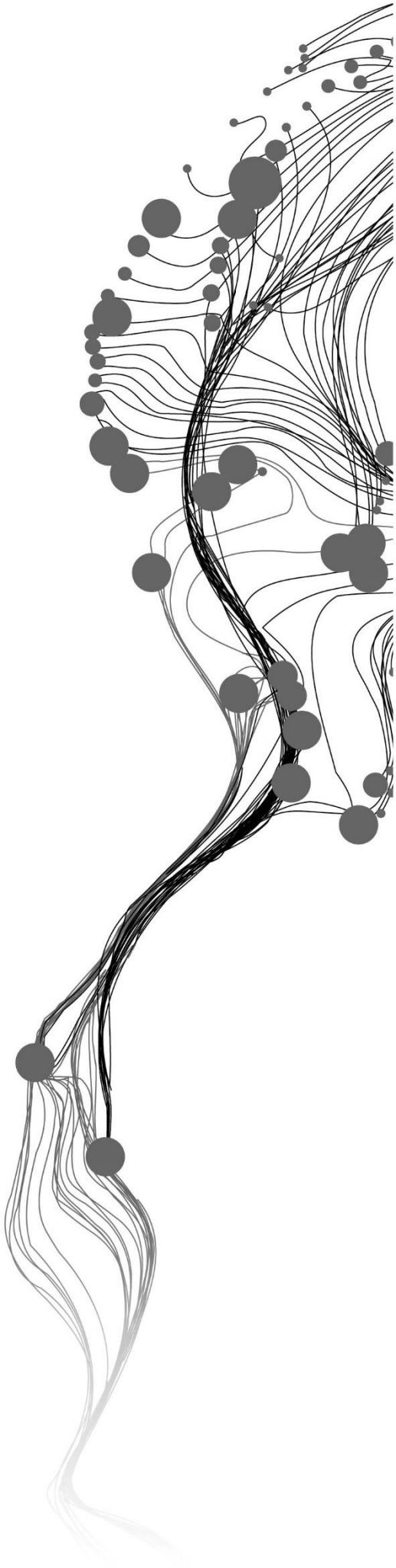
Leveraging LiDAR Data and Local Digital Twins Framework for Data-Driven Traffic Simulation

MAULANA IKRAM WIBISANA
July 2024

SUPERVISORS:
Dr. M. N. Koeva
Dr. P. Nourian MArch



Leveraging LiDAR Data and Local Digital Twins Framework for Data-Driven Traffic Simulation

MAULANA IKRAM WIBISANA
Enschede, The Netherlands, July 2024

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Urban Planning and Management

SUPERVISORS:
Dr. M. N. Koeva
Dr. P. Nourian MArch

EXTERNAL SCIENTIFIC ADVISORS:
Prof. D. Petrova-Antonova
Kaloyan Karamitov

THESIS ASSESSMENT BOARD:
Prof. Dr. K. Pfeffer (Chair)
Dr. Ir. O. A. L. Eikenbroek (External Examiner, University of Twente - ET)

ABSTRACT

Urban planning, specifically in the field of transportation planning, is important for promoting economic and social activities within urban area. Globally, transportation planning faces significant challenges, including increment in traffic congestion, which affects over half of the urban areas. Despite temporary traffic density reductions during the COVID-19 pandemic, congestion remains a persistent issue, with average drivers losing considerable hours to traffic annually. Additionally, traffic accidents cause numerous fatalities and injuries worldwide each year, underscoring the need for improved traffic management strategies. Similarly, Bulgaria, particularly Sofia, face similar challenges, including high road fatality rates and significant congestion, driving the implementation of the Sustainable Urban Mobility Plan (SUMP) 2019-2035. This plan aims to digitalize city transport to enhance urban mobility. Accordingly, this research aims to develop an urban traffic simulation Digital Twin (DT) framework by utilizing detailed traffic data, primarily in the Open Serialization Format (OSEF), captured by LiDAR sensors. These sensors provide comprehensive information about real-world traffic conditions. By integrating this detailed data, the DT framework seeks to improve traffic simulations, enable traffic observation in the digital world and potentially reduce congestion. The study area chosen for this research is a busy intersection in Sofia near the Paradise Center Mall, the largest mall in Sofia, where the LiDAR sensors are located. This area is suitable for extensive traffic observation and analysis. The traffic simulation tool chosen for this research is Simulation of Urban MObility (SUMO) due to its versatile uses and analytical capabilities, as well as its open-source nature, which is well-suited for the DT framework.

To develop the DT framework, the methods involve proper identifications, including issues inherent in the .osef dataset. It is found that there are challenges in the coordinate nature of the dataset and inherent classification that include a lot of unknown classes. Therefore, the method is first to transform the local coordinates of the .osef dataset concurrently with road segmentation of the intersection, followed by object type and trajectory type analysis, which is used as the base information for the reclassification process using Random Forest (RF) mode. This handles the issues of classification, including the existence of multiple classes in a single tracked object. Utilizing the PostGIS database as middleware to fetch enriched datasets for the traffic simulations, the framework demonstrates a successful attempt at running initial traffic simulations based on real-world traffic conditions.

The findings in this research show that DT frameworks have shown that it is capable of integrating the .osef dataset into the traffic simulations while addressing issues such as the coordinate transformations, including the reclassification of unknown class. The RF model employed in this research able to predict the true label of the unknown class into a specific object type (e.g., car, truck, two-wheeler or person). The minimization of multiple class prediction is reduced by 76% with the 2nd tuned model. The DT framework itself is validated against real-world traffic survey data, revealing a strong alignment with an R-squared value of 0.97, indicating that 97% of the variability in the observed traffic counts is explained by the simulated counts. Additionally, the cosine similarity analysis for vehicle trajectories demonstrated high directional accuracy, with most cosine similarity values nearing perfection (one in scale), and the Euclidean distance confirming minimal positional deviation. The implications of this research are more prominent in the what-if scenario testing, where it shows that roundabout road network design seems to be the most suitable to reduce traffic congestion. This research recommends that for future work, it is better to focus on testing the framework scalability across different intersections and residential areas, improving the Random Forest model with more advanced machine learning or deep learning techniques, and extending the observation period to capture more comprehensive traffic patterns.

Keywords: Transportation planning, digital twin, traffic simulation, LiDAR data, data enrichment, random forest, traffic management.

ACKNOWLEDGEMENTS

----------**Disclaimer**, this acknowledgement chapter use casual tone and wording----------

The journey of completing this research, especially at ITC, is not easy. There are a lot of struggles along the way, but I am able to overcome them with the support of so many people in my life. I am grateful that I am always able to learn and given the opportunity to do so during my whole journey. If I can exaggerate, blood, tears, and sweat are the foundation of this thesis research—of course, it is just an exaggeration. But truthfully, the journey has been fun for me, starting from applying to ITC itself to completing this thesis.

First of all, I would like to thank my family, my parents (Pak Djoko and Bu Yani), and my siblings (Mas Nanda, Ayu, and Lala) for all their support in encouraging me to continue my studies in the Netherlands, specifically in a top-tier faculty of geo-information, ITC. It truly is a once-in-a-lifetime opportunity.

I also want to express my utmost gratitude and thanks to my supervisors. Without their guidance, I would not have been able to complete this research. I would like to thank Mila for her enormous support, her valuable suggestions, and for being such an understanding supervisor. I would also like to thank Pirouz, whose guidance and insightful suggestions have been crucial in properly formulating my research. Additionally, I am very grateful for the cooperation from The Big Data for Smart Society Institute (GATE), specifically Dessi and Kaloyan. Thank you for the opportunity to collaborate with you and for the continuous support during my thesis process. Your role as my external scientific advisor has been invaluable. Without the help of these influential people, I would not have been able to finish this research.

During my time in Enschede (only 2 years), I have met many people who have made a significant impact on my life. I would like to thank all of them. First of all, the UPM kids in my batch—my study life at ITC would have been so bland without your company. The potlucks, picnics, and all the eventful activities we did together helped me get through my studies. I hope the universe lets us see each other even after graduation.

I also want to give many thanks to my ITC Indonesian friends, namely Salsa, Sry, Andi, Ghaly, Clava, Rifqi, Nasir, and Mas Ganda. All the unnecessary laughter, unimportant chat, unhinged jokes, and eating out helped me remember that, after all, I am still Indonesian. Who knows, if I had not hung out with you guys, I might have lost my identity (hypothetically).

Last but not least, to all of my 2022 batch mates at ITC—without you, my journey would surely have been different. The potlucks, parties, general trips around Europe, and all the other eventful times—too numerous to mention—are engraved in my brain and heart. I wish the best outcomes for all of us!

----------Thank you----------

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1. INTRODUCTION

1.1. Background

Urban planning is essential for efficient and safe city transportation (Yedavalli et al., 2021). It fosters economic and social activities, making cities attractive for various purposes (Rodrigues et al., 2021). However, on a global scale, transportation planning faces numerous challenges; for instance, according to Pishue (2023), traffic congestion is a major issue with 58% of urban areas experiencing increased delays. Despite a temporary reduction due to the COVID-19 pandemic, global congestion persists (WHO, 2019). Fernandez's 2023 report reveals that the average driver lost 51 hours to congestion in 2022, a 15-hour increase from 2021 (Fernandez, 2023).

Traffic accidents are another critical concern in transportation planning (Boonyotsawad et al., 2022). In the US, there were about 35,766 fatal accidents in 2020, while globally, road crashes cause 1.3 million deaths and 20-50 million non-fatal injuries annually (Boonyotsawad et al., 2022; Zhang, 2020). The National Highway Traffic Safety Administration (NHTSA) projected 42,795 traffic fatalities for 2022, a minor decrease of 0.3% compared to 2021(FHWA, 2023; NHTSA, 2023).

Traffic congestion in Europe incurs an annual cost of approximately one percent of its Gross Domestic Product (GDP), with urban mobility being responsible for 40% of CO2 emissions and 70% of other pollutants from road transport (Rodrigues et al., 2021). European cities are investing in cycling and walking infrastructure to mitigate these impacts. The reduced interest in shared mobility, the surge in private autonomous vehicle usage, and the effects of the pandemic have all contributed to this shift (Elks, 2021). Authorities have responded by implementing policies that promote active transportation modes and flexible work arrangements, employing data-driven mobility pattern monitoring and prioritising green and digital transitions in transport planning to achieve sustainable long-term growth (European Commission, 2022).

The operation of urban transportation is impacted by demographic issues, environmental concerns, and issues relating to traffic safety, and these challenges are not exclusive to Europe. It is important to highlight that Bulgaria's urban transportation is not an exception to these issues. In 2021, Bulgaria had the second-highest road fatality rate in the European Union, with 81 fatalities per million inhabitants, second only to Romania (The Sofia Globe, 2022a). However, the situation worsened in 2022, with a total of 175 people dying in road accidents in Bulgaria between January and May, an increase of 24 deaths compared to the same period in 2021 (The Sofia Globe, 2022b). To address the transportation challenges, Bulgaria's national transport policy focuses on sustainable infrastructure development, improving transport safety, and integrating various transportation modes (Ministry of Transport and Communications of Republic of Bulgaria, 2017).

Despite Bulgaria’s improvements and digitization efforts in public transportation, big cities such as the capital Sofia continue to struggle with overcrowding and congestion problems especially during peak tourist seasons, partially due to a high per-capita vehicle registration rate and a substantial number of unregistered citizens (Intelligent Transport, 2015; International Trade Administration, 2022; Modijefsky, 2023; World Bank, 2020). As the economic centre of the country, Sofia generates over 43% of the GDP, with an estimated population between 1.6 and 1.8 million due to commuting from adjacent cities which contribute to the increase of traffic density (World Bank, 2020). Based on data from the TomTom Traffic Index, during rush hour, drivers in Sofia spend an extra three days and 18 hours of driving compared to the 2021 record; this indicates that a high-level congestion occurred in Sofia (TomTom International BV,

2023). Aside from those, several other transportation issues also exist in Sofia. These include urban traffic jams and noise pollution from high motorization rates, older vehicles that emit more pollutants, a lack of sustainable transportation options, traffic jams brought on by subpar infrastructure, a lack of parking, and the need for better infrastructure for pedestrians and cyclists (Panayotova, 2022, pp. 14, 19, 27).

Sofia's City Council responded to the problem by enacting the Sustainable Urban Mobility Plan (SUMP) in 2019. The SUMP, a strategic document outlining principles for improving mobility in Sofia, spans 2019 through 2035. Its main goal is to create an integrated transportation system by advancing green transportation options, digitalizing city transportation, and encouraging sustainable transportation modes (Modijefsky, 2023). By 2035, the plan wants to lower the negative effects of transportation development, improve the urban environment's attractiveness, and raise living standards.

The SUMP 2019–2035 challenges traditional traffic regulation by emphasizing data-driven shared mobility and environmentally friendly urban transportation, focusing on public transportation, walking, and cycling (Sofia City Council et al., 2019, pp. 96–99). It highlights the need to increase public transportation, lessen reliance on private vehicles, and improve infrastructure. Designated transit lanes, pricing that discourages driving, and enhanced pedestrian crossings are some of the strategies that fall under the categories of "Pull" measures, which encourage bicycle and pedestrian traffic, and "Push" measures, which limit parking and road traffic (Sofia City Council et al., 2019, pp. 122-125). Implementing these measures is crucial.

1.2. Related work and research focus

Recent developments in mobile light detection and ranging, or LiDAR technology, greatly influenced transportation applications by making it possible to perceive the environment in detail and accuracy. This technology makes real-time traffic data collection easier, especially when used on roadside platforms. This supports both thorough traffic management research and improvements to road safety. Intelligent transportation systems can benefit greatly from LiDAR's ability to capture precise, high-resolution geometric (point-cloud) data. This is because it can be used in single-sensor and multi-sensor fusion approaches to create holographic scenes of traffic conditions. Such extensive data gathering highlights the importance of LiDAR in the current state of transportation research and is essential for improving traffic management methods and reducing congestion (Williams et al., 2013).

Accurate coordinate transformations are crucial for aligning local Cartesian coordinates with global geographical coordinate reference systems (CRS). Inaccuracies in this process can significantly impact spatial analysis outcomes, which is particularly relevant to LiDAR technology because the sensors often provide data only in local coordinates (Fan et al., 2014). Effective transformations improve the alignment of spatial datasets, which is critical for applications like traffic simulation. Building on the importance of spatial data alignment, high-resolution satellite imagery plays a crucial role in precise road extraction. Research by Xu et al. (2018) has demonstrated the use of such imagery for road extraction. However, the automatic extraction of detailed road segmentation has yet to show promising results. Detailed road segmentation, particularly at the level of lane distribution, enhances the accuracy of traffic simulations and improves traffic management systems. To address this, manual digitization is often used to complement the automatic methods.

Understanding traffic dynamics requires analysing Global Positioning System (GPS) data provided in sequences, which indicate the paths vehicles have travelled. This necessitates classification processes to better interpret these dynamics. Y. Zheng (2015) presents methods for trajectory data mining, highlighting the importance of spatial analysis in identifying travel behaviours and patterns. Complementing this, another research has explored the possibility of estimating and predicting trajectories using generative Artificial Intelligence (AI) with short-term memory algorithms (Gu et al., 2024). Existing

literature mostly focuses on predicting and estimating trajectories due to limited information. However, when granular input data provides detailed information about object positions per millisecond, it becomes feasible to classify trajectories directly, offering a more accurate reflection of real-world conditions as compared to estimations.

Handling real-time LiDAR point cloud data with embedded GPS information, in particular, often involves dealing with visual obstructions caused by factors such as bad weather or unexpected damages to the sensor component. This often results in a missing label, a common issue that artificial intelligence, particularly machine learning, is well-suited to handle. Random Forest (RF) is a powerful machine learning algorithm commonly used to predict unlabelled data into appropriate labels or categories, effectively handling complex datasets with mixed data types. This method is particularly effective in handling complex datasets with mixed data types. The robustness and accuracy of the RF algorithm in classification tasks have been well-documented (Breiman, 2001). In traffic simulation, RF can significantly improve the accuracy of moving object classifications, such as vehicles and pedestrians. This detailed classification is crucial for developing more realistic simulations. A study by Liaw & Wiener (2002) further elaborates on the implementation and advantages of RF in classification problems, highlighting its ability to handle large datasets with higher accuracy compared to other algorithms such as decision trees and support vector machines. The study also notes RF robustness to overfitting due to the ensemble approach, and its capability to provide estimates of feature importance, which are essential for understanding the model's decisions.

Parallel to the development of LiDAR applications, the concept of Digital Twins (DTs) has gained popularity in the Intelligent Transportation System (ITS) field. A digital twin is a dynamic digital model of a physical object or system that combines sensor data and analytics to reflect its real-world status, enabling real-time monitoring, simulation, and decision-making (Digital Twin Geohub, 2023). The research by Kušić et al. (2023) on the digital twin model of the Geneva Motorway (DT-GM) exemplifies the use of the microscopic traffic simulator SUMO (Simulation of Urban MObility) to model and simulate synchronized virtual representations of transportation dynamics.

SUMO, an open-source traffic simulation tool, is crucial for integrating digital twins and LiDAR data in transportation research. Its flexibility and adaptability present significant advantages over commercial traffic simulation programs. SUMO offers detailed, scalable simulations suitable for both urban and regional assessments, achieving an ideal balance between detail and computational efficiency (Maciejewski, 2010). This shows contrast to the commercial VISSIM, which delivers high detail at a higher cost, and TRANSIMS, which favours computational simplicity over detail.

Building on the success of prior work using SUMO, this research aim is to enhance operational efficiency and support dynamic traffic management by incorporating real-time LiDAR data streams into traffic simulation models. The research utilizes LiDAR data, focusing particularly on the Open Serialization Format (OSEF) dataset, which has not been extensively studied. The .osef dataset, developed by Outsight —a company specialized in 3D LiDAR data where the .osef dataset originated from— uses an advanced binary format for serialization, known as Type-Length-Value (TLV) encoding, specifically designed for data from LiDAR sensors. This format streamlines the processing of the large volumes of data produced by LiDAR sensors by focusing on delivering relevant information to specific applications (Vincent, 2023). Incorporating a parsing technique into this workflow enables a seamless transition from raw data streams to database management and, ultimately, to traffic simulation for the purpose of traffic conditions monitoring and assessment. While visualizing real-time data provides an immediate snapshot, traffic simulation allows for the analysis of potential scenarios and the evaluation of traffic management strategies, leading to a more comprehensive understanding.

In continuation, this research explores and tests the novel workflow to address prevailing challenges, enrich datasets, and seamlessly integrate LiDAR datasets into a digital twin-based traffic simulation. The objective is to demonstrate the possibility of this integrated approach in mitigating known issues while enhancing the overall functionality of traffic simulations. Through this integration, testing diverse traffic scenarios is made possible, transcending the limitations of real-time data analysis. This traffic simulation DT framework is able to support informed decision-making in traffic management and infrastructure planning. By exploring various scenarios and their corresponding outcomes, the research seeks to provide a comprehensive understanding of traffic dynamics, thereby facilitating the formulation of more effective strategies to optimize transportation systems.

1.3. Research objectives

To achieve the research purposes outlined in the previous section, this study sets specific objectives and research questions. By detailing the research objectives and sub-objectives, a clear workflow is established. The research and investigation ensure a comprehensive approach to understanding and improving traffic dynamics and management. Additionally, the research questions drawn from each sub-objectives will guide the inquiry towards these objectives, providing a structured thinking to assess the proposed workflow.

1.3.1. Main objectives

The main objective of this research is **to develop a digital twins-based urban traffic simulation utilizing LiDAR Data**.

1.3.2. Sub-objectives and research questions

Derived from the main objective, there are four sub-objectives and each sub-objectives consists of several research questions. They are stated as follows.

1. **To conduct a review of the state of the art including tools, models, and methods for traffic simulations.**
   - 1.1. What are the key best practices and case studies from existing literature that demonstrate the successful application of traffic simulation in intersection?
   - 1.2. What are the traffic simulation models and tools that have been developed and used for traffic simulation?
   - 1.3. What methods have been used for LiDAR data integration to traffic simulation?
2. **To process and enrich the LiDAR dataset, ensuring it is suitable for integration into traffic simulation.**
   - 2.1. What are the structures of the LiDAR dataset?
   - 2.2. What methods are appropriate to process and enrich the LiDAR dataset?
   - 2.3. What methods are suitable to integrate the processed data into the digital twin framework?
3. **To develop the traffic simulation digital twin framework**.
   - 3.1. What are the appropriate methods to generate traffic simulation at the intersection?
   - 3.2. How do different what-if scenarios affect the traffic simulation?
   - 3.3. What type of digital twin (2D or 3D digital twins) is suitable for the traffic simulation model?
4. **To assess the traffic simulation digital twin framework**.
   - 4.1. What is the accuracy of the traffic generated by the simulation compared to real-world traffic?
   - 4.2. What is the similarity of the traffic trajectories generated by the simulation algorithm compared to the recorded data trajectories?

1.4. Study area and datasets

1.4.1. Study area

The research is located in Sofia, Bulgaria, specifically focusing on one of the city's busiest intersections near Paradise Center, the largest mall in Sofia. This particular crossroads is located in a densely populated area of the city, and the construction of the mall has led to significant congestion in the local region, as illustrated in Figure 1. Furthermore, it's worth noting that this intersection is situated in the Lozenets (Лозенец) district, which, as of June 15, 2023, had a population of 63,214 people living at their current address and 67,093 people living at their permanent address (Ministry of Regional Development and Public Work of Bulgaria, 2023). This suggests that the area is quite densely populated. Numerous obstacles are brought about by the district's rapidly growing population. The area is covered by a network of busy, high-speed roads as well as smaller, neighbourhood-scale streets, cul-de-sacs, and public transportation options like the internal combustion engines, hybrid-electric and electric buses, and trams (Hristov et al., 2022).

Figure 1. Paradise Center mall intersections. (A) Source: Google Street View (2023).. Retrieved November 18, 2023, from https://www.google.com/maps/, (B) source: Rangelov (2023). Retrieved November 17, 2023, from https://www.youtube.com/watch?v=jFmi3rBEl, (C) source: Rashkov (2023). Retrieved November 18, 2023, from https://www.youtube.com/watch?v=_uAGMuzPeLI.

The intersection is composed of a single lane (Blvd. "Cherni vrah") that runs from south to north and a dual-lane road that splits into two streets: "Srebarna" that runs northeast and "Henrik Ibsen" that leads southwest. The numerous types of vehicles, such as cars, two-wheelers, high-load trucks, and buses, impact the traffic dynamics in this area. The traffic is monitored by a LiDAR system, comprising 6 sensors, which are the primary data source for this study. The specification of the LiDAR sensors is presented in Table 1.

Table 1. LiDAR sensors specification. Source: Author (2024).

Specifications	Details
Digital LiDAR	Ouster LiDAR OS1-128 uniform
FoV (Field of View)	45°, 64 layers
Processing module (Node)	ALB (Augmented LiDAR Box) processing computing interface
Classification	Cars, Trucks, Pedestrian, Two wheelers
Dimensions	• Diameter: 85 mm (3.34 inch) • Height: 58.45mm (2.4 inch) – 73.5 mm (2.9 inch)
Power Consumption	14 – 20W
Operating voltage	22 – 26V, 24V nominal
Operating Temperature	Between +53 °C and +60 °C, automatic range reduction (max 20% reduction)
Ingress Protection	• IP68 (1m submersion for 1 hour) • IP69K
Shock	IEC 60068-2-27 (Amp: 100g, Shape: 11ms half-sine)
Vibration	IEV 60068-2-64 (Amp: 3G-rms, Shape: 10-1000 Hz)
Range	• Resolution: 0.3 cm • Minimum: 0 - 0.3 m blockage flag, 0.3 m point cloud data • Accuracy: ±3 cm Lambertian and ±10 cm retroreflectors

The LiDAR sensor, Ouster OS1-128, features a 45° field of view across 64 layers, utilizing the Augmented LiDAR Box (ALB) processing module for classifying cars, trucks, pedestrians, and two-wheelers. It is compact in terms of size, measuring 85 mm (3.34 inches) in diameter and 58.45 mm (2.4 inches) to 73.5 mm (2.9 inches) in height. The amount of power needed for the sensor, which uses 14–20 watts to operate, is between 22 and 26 volts. It performs well in the extremes, from -53°C to 60°C, and at high temperatures, it automatically reduces range.

The Ouster OS1-128 is an exceptionally durable device that can withstand high-pressure water and submersion in part due to its IP68 and IP69K ingress protection ratings. It complies with the standards of IEC 60068-2-64 for vibration resistance and IEC 60068-2-27 for shock resistance. With a resolution of 0.3 cm, the sensor can identify objects up to 0.3 meters away, and its accuracy is ±3 cm for Lambertian surfaces and ±10 cm for retroreflectors.

The sensors, positioned along various borders surrounding the main intersection, record the movement of vehicles and pedestrians over time. The study area and sensor distribution are shown in Figure 2. This highly populated and traffic-congested area is a perfect case study to show the added value of data analysis and simulation supporting decision-making in the transportation planning domain. The opportunities and challenges presented by this study area reflect larger urban issues globally, suggesting that the research findings could benefit areas facing similar traffic and mobility challenges.

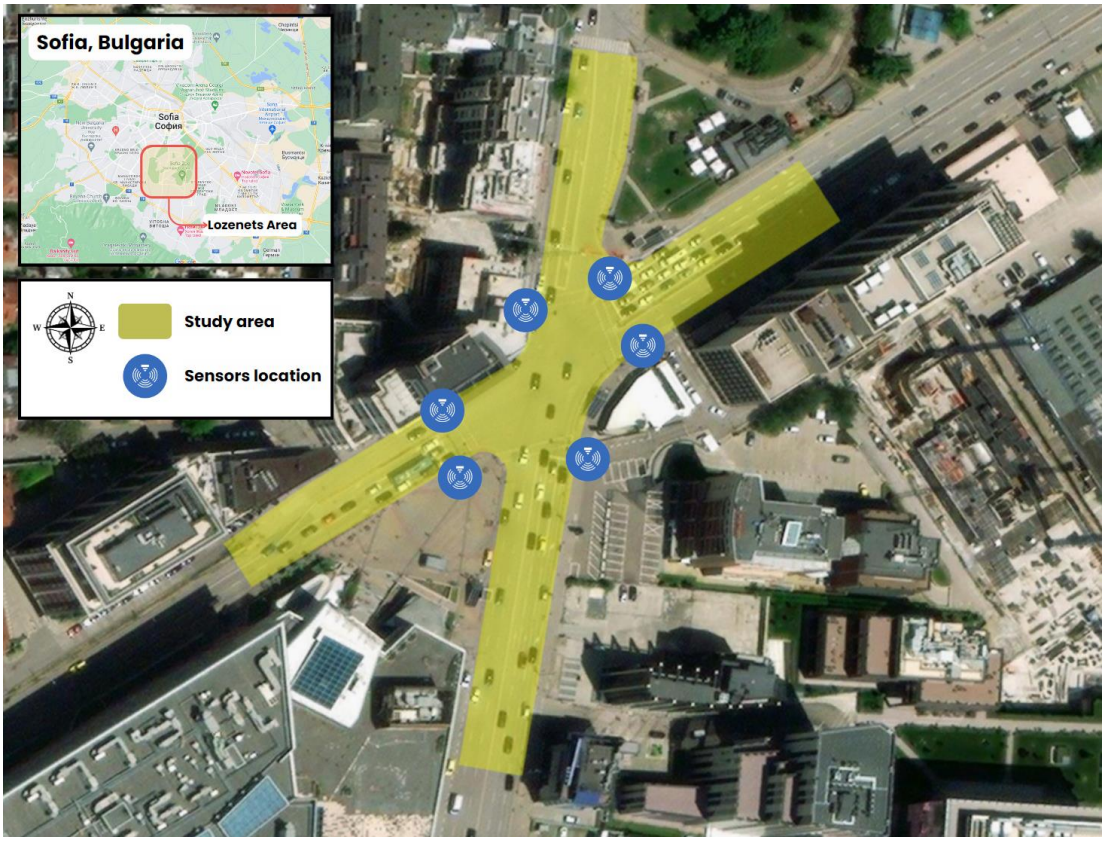


Figure 2. Study area and sensors distribution. Source: Author (2024).

1.4.2. Research datasets

The LiDAR data collected from the sensors is processed in real-time and made available for further analysis in .osef format, which is the main input dataset for this research. The .osef data format was developed to make LiDAR point cloud technology more manageable and accessible, addressing integration problems into ITS applications. The .osef datasets offer precise and comprehensive spatial measurements, making them invaluable across diverse applications. Its key advantages include adaptability, which simplifies processing; efficiency, by reducing processing overhead; simplicity, through straightforward parsing; robustness, offering versatile data management; compatibility, facilitating integration with new features; and scalability, accommodating varying data volumes.

The real-time download of the .osef dataset was managed using a TCP (Transmission Control Protocol) stream. The time-stamped data, specifically in GMT+3 to match the local time zone of each recorded frame, is organized systematically by the preprocessor of the sensors. The datasets used in the research vary, ranging from a two minute to 10-minute period; they are utilized as foundational data for specific research steps, which further explained in the methodology section. As shown in Figure 3, the data can be parsed to distinguish between objects that are considered dynamic and those that are static, belonging to the urban infrastructure. The left figure shows the collection of static objects LiDAR points captured over time, visualized using the pythreejs library. It depicts the shape of the study area intersection, from the road to the surrounding buildings. The right figure shows the filtered data of only tracked objects, displaying the shapes of vehicles and persons. This visualization was created using plotly, which also shows details of each object, such as the ID and local pose coordinates.

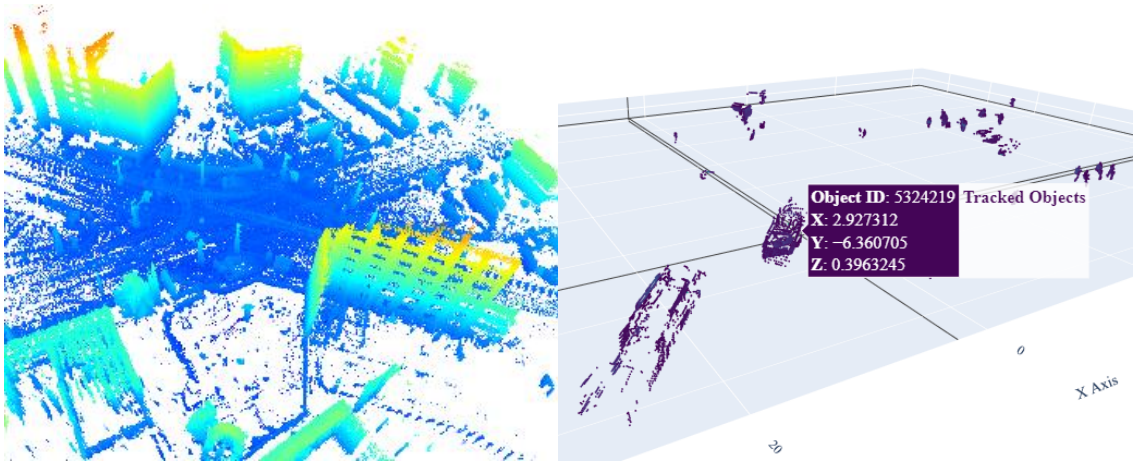


Figure 3. Intersection point cloud visualisation; static objects (Left), and dynamic objects (Right). Source: Author (2024).

The nested TLV (tag-length-value) tree structure is traversed while processing binary data into an array of data points or other formats, such as .csv. Figure 4 illustrates the TLV tree's detailed structure. As can be seen from Figure 4, the .osef data contains information on tracked objects and the augmented cloud, in addition to the previously described information pertaining to the base data. Unique object identifiers (IDs), classes consisting of CAR, PERSON, TRUCK, TWO_WHEELER, and UNKNOWN, speed in km/h (convertible to other units of measurement), volume computed using bounding boxes, coordinates (local pose x, y, and z, or Cartesian coordinates) and zones are among the information that can be extracted from tracked objects.

```
-> TLV #0

timestamped_data                                (type: 20)
|
├── timestamp_microsecond                       (type: 12)
|
├── scan_frame                                  (type: 25)
|   |
|   ├── augmented_cloud                         (type: 1)
|   |   |
|   |   ├── number_of_points                    (type: 2)
|   |   |
|   |   ├── number_of_layers                    (type: 14)
|   |   |
|   |   ├── cartesian_coordinates               (type: 6)
|   |   |
|   |   ├── reflectivities                      (type: 4)
|   |   |
|   |   ├── object_id_32_bits                   (type: 47)
|   |
|   ├── tracked_objects                         (type: 26)
|   |   |
|   |   ├── number_of_objects                   (type: 10)
|   |   |
|   |   ├── object_id_32_bits                   (type: 47)
|   |   |
|   |   ├── class_id_array                      (type: 18)
|   |   |
|   |   ├── speed_vectors                       (type: 28)
|   |   |
|   |   ├── bbox_sizes                          (type: 27)
|   |   |
|   |   ├── pose_array                          (type: 29)
|   |   |
|   |   ├── object_properties                   (type: 39)
|   |
|   ├── zones_def                               (type: 33)
|   |
|   ├── zones_objects_binding_32_bits           (type: 48)
```

Figure 4. TLV tree structure. Source: Author (2024).

The temporal sequence of the data points that is collected for every object is illustrated in Figure 5. The LiDAR GPS data shows that each recorded objects are composed of several frames organized according to the timestamp at the millisecond level. Multiple frames are captured of each object ID while it is within the bounding box of the sensors. The object position at a particular timestamp is represented by each frame, which is indicated as F_i where i is the frame index. These frames offer extensive information about the object, such as its class, local pose, volume, and speed at the given frame. With each frame recording a single instance in the movement and attributes of an object as captured by the sensors, it allows for accurate tracking of objects across time.

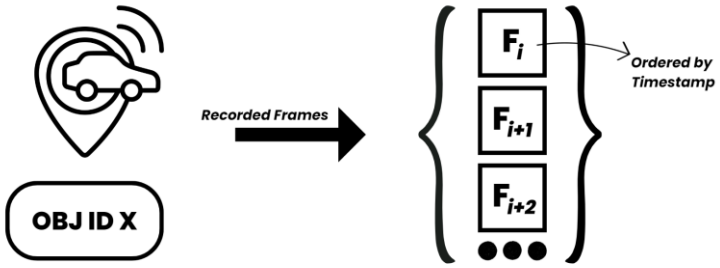


Figure 5. LiDAR .osef data structure. Source: Author (2024).

However, it is important to note a few issues with the dataset. Firstly, concerning the captured coordinates, the dataset currently records coordinates only in a local pose or cartesian coordinate system. They have not yet undergone geo-location correction and must be converted into geo-coordinates. Second, the data contain an unknown class, describing the dynamic objects that are not recognised. Third, due to the nature of the .osef dataset, an object ID might have multiple different classes since each frames have its own information about the recorded object class. Therefore, addressing these issues as practically as possible should be the initial step in this research. For details on the datasets used in this research, see Table 2 bellow.

Table 2. Research datasets. Source: Author (2024).

Datasets	Details	Nature	Usage
OSEF	Two-minute period Morning 2023-05-30	Proprietary- GATE owned	To develop the research workflow. (sub-objective 2 and 4)
	Four-minute period Afternoon 2024-01-29	Proprietary- GATE owned	
	10-minute period Morning 2024-05-30	Proprietary- GATE owned	Used for traffic simulation and DT framework validation. (sub-objectives 3 and 4)
Open street map (OSM)	Road network	Open source	To develop the traffic simulation DT framework. (sub-objective 3)
	Building footprint	Open source	
Satellite imagery	USGS Landsat 7 ETM+ C2 L1 15-meter panchromatic blend	Open source	To develop the traffic simulation DT framework. (sub-objective 2)
	ESRI World Imagery service Up to 1-meter resolution	Proprietary - ArcGIS Pro license	
Real-world traffic count	Traffic count of the intersections on 2024-05-30 Morning 9.00 – 9.50 AM	Proprietary – Self survey	To evaluate the traffic simulation DT framework (sub-objective 4)

1.5. Summary

This introduction chapter sets the foundation for the research by addressing critical issues in urban transportation planning, particularly focusing on traffic congestion and accidents. It highlights the global and local challenges, emphasizing the significant economic and environmental impacts. Specifically, it discusses the congestion problems in Sofia, Bulgaria, and the strategic responses, such as the Sustainable Urban Mobility Plan (SUMP), aimed at improving urban mobility and living standards. The chapter also delves into advancements in mobile LiDAR technology and its transformative potential in traffic management. It emphasizes the importance of accurate data processing, including coordinate transformations and detailed road segmentation, for effective traffic simulations. The research leverages the Random Forest algorithm and DT frameworks, using tools like SUMO for dynamic traffic simulations to integrate real-time LiDAR data and enhance traffic management strategies. The chapter outlines the primary and sub-objectives of the research, focusing on developing a traffic simulation DT framework utilizing LiDAR data. It specifies the study area in a busy intersection in Sofia and describes the datasets, particularly the .osef format, highlighting their structure and the challenges in data processing. This introduction sets the stage for exploring innovative methodologies to improve traffic simulations, ultimately supporting better decision-making in urban transportation planning.

2. LITERATURE REVIEW

This literature review chapter explore existing research and literature on traffic simulations, with a particular focus on intersections. It reviews various traffic simulation tools, assessing their suitability for the current research based on their features and capabilities. Additionally, the chapter evaluate different traffic simulation models to determine which ones align best with the selected tools. A significant part of this review also cover the integration of LiDAR data into traffic simulations, especially focusing on GPS-based datasets from LiDAR scanning tools that record detailed traffic information. By examining these aspects, the literature review aims to provide a comprehensive understanding of the current state of traffic simulation research and identify the most appropriate methodologies and tools for this research study.

2.1. Traffic simulation in intersections

Traffic simulation has commonly been used and has become an important tool in transportation research, with its ability to address various issues such as congestion, safety, and environmental impact. The evolution of computation technologies itself has made traffic simulation indispensable for analysing and designing a traffic system. Notable advancements in this research field include data-driven frameworks that are able to integrate human-driving data, which enhance the realism and precision of simulations (Hao & Ruan, 2024). This example of developments allows traffic simulations to incorporate a wider range of variables that affect traffic and vehicle operations, providing a more comprehensive traffic dynamics analysis.

Previous research in traffic simulations spans to various applications, from urban traffic management to autonomous vehicle testing. For instance, studies have shown that data-driven microscopic traffic simulations are particularly effective for autonomous driving tests due to their ability to utilize highly detailed traffic data, enable large-scale testing, and ensure scenario reproducibility (Chen et al., 2023). Another study by Barcelo (2010) has also discussed the fundamentals of traffic simulations and emphasized the significance of using different categories of traffic simulation for different study cases; it specifically prominent that for capturing the complexities of urban traffic in a detailed manner, microscopic approach is often used and proven effective.

Traffic simulations can broadly be categorized into macroscopic, mesoscopic, and microscopic. Macroscopic simulation is able to simulate traffic flow at an aggregate level, focusing on the bigger picture and overall traffic density, flow, and speed using equations derived from fluid dynamics. This type of traffic simulation is suitable for large-scale traffic research and studies, such as highway traffic flow in a district or bigger study area, where individual vehicle interactions are less critical (Mohan & Ramadurai, 2013; Zeb et al., 2023). Disadvantages of this type of traffic simulation, however, they lack the granularity needed for more detailed analysis, such as in an intersection.

Another type of traffic simulation is mesoscopic simulation. This type of traffic simulation offers an intermediate level of detail, capturing both aggregate traffic flow and some aspects of individual vehicle behaviours. Mesoscopic offers a balance of computational efficiency with possibilities of detailed analysis, making them useful for urban traffic management and traffic decision dynamics (Chang et al., 1985). For example, Burghout et al. (2005) discussed a certain traffic simulator named Mezzo, which is designed for simulating large networks with individual vehicle behaviours aggregated at link levels, and it can also function as a hybrid model with microscopic simulations for specific areas of interest.

As mentioned previously, microscopic simulations are the type of traffic simulations that focus on a specific area of study. It is able to provide the highest level of detail, compared to the other traffic

simulation categories, by simulating individual vehicle movements and interactions. Each vehicle is modelled with specific attributes and behaviours settings. This allows for a more precise analysis of traffic dynamics in a small study area, including lane changes behaviour analysis and detailed scenarios exploration. Microscopic simulations are essential for evaluating the impact of traffic control measures, road designs, and safety assessments, which involves analysis of individual vehicles' traffic behaviour (Ben-Akiva et al., 2010; Mahmud et al., 2019).

Previous research has compared those three different types of traffic simulations based on their effectiveness. Considering that the specific area of interest of this research is an intersection, microscopic traffic simulation determined to be the most appropriate traffic simulation category to be used for studying intersection dynamics due to its ability to model detailed vehicle intersections and behaviours. Microscopic models can accurately represent vehicle behaviours at intersections, including stopping, turning, and yielding (Cameron & Duncan, 1996). Zhong et al. (2023) have demonstrated the advantages of microscopic simulations in urban environments, highlighting their ability to capture detailed vehicle interactions that macroscopic and mesoscopic models often overlook. Similarly, research by Barcelo (2010) found that microscopic models like VISSIM and SUMO provided more accurate and reliable results for urban micro analysis such as intersection, compared to their macroscopic and mesoscopic counterpart. Aside from those, microscopic simulations also provide detailed presentation of vehicle movements making it possible to evaluate impact of road infrastructure and traffic control measures in high accuracy. By incorporating detailed driver behaviour models, microscopic simulations are also capable of mimicking real-world traffic conditions (Brockfeld et al., 2005; D. Wang et al., 2023). This traffic simulation can handle high-quality and detailed traffic data, making it perfect for managing the dataset of this thesis research, especially in an intersection as the area of interest.

2.1.1. Traffic simulation tools

There are several traffic simulation tools that have been developed over the years, each with unique features and capabilities. In microscopic traffic simulations especially, the tools have an important role in analysing and understanding traffic dynamics at a granular level. These tools model individual vehicle behaviours and interactions and expected to be able to offer detailed insights into traffic flow, congestion patterns, and impacts to various traffic management decisions and strategies. Some of the known and prominent tools in the microsimulation category include VISSIM (Verkehr In Städten – SIMulationsmodell), AIMSUN (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks), Paramics (Parallel Microscopic Simulation), MITSIMLab (MIT Simulation Laboratory), and SUMO (Simulation of Urban MObility).

VISSIM is one of the most widely used traffic simulation tools in microscopic traffic simulation, owned by PTV Group, this tool allows for the simulation of various traffic scenarios, including mixed traffic environments and public transportation systems. Research by Khare (2024) highlights VISSIM's ability to model detailed traffic interactions in heterogeneous traffic conditions, which showcases its effectiveness in modelling traffic conditions. Similarly, Ziemska-Osuch & Osuch (2022) have used VISSIM to study the traffic signal effects on the number of pedestrians in an intersection; this demonstrates the sensitivity of the tools towards specific parameters. Additionally, VISSIM has also been utilized to evaluate pedestrian safety measures, including the vehicle emissions, which highlights its robustness in analysing various traffic issues (Medapati et al., 2022; Wu et al., 2012). Despite these advantages of VISSIM that have been discussed, the proprietary nature and high cost of this tool can be barriers for some users. It has a steep learning curve, computationally intensive, which leads to longer simulation times (Espejel-Garcia et al., 2017; Fellendorf & Vortisch, 2011; Park & Schneeberger, 2003).

AIMSUN, on the other hand, has the ability to integrate both microscopic and mesoscopic simulations. It offers capabilities to simulate the behaviours of individual vehicles in a network, making it

suitable for a wider range of applications from traffic management to research and planning. It is also known for its efficient simulation algorithms and user-friendly interface. A study by J. Wang et al. (2017) demonstrated AIMSUN effectiveness in analysing collision possibilities of vehicles, emphasizing its strength in safety assessments and conflict analysis in traffic studies. This study also highlights the possibility of utilizing AIMSUN for detailed microsimulation by developing a plugin that adopts the tools algorithm and traffic models. Other relevant studies include Barcelo & Casas (2005), which demonstrated the effectiveness of this tool in evaluating traffic control measures and its ability to handle a bigger-scale simulation. However, the AIMSUN license can be costly, and its nature is proprietary, which can hinder research with this tool due to limited access for some future users. Furthermore, this tool requires extensive calibration for accurate results, indicating the complexity and resource-intensive nature of this tool (Barcelo & Casas, 2005; Bessa Jr et al., 2021; Paulsen et al., 2022).

Paramics is another microscopic traffic simulation tool that is powerful; it is often used in the traffic engineering field. A study by W. Li et al. (2017) emphasized this particular tool's ability to simulate a slightly wider range and scale of road networks compared to the other microscopic traffic simulation tools. Moreover, Mynuddin & Gao (2020) validated a distributed predictive cruise control approach through Paramics simulations, which highlights its role in innovative traffic control strategies. Paramics, however, just like other traffic simulations tools, it has limitations. It has been discussed that Paramics indicate complexity and resource-intensive nature to fine-tune the model for accurate results (Cameron & Duncan, 1996; Reza et al., 2016). Furthermore, Fang & Tettamanti (2021) pointed out that the impact of automated driving technologies on microscopic traffic simulation practices, suggesting potential gaps in Paramics adaptability to evolving transportation trends.

MITSIMLab is more sophisticated traffic simulation tools that has been shown to be effective in intersection microsimulation studies. Giuffrè et al. (2018) has demonstrated the utility of MITSIMLab in evaluating roundabout safety performance through surrogate safety measures. This emphasize its role in assessing intersection safety accurately. Additionally, existing research has indicated that microsimulation tools, including MITSIMLab can accurately predict traffic conflicts at signalized intersections, highlighting the safety enhancement factors (AlRajie, 2018). Similarly to VISSIM and Paramics, certain challenges can be seen in MITSIMLab. It can suffer from accuracy limitations due to the need of extensive calibration of the parameters, which may impact the precision of the results, not to mention its complex structure and methodology that may hinder the usability of the tool (Ben-Akiva et al., 2010; Ilyas et al., 2024; Rashid et al., 2020). In a study by Ma & Fukuda (2014), they highlight certain disadvantages of MITSIMLab regarding its shortcomings in geographical data handling and interoperability, which shows deficiencies in GIS support.

Compared to the other tools that have been discussed, SUMO is the most common and widely used and powerful traffic simulation platform. It is an open-source and multimodal platform that allows for the modelling of microscopic interactions among various types of vehicles, including autonomous vehicles, transit passengers, and other traffic participants (Clemente, 2022; Huang et al., 2021). Due to the open source in nature, it allows customization and extension of its functionalities. One of the advantages of using SUMO for microsimulation in intersections is its ability to accurately model and simulate the behaviour of different types of vehicles and their interactions. This includes modelling the movement and decision-making of individual vehicles, such as lane-changing behaviour and traffic signal control. Existing research has shown the capabilities of SUMO in simulating traffic especially microscopic traffic in intersections, which align with this research scope. Warchol et al. (2017) assess signal-phasing schemes at diverging diamond interchanges, illustrating the complexities in optimizing signal-phasing strategies at intersections, which SUMO is capable of handling. A study by Gavric et al. (2024) has shown a limitation of SUMO by conducting an environmental evaluation of automated pedestrian detection systems; they pointed out that the systems are not too precise in detecting pedestrians.

To comprehensively understand the suitability of different microscopic traffic simulation tools, it is essential to delve into a detailed comparison based on their capabilities, advantages, and limitations. The tools reviewed, including VISSIM, AIMSUN, Paramics, MITSIMLab, and SUMO, offer varied features that cater to different traffic simulation needs. Table 3 summarizes these aspects to highlight the most appropriate tool for intersection traffic analysis.

Table 3. Traffic simulation tools comparison table. Source: Author (2024).

Traffic simulation tools	Simulation capability	Detailed traffic behaviour	Real-time data integration	Geographical integration	Software nature
VISSIM	Meso and Micro	Yes	Limited	Yes	Licensed
AIMSUN	Meso and Micro	Yes	Limited	Limited	Licensed
SUMO	Macro, Meso and Micro	Yes	Yes	Yes	Open source
Paramics	Micro	Yes	No	Limited	Licensed
MITSIMLab	Micro	Yes	No	Limited	Licensed

As shown in the table, SUMO stands out as the most suitable tool for the research focus on intersection traffic analysis. Its open-source nature, capability to handle real-time data and detailed traffic behaviour modelling make it the ideal choice for this study.

2.1.2. Traffic simulation models

Traffic simulation utilizes various traffic models, specifically car-following model. Car-following models are used to describe how vehicles adjust their speed and position in relation to the vehicle in front of them. Various models, such as the Intelligent Driver Model (IDM), Krauss model, and Wiedemann model, are commonly used in microscopic traffic flow simulations to capture car-following behaviours and interactions (Qi & Ying, 2023). SUMO itself utilizes various traffic models, including car-following models, to simulate the behaviour of vehicles in traffic as realistic as possible.

The Krauss model, a space-continuous traffic model, is one of the traffic models that is integrated into traffic simulation tools like SUMO. Developed by Stefan Krauss, this model focuses on safety and ensures that vehicles maintain a safe distance. It is a deterministic model based on the safe time headway principle and is particularly useful in scenarios requiring strict safety compliance. Furthermore, it offers comprehensive framework for simulating traffic dynamics (Krauß et al., 1998; Wei et al., 2020).

Wiedemann's car-following model is another traffic simulation model that is a fundamental component of traffic simulation tools. Originally developed for VISSIM, the Wiedemann model has been adapted for SUMO. It includes more parameter to describe vehicle interactions, making it suitable for detailed traffic analysis, but more complex to calibrate (Fellendorf & Vortisch, 2011). L. Wang et al. (2024) highlighted the significance of the Wiedemann model in analysing mixed traffic flow characteristics, emphasizing its role alongside other traffic simulation models.

One commonly used car-following model in SUMO is the Intelligent Driver Model (IDM). The IDM is a social force model that accurately represents the car-following behaviour of drivers. Compared to other models, IDM is known for its simplicity and effectiveness in representing realistic driving behaviours, based on this balance between realism and computational efficiency alone, IDM is the ideal model to be used in this research (Treiber & Kesting, 2013). It considers factors such as desired speed, desired headway, and acceleration and deceleration capabilities of the (Lin et al., 2017; Zhao et al., 2022). Based on Zhao et al. (2022) the model is defined by equation as follow:

$$a_{i,k+1} = a_i \left[1 - \left(\frac{v_{i,k}}{v_f} \right)^{\delta} - \left(\frac{s_d\left(v_{i,k}, \Delta v_{i,k}\right)}{s_i} \right)^2 \right], \quad (1)$$

where v_f is the free flow velocity, $\bar{a}_i$ is the ideal acceleration of the i_{th} vehicle, and $a_{i,k+1}$ is the acceleration of the i_{th} vehicle at $k + 1$ time step; the i_{th} vehicle's speed at the k time step is given by $v_{i,k}$; the speed difference between the $i - 1_{th}$ and i_{th} vehicles at the k time step is denoted by $\Delta v_{i,k}$; the actual distance between the $i - 1_{th}$ and i_{th} vehicles is given by s_i; and s_d is the expected distance between the $i - 1_{th}$ and i_{th} vehicles (Zhao et al., 2022).

The speed of the i_{th} vehicle and the speed differential of the previous car determine the expected distance s_d, and the calculation equation is,

$$s_d\left(v_{i,k}, \Delta v_{i,k}\right) = s_0 + T v_{i,k} + \frac{v_{i,k} \Delta v_{i,k}}{2\sqrt{a_i b_i}}, \quad (2)$$

where T is the headway at a safe time, $\bar{b}_i$ is the vehicle comfortable deceleration when it is stationary, and s_0 is the minimum headway (Zhao et al., 2022). It is important to note that equation (2) is evaluated first, and its output serves as an input for equation (1). These functions constitute an existing implementation within the simulation models. The desired outputs of the function for the simulation model include the acceleration of the vehicle and the expected distance between vehicles, resulting in dynamic vehicle movements in the microsimulation model.

Aside from that, SUMO also employed a routing algorithm; Dijkstra and A* algorithm are the one that is used in SUMO simulations. The main purposes of these algorithms are to make the vehicles in the traffic simulations able to find their path in the simulation's environment. Dijkstra algorithm, one of the routing algorithms employed in SUMO, find the shortest path between nodes in a graph by systematically exploring all possible routes. It starts with setting up the initial node distance to zero and all other nodes to infinity, marking all nodes as unvisited. The algorithm then evaluates the unvisited neighbours of the current node, updating their tentative distance. The node with the smallest distance becomes the next current node, and this process repeats until all nodes are visited (Cormen et al., 2009; Dijkstra, 2022). The equation is as follows:

$$\mathrm{D[v]} = \min\left(\mathrm{D}[v], \mathrm{D}[u] + w(u, v)\right), \quad (3)$$

Where $\mathrm{D}[v]$ is the current shortest distance to node v, u is the current node, and $w(u, v)$ is the weight of the edge between u and v.

As for the A* algorithm, it enhances Dijkstra by incorporating heuristics to prioritize node exploration, therefore speeding up the route search process. Initially, the algorithm sets the starting node cost to zero and places it in an open set, with all other nodes marked as having infinite cost. The algorithm selects the node with the lowest f-score (sum of path cost g and heuristic h) from the set, evaluates its

neighbours, and updates their cost if a lower cost path is found (Hart et al., 1968). This function, often the Euclidean distance for grid-based paths, is defined as follows:

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

Where $g(n)$ is the cost from the start node to node n, and $h(n)$ is the estimated cost from n to the goal.

2.2. LiDAR data integration in traffic simulations

The integration of LiDAR data into traffic simulations represents a significant advancement in traffic modelling, offering detailed and accurate data for real-time traffic analysis. Although research specifically focusing on LiDAR data integration in traffic simulations is limited, several studies have utilized similar high-resolution traffic data with GPS information to enhance traffic simulation models, which shares similarities with the LiDAR data used in this research in terms of high-resolution and real-time capabilities. For example, Berbar et al. (2022) investigated reinforcement learning-based control of signalized intersections, demonstrating the integration of real-time data to optimize intersection operations. This study illustrated the potential of leveraging detailed traffic data with platoon, for better dynamic traffic control strategies. Furthermore, Kitajima et al. (2019) utilized multi-agent traffic simulations to assess the impact of automated technologies on safety, it used real-world data to evaluate crash reductions associated with advanced driver assistance systems. By incorporating detailed traffic data into simulation frameworks, researchers could effectively analyse the safety implications and contribute to informing future transportation policies and infrastructure development.

In addition, study by J. Zheng et al. (2012), where modelled vehicle speed fluctuations using a cellular automata model showcased the integration of real-time traffic data to accurately stimulate speed dynamics. This method opens the possibility of capturing nuanced speed variations and enhancing the accuracy of traffic simulations. Kušić et al. (2023) study has also shown promising possibilities of integrating real-time data into traffic simulations. Genova model, which is introduced in this study, emphasizes the uses of real-time traffic count data of a highway to further improve the traffic simulations in SUMO. The integration method proposed in this study involves the connection of the Genova traffic count API. The integration was made possible through a database approach, where this database is used as a middleware to send the information to traffic simulations. These integration frameworks offer valuable insights into optimizing traffic simulations and enhancing traffic management strategies through the integration of detailed traffic data. While direct studies on LiDAR data integration are scarce, the successful integration of detailed traffic data with GPS information in various research studies underscores the significance and possibilities of leveraging real-time data for traffic simulations. This research aim to develops a suitable integration framework compatible with the .osef data by taking into account the existing research that has successfully integrated real-time and GPS-based traffic data into the traffic simulations.

2.3. Summary

This literature review chapter explores the state of the art in traffic simulation, with a focus on intersections. Traffic simulation is essential in transportation research, addressing congestion, safety, and environmental impacts, with advancements in computation technologies enhancing realism and precision. The chapter categorizes simulations into macroscopic, mesoscopic, and microscopic, emphasizing the effectiveness of microscopic simulations for detailed intersection analysis due to their ability to model

individual vehicle movements and interactions. Tools like VISSIM, AIMSUN, Paramics, MITSIMLab, and SUMO are highlighted, with SUMO standing out for its open-source nature, real-time data handling, and detailed traffic behavior modeling. It also covers various traffic models such as the Intelligent Driver Model, Krauss model, and Wiedemann model, essential for simulating realistic driving behaviors. Additionally, the chapter discusses integrating LiDAR data into traffic simulations, noting that while direct studies are limited, research using high-resolution GPS data shows potential for enhancing simulations. Methods such as database middleware for real-time data integration are explored, illustrating the benefits of detailed traffic data in optimizing simulations.

3. RESEARCH METHODOLOGY

The methodology framework in this research is conducted to achieve the main objective of the research of developing a digital twin-based workflow that can integrate real-time LiDAR data into traffic simulations for effective traffic monitoring and assessment. The research methodology explores the potential of OSEF data integration into traffic simulation while addressing data issues such as unknown classes and spatial alignment problems for the traffic simulation input. The use of the digital twin-based workflow can potentially improve the decision-making process. This is explored by analysing various scenarios based on the dataset and the intersection geometry within the workflow's functionality. The conceptual diagram of the traffic simulation DT framework can be seen in Figure 6 bellow.

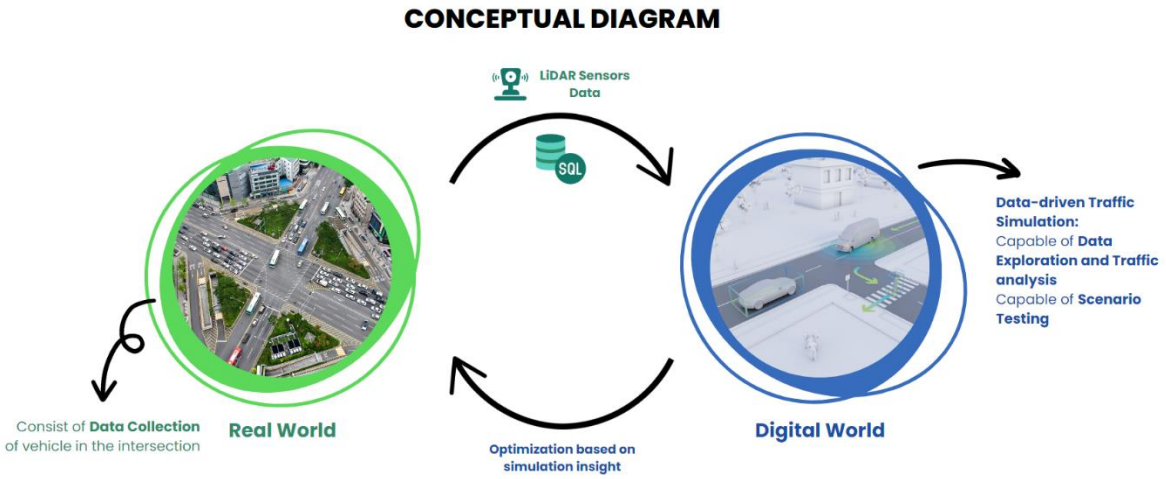


Figure 6. Conceptual diagram. Source: Author (2024).

Depicting from the conceptual diagram, the proposed approach starts with the acquisition of primary datasets in the single crossroad settings for the input of the microsimulation model. These datasets include OSEF real-time traffic data, and network data from OSM. These datasets are then transmitted to a common database view platform. The platform involves a continuous update process on dynamic datasets (traffic datasets), including data transformation and filtering, before establishing a connection with the microsimulation model. This database is linked to a traffic simulation model, enabling the model to consistently gather real-time data and subsequently update the simulation based on the latest dataset in the database. Conceptually, the digital twin-based traffic simulation is then able to provide a data-driven traffic simulation with capabilities for data exploration and traffic analysis. Furthermore, testing outcomes in various what-if scenarios is possible, thereby providing valuable insights for informed decision-making.

The methodology flow chart shown in Figure 7 (next page) outlines the detailed and structured process through which the thesis research is conducted. This process consists of three different predefined processes, each of which is detailed in their respective sections. These predefined processes provide a clear explanation of the steps and procedures involved in the research, showing how each component contributes to the overall study. The sub-sections cover various aspects of the research, such as data collection, data analysis, model development, and validation. Each predefined process is described thoroughly to ensure a comprehensive understanding of the research workflow. This detailed sub-process aims to clarify the systematic approach taken, explaining the reasoning behind each methodological choice and its impact on the research results.

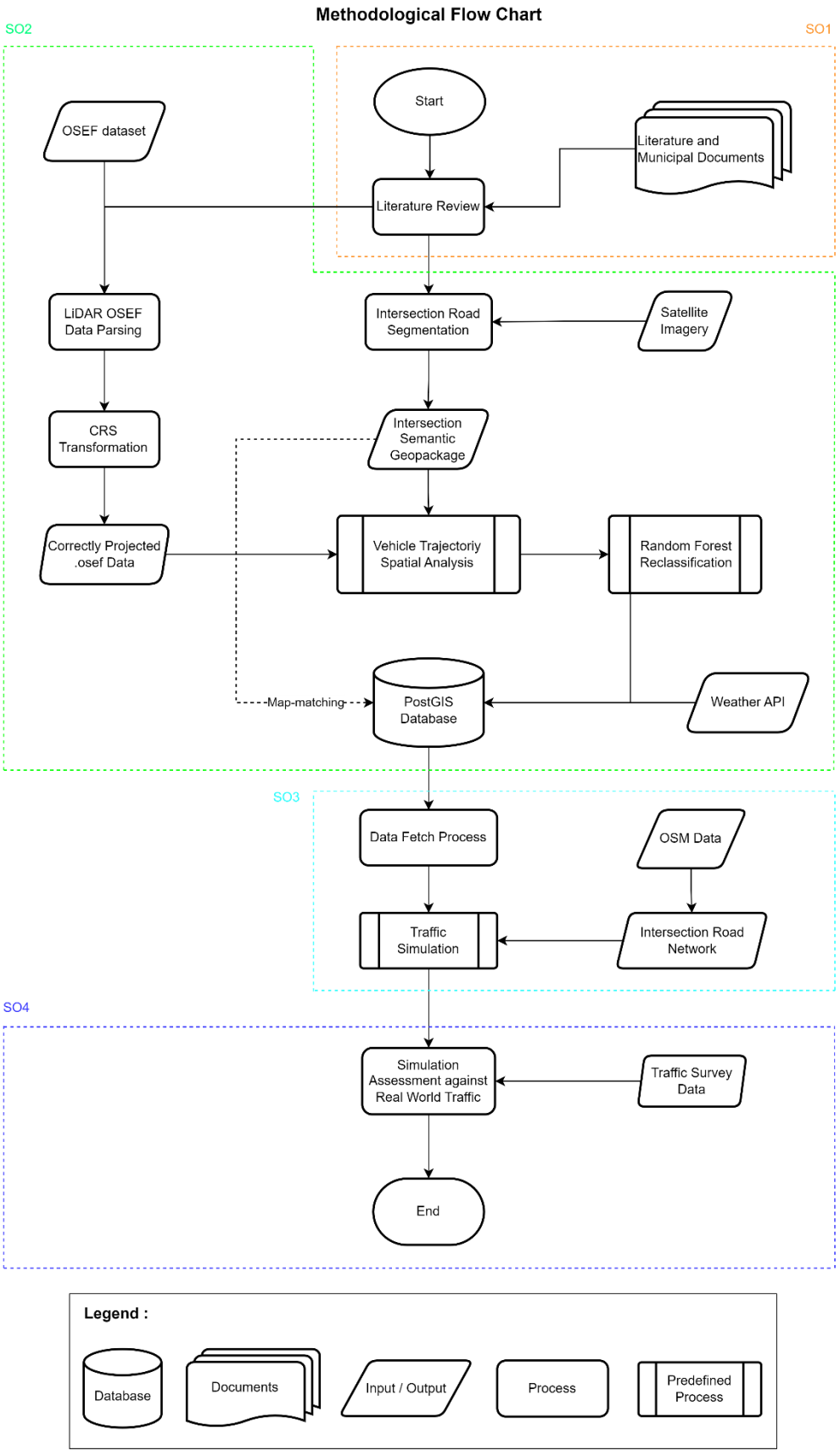


Figure 7. Methodological flowchart. Source: Author (2024).

The methodology involves several key steps: the preparation stage to review existing literature, collecting data from the .osef dataset and satellite imagery, parsing and transforming it into an appropriate CRS, and classifying object types within each frame. Trajectory spatial analysis is performed to classify each moving object's trajectory, providing features for random forest reclassification to address unknown classes and enhance data enrichment. Properly classified trajectories are essential for accurate traffic simulations, as they must align with the simulation network. Processed data is stored in a PostGIS database for efficient retrieval, and incomplete trajectories are corrected through map-matching. Data is then fetched into the simulation in two different ways: using XML-based traffic simulations and the dynamic TraCI (Traffic Control Interface) method for real-time monitoring and assessment. This approach improves the data processing and simulation process.

During the preparation stage, a thorough literature review is carried out to gather information from current studies, relevant literature, and other required documents for developing the digital twin models for traffic simulation. Determining the best traffic simulation tools for the case study of this research and figuring out which traffic models could be employed are important factors to take into account. This include includes the review of the potential approach to integrate the LiDAR dataset as the main research dataset for the traffic simulation DT framework. Furthermore, potential datasets is identified and shortlisted at this step, after which they undergo the necessary processing. This stage aligns with the first sub-objectives.

The next stage involves parsing and transforming the .osef dataset in an appropriate CRS using a Python script, along with OSEF 3.0 libraries. The parsing algorithm for extracting the information is structured to efficiently process large volumes of data and retrieve specific details about tracked objects. After the local coordinate transformation, iteration through the zone bindings linked to the object is performed, which involves frame-by-frame parsing. By iterating through these frames, the algorithm ensures that all spatial and temporal data are accurately captured, allowing for detailed analysis and classification of object movements within the dataset.

For each tracked object in a frame, the parsing algorithm retrieves the object ID, bounding box dimensions and timestamp in string format. The speed of each object is calculated from its speed vectors, and volume is estimated using each object's bounding box dimensions. All this information is extracted and stored in the array of data points, which includes frame number, date, timestamp, object ID, class name, class ID, speed, volume, and geo-coordinates. This array of data points are the new input for the following process of trajectory spatial analysis. This process includes the interoperability of the parsing algorithm towards the main TCP streams, not just the downloaded .osef data. This means the parsing algorithms are suitable for direct connections to the stream as well without much adjustment.

Satellite imagery is the main input for semantic road segmentation. High-resolution satellite imagery—USGS Landsat 7 ETM+ C2 L1 with a resolution of 15-meter panchromatic blend—and ESRI's world imagery service (up to 1-meter resolution) are used to produce detailed segmentation of the intersection road network based on the blend of the satellite images. ArcGIS is then used to process these images and build the road segmentation. By integrating these high-resolution images with ArcGIS, accurate and detailed segmentation of the intersection network is achieved efficiently. The segmentation handles the road segments as detailed as possible, capturing the junction and lane distribution. The output of this process is a geopackage (.gpkg), an open, OGC (Open Geospatial Consortium) standards-based format for geographic information system data containing the intersection of semantic information. This .gpkg is then used as input for the trajectory spatial analysis to classify the trajectory type based on the parsed GPS sequences and the position in the intersection segments.

Trajectory spatial analysis determines if an object has a complete trajectory and classifies objects as vehicles or non-vehicles based on their intersection location. This analysis serves two key purposes:

first, to classify the object and trajectory type as an additional feature for the random forest model training, enhancing its performance by learning patterns based on trajectory type, and second, to enrich the object information with their sequence of trajectory. This involves identifying which road segments the objects travelled through, rather than relying solely on geocoordinates. This is crucial for input in SUMO, as it requires the identification of the object's starting location, the middle sequences, and the end location in the road segments to properly load the objects into the simulation.

Reclassification —where the unknown as a class is predicted to specific pre-existing classes like car and truck— is a crucial step in this research because most objects are classified as unknown, which hinders the input process to the traffic simulation. The random forest model, a supervised learning algorithm, is employed for this purpose. All information combined in the array of data points, except the class column, is used as features, while the class column serves as the label. This enhances the model's performance in accurately reclassifying the unknown objects. Subsequently, an array of data points that have undergone the reclassification procedure are uploaded into the PostGIS database. The final database includes the original information from the parsing process, object and trajectory type, feature engineering attributes including max-min speed, acceleration, deceleration, and new class assigned after reclassification. The database schema is automatically created if it does not exist. The transformation of the array of data points into a data structure suitable for the PostGIS database is performed, including the transformation of the geo-coordinates into geometric points, as well as the incorporation of weather details from weather API (Application Programming Interface) based on those geometric points and timestamp to further enrich the dataset.

The SUMO traffic simulation then retrieves and transforms the data into XML format from the database for its input. The developed function used in this process is designed to fetch real-time data, seamlessly delivering it from the database to the traffic simulation. The purpose of the simulation is to replicate real-world traffic conditions using .osef data, enabling monitoring and traffic assessment. Two different approaches are considered in this traffic simulation process: by using XML tree libraries and by using TraCI to dynamically add new vehicles to the simulation according to the database inputs. TraCI, an open-source component of SUMO, can establish a dynamic connection between the simulation and databases, enabling real-time adjustments. The overall methodology is designed to develop a pipeline that is able to process the .osef dataset and deliver compatible data in an XML and binary format for the SUMO traffic simulation.

After the initial traffic simulation is properly configured, several what-if scenarios is explored. The scenario testing in this research evaluates various junction configurations, including roundabouts and existing intersection proposals, both with and without Traffic Light Systems (TLS), to understand their impact on traffic flow. Furthermore, the DT framework is assessed by comparing simulated traffic with real-world observations using statistical measures like Root Mean Square Error (RMSE) and cosine similarity for trajectory accuracy. These evaluations ensure that the simulation accurately mirrors real-world dynamics, providing valuable insights for urban planners and traffic engineers. Following the methodological workflow, a complete data pipeline is produced, encompassing structured architecture designed to process initial data, store it in the database, and subsequently retrieve it for integration into the simulation to be used for the real-time data stream in the future.

3.1. Local coordinates transformation

Cartesian coordinates or a local pose array represent the primary coordinate type that is available in the .osef dataset. Initially, this dataset lacked the global Coordinate Reference System (CRS) geo-coordinates necessary for further spatial analysis. Therefore, the dataset needs a transformation process to convert its local pose coordinates into global CRS geo-coordinates. This transformation is essential to enable comprehensive spatial analysis and ensure seamless integration with existing geographical data

frameworks, including the alignment with SUMO road network. To achieve precise conversion from local coordinates to the desired geographical coordinates (latitude and longitude), the process utilized Well-Known Text (WKT). WKT is a standardized text format used to define the geographical properties and geometries of the LiDAR dataset. It facilitates accurate spatial referencing, ensuring that the transformed coordinates are correctly interpreted and aligned with geographic standards (refer to Annex 1: Study Area WKT information for the details).

With given WKT information, there are multiple processes involved in converting local coordinates (x, y, z) to geographical coordinates $(longitude, latitude)$ using PyProj. A Python interface to PROJ, a general coordinate transformation software library, is used in this process using the PyProj package. The position of a point in three dimensions, where $x, y, and\ z$ represent distances from an origin in a localized coordinate system, are converted to a position on the surface of the Earth, which is represented by latitude (north-south position from the equator) and longitude (east-west position from the Greenwich meridian).

The local pose, which refers to point or object position and orientation within local coordinate systems, is transformed directly into WGS84 coordinate systems using WKT information. To ensure the alignment with SUMO road network projection, which follows an ellipsoid rather than a 2D plane projection, EPSG:4326 is determined to be the appropriate coordinate system for this research. EPSG:4326 uses an ellipsoid to model the surface of the Earth. The transformation procedure requires a datum shift to adjust the coordinates from the local datum relative to WGS84.

The transformation of local pose to geographical coordinates involves converting the Cartesian coordinates, from the LiDAR sensors, into global Cartesian coordinates through rotation and translation process. The method and parameters of these transformations are detailed in the documentation by Outsight (2022). The earlier transformation is then followed by conversion of the global Cartesian coordinates to geographical coordinates $(longitude, latitude, altitude)$, this process involves using an ellipsoid model of the Earth (WGS84).

This conversion is done using geodetic libraries of PyProj. PyProj performs straightforward coordinate reprojection for each pose array $(x_1, y_1, z_1, \dots)$ in the array of data points utilizing the WKT information. This involves applying translation and rotation parameters, such as Helmert and Molodensky-Badekas transformations, to align the local frame with the global coordinate system. The conversion involves projecting the Cartesian coordinates onto the Earth's ellipsoidal surface accurately through geographic offsets and projection techniques (Alcaras et al., 2020).

This multi-step process is crucial for accurate spatial data integration, as illustrated in the Figure 8 on the next page from de By et al. (2004). The figure effectively illustrates the transformation of local Cartesian coordinates to global Cartesian coordinates and subsequently to global geographic coordinates (WGS 84) which supports the transformation process concept and serves as a valuable reference for understanding the theoretical and practical aspects of coordinate transformation.

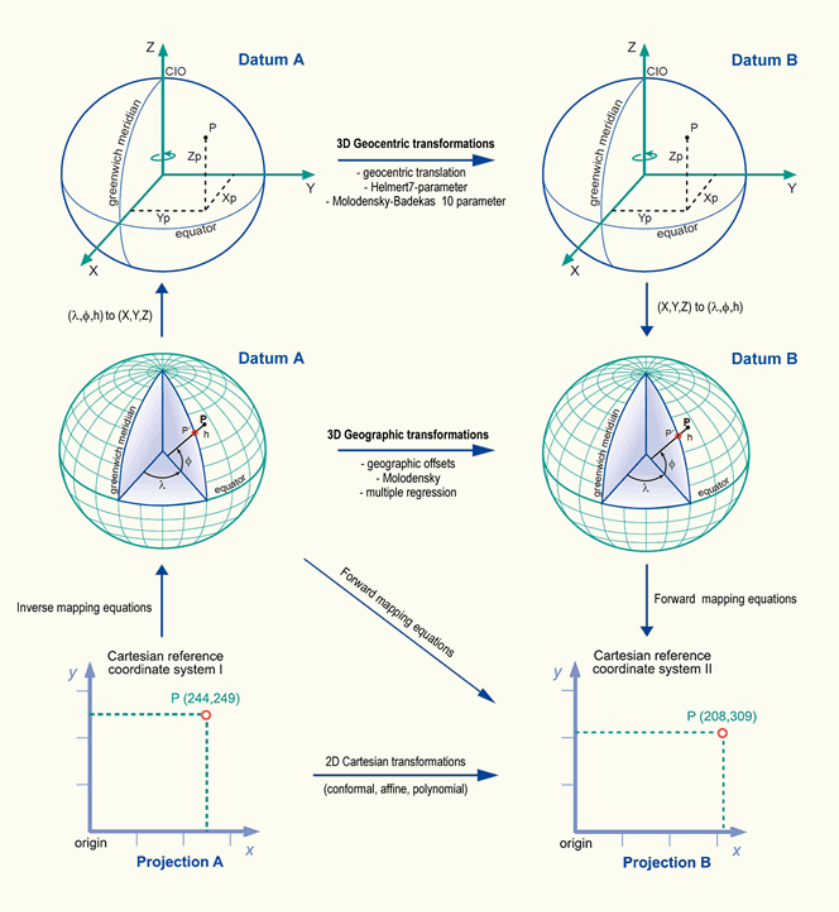


Figure 8. Overview of several types of coordinate transformations; this figure shows the 3D geocentric transformations (top section) and geographic transformations (middle section) relevant to converting local Cartesian coordinates from LiDAR sensors to global geographic coordinates. Source: de By et al. (2004), ITC core book, Principles of geographic information systems (Chapter 4.2 on spatial referencing).

A global Cartesian reference frame is aligned with local coordinates from LiDAR sensors through the required rotation and translation processes, as shown in the top section that demonstrates 3D geocentric transformations between various datums. Using the WGS84 ellipsoid model, the conversion from global Cartesian coordinates to geographic coordinates is displayed in the middle part. This entails precisely projecting the Cartesian coordinates onto the surface of the Earth using forward mapping equations and geographic transformations.

Particular planar projections, including conformal, affine, and polynomial transformations, are covered in the bottom half of the illustration, which is devoted to 2D Cartesian transformations. These kinds of transformations are frequently used in mapping and geographic information system (GIS) applications to convert between various planar coordinate systems. These two-dimensional transformations are significant within their context, but they are not directly related to the PyProj transformation process that converts local Cartesian coordinates to global geographic coordinates (WGS84). The more intricate 3D geocentric transformations and ellipsoidal projections required for precise geospatial data integration are mostly handled by PyProj. Thus, while the lower part is helpful for planar transformations, it is not essential to comprehend the 3D coordinate conversion procedure needed to convert local pose data to a global CRS such as WGS84.

By focusing on the top and middle sections of the figure, the essential steps of transforming local LiDAR data into global geographic coordinates are captured. The top section highlights the initial rotation

and translation processes. The initial rotation and translation steps needed to match local coordinates with a global reference frame are shown in the top section. The middle section that follows shows how to use the WGS84 ellipsoid model to convert these global Cartesian coordinates to geographic coordinates. Accurate geographical placement is ensured by this sequential transformation. These sections —illustrated in Figure 8— are essential to comprehending how PyProj uses the WKT information (WGS84) and geodetic principles to reproject local pose data precisely and accurately into a globally referenced geographic coordinate system.

3.2. Road segmentation

Road segmentation plays an important part in initiating the process of classifying objects and their trajectories. It allows the understanding of semantic information about the road edges at the intersection, which aids in the spatial identification of areas classified as roads from the ones that are not. In order to correctly classify object trajectories, the key objective of semantic segmentation is to identify the main lanes and junctions. During this process, roads are segmented into polygons, following the real-world shape of the intersection, and traced from satellite imagery using ArcGIS software. The main focus is to identify the main road segments, the turning point, the lane distribution details, and the junctions that existed in the study area. The output is stored in a .gpkg format, ensuring compatibility for the next spatial analysis process. The segment name and junction type constitute the semantic information structure; segment names are modified in accordance with the SUMO road network naming format. An illustration of this segmentation process can be seen in Figure 9.

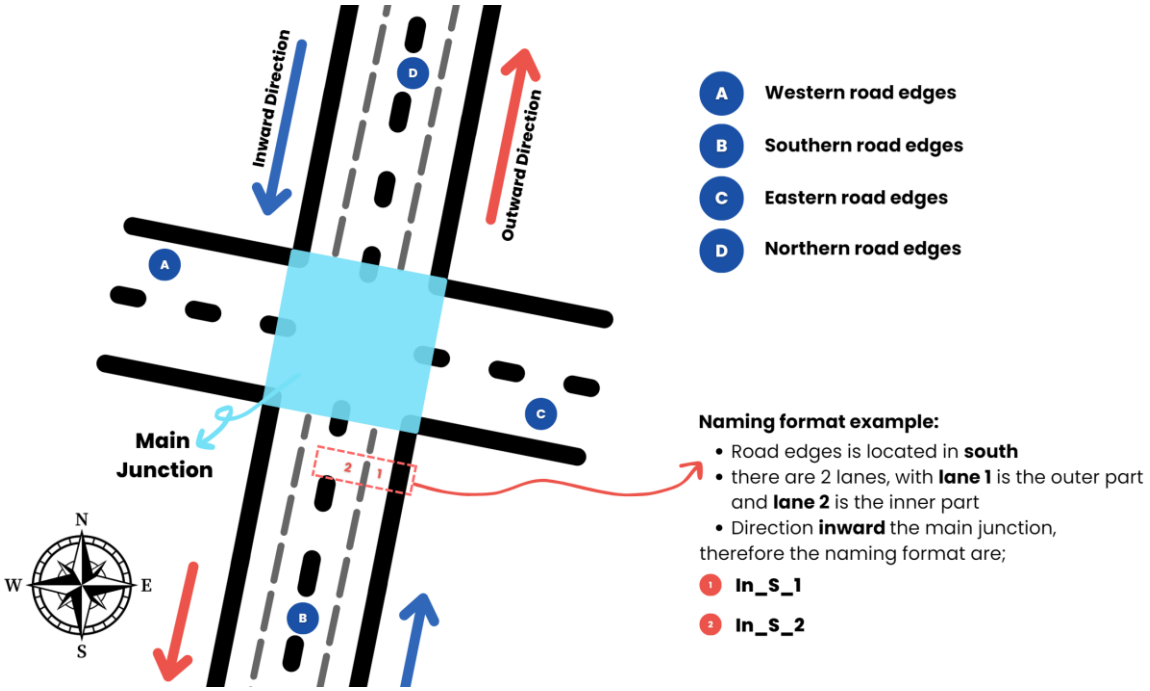


Figure 9. Road segmentation naming process. Source: Author (2024).

The naming format for the road segments is "In/Out_name_lanenumber". "In/Out" indicates the direction of the road flow, i.e., whether the segment is travelling inward or outward from the main intersection junction. "Name" indicates the road edges' name based on their location (for example, if they are in the north, their name will have a "n" affixed to it), and "lane number" indicates the exact lane number of those road edges. Lane numbers (e.g., 1, 2, 3, ...) are assigned in ascending sequence from the outside to the inner portion of the road boundaries. When it comes to junction types, they are referred to

according to the type of junction, such as intersection or fork (a junction where several roads merge into one or diverge into more than one). The key components of the road segments naming format can be summarized in Table 4.

Table 4. Summaries of road segmentation key components. Source: Author (2024).

Component	Description
In / Out	Direction of road flow, whether it is inward or outward from the main junction of the intersection.
Name	Identifier of the road edges based on geographic orientation or cardinal position (e.g., "n" for north or "s" for south)
Lane Number	Sequential lane numbers from the outermost to the innermost part of the road edges. (e.g., "_1" indicates the first lane which is most likely located in the outermost part of that particular road edges)
Junction Types	Junctions are named based on their type and function; for the main junction where the roads meet, it is indicated as "Intersection" and others are "Fork" to indicate the road splits or converging junctions.

In addition to the road segments' naming format and junction types, the segmentation process includes detailed information about the allowed lane connections, which further boosts the accurate representation of the intersection. For instance, certain lanes may allow only straight-through movements in the main junctions, while others may permit left or right turns. As illustrated in Figure 10, lane connection rules are clearly defined; for example, the source lane in the green is connected to the lane in the west road edge and south road edge, with two possible lane connections coloured as red. This semantic information of the allowed connections is derived from real-world traffic rules and is indicated by the directional arrows in each lane, as observed in the satellite images.

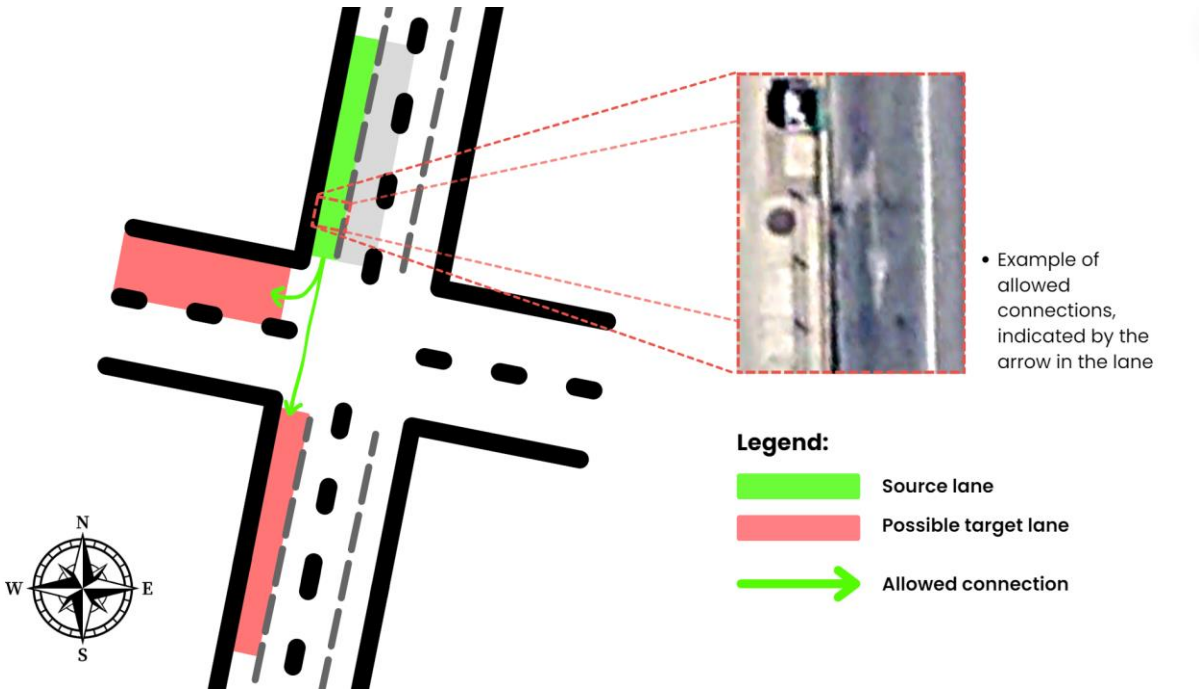


Figure 10. Defining connection rules. Source: Author (2024).

These connection rules are important for the map-matching process for incomplete trajectories. By incorporating detailed lane connection information, the map-matching process becomes more efficient in determining the possible connections for missing parts of an object trajectory. This ensures the vehicle paths are correctly reconstructed, even when partial data sequence is unavailable. This road segmentation process not only helps in creating a detailed and organized representation of the intersection road network but also significantly enhances the accuracy and integration process of the traffic simulation. By representing roads as detailed polygons, the segmentation allows for a more precise mapping of vehicle trajectories, which in turn leads to more realistic simulation results.

3.3. Vehicle trajectory spatial analysis

Spatial analysis of objects is carried out using the outcomes of road segmentation and local coordinate transformations. Two components are included in this geospatial analysis: the first is object type categorization, which involves categorizing possible vehicles (due to the fact that a person class might be found inside the road network) or non-vehicles; the second is trajectory type categorization, which categorizes the points into their identified trajectories type comprised of complete trajectories, (short) complete trajectories (complete trajectories but with start/end points being too close to the main junction), violations, and incomplete trajectories. The geospatial analysis approach consists of determining where the road segmentation intersects with the frame sequence for each object ID that generates unique trajectories. The main library used in this approach is Geopandas, which allows easily to handle spatial data. The detailed predefined process can be seen in Figure 11 bellow.

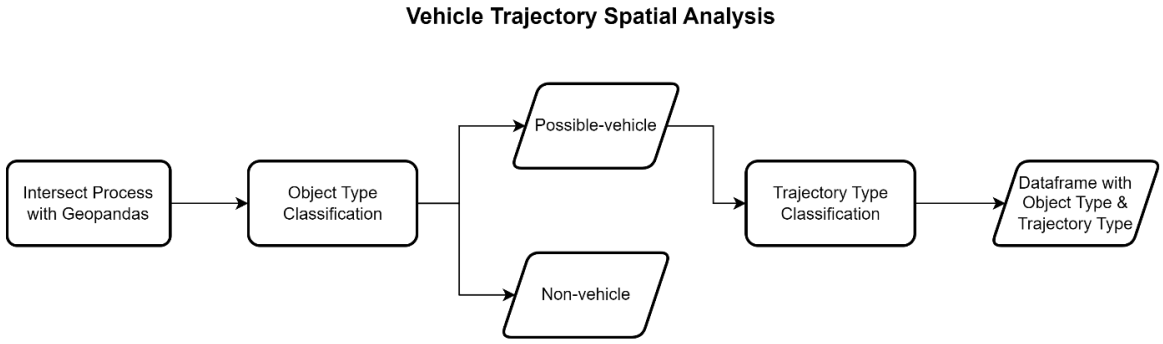


Figure 11. Vehicle trajectory spatial analysis predefined process. Source: Author (2024).

The logic behind object classification is the following: an object ID is classified as a possible vehicle if the majority of its geo-coordinate sequence $(Lon_1, Lat_1, Lon_2, Lat_2, \dots)$ intersects within a road segment (such as a lane or junction). This suggests that although it is presumably a vehicle, it might also be a person. On the other hand, it is categorized as a non-vehicle or a person if the majority of the data-point sequence of geo-location does not traverse the road segment.

Trajectory-type classification is more complex than object-type classification. Trajectory type classification is more refined, considering semantic road segment information and the need to generate multiple unique instances, leading to a detailed classification process. One of the instances is when the object data-point sequences are primarily composed of "non-vehicle" object types; in this case, their trajectories are promptly classified as "none" because they do not spatially align with the road segment.

As for the "possible vehicle" classification, the objects that fall under that category are filtered and assumed as vehicles due to their frame sequence located inside the road segments. Subsequently, the existence of particular segment types (inward, outward, intersection, and special turns) is checked and is used to classify the trajectory type. An analysis is performed on the "possible vehicle" trajectory's

beginning and ending point inside the frame sequences, including whether it starts with an inward segment and concludes with an outward segment or a specific turning point. This helps to determine whether a trajectory is complete or not.

If an object ID geo-coordinate sequence follows a pattern of (in, junction, out) , their trajectory is classified as complete due to the fact that the object follows the full pattern of going from inward road segments, main junction, and ending in outward road segments. In addition, the logic function uses a threshold of 8-meter proximity from the main junction to distinguish between complete and short trajectories when calculating the distance to an intersection. In traffic simulation, the (short) complete category is required for XML adjustments.

Additional assessments involve looking for unexpected direction changes that could point to possible rule violations and same-side violations, which occur when a trajectory starts and ends on the wrong side of the road. For example, if an object follows a sequence of (in, junction, out, in), it means that the vehicles made an illegal turn at the end of the sequence. Another example would be same-side violation, where an object has a sequence of following inward road segments to the main junction. However, they end in outward road segments that have the same segment name as their first points inward segment (e.g. from in north to out north), which indicates same-side violations. This meticulous approach ensures that trajectories are appropriately identified and saved as a separate column in the main array of data points.

3.4. Unknown class reclassification

Given the issues associated with many objects that are labelled as "unknown," reclassification using the Random Forest (RF) algorithm as a supervised machine learning approach is used. This process is considered reclassification due to the nature of unknown as inherent classification found in .osef dataset; alongside to more specific classes such as car, truck, two-wheeler and person. It is considered to be appropriate for handling complicated information with a lot of uncertainties, notably volume, speed, and the trajectory and object types that come from spatial trajectory analysis. RF is able to deal with mixed data types in a dataset, especially when the dataset contains both categorical (e.g., object type and trajectory type) and numerical (e.g. speed and volume) data. Additionally, RF reduces the possibility of overfitting by using an ensemble approach, in which the results are influenced by several decision trees, producing broadly applicable predictions. This means that the predictions generated by Random Forest are not only applicable to the specific data used for training but also have relevance and reliability across a wider range of datasets. Furthermore, RF provides insightful information about the significance of features, making it easier to identify which features have a major influence on reclassification. The main features selected are the speed and volume patterns, as well as the patterns of object type and trajectory type that are shown across the array of data points. To improve prediction precision and enrich the model's training data, feature engineering is performed. Therefore, the average speed and volume for every unique object ID, as well as changes in volume, is calculated. The predefined process of the random forest reclassification can be seen in Figure 12 (next page).

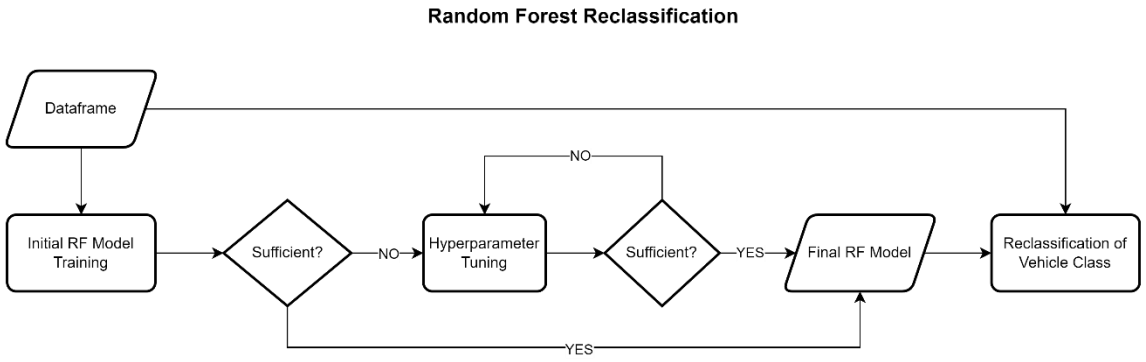


Figure 12. Random forest reclassification predefined process. Source: Author (2024).

The dataset for training the reclassification model is split initially into a 70% training set and a 30% test set. The dataset is split using scikit-learn's train_test_split function, with a 70-30 ratio and a random_state of 42 to ensure reproducibility. The test set is then further split into a 15% test set with an unknown label and a 15% test set without an unknown label. The reason for dividing the test set into two different test sets is to check the model prediction towards unseen data with unknown labels and without. The model should be able to predict the vehicle's unknown class into more specific pre-existing classes (car, truck, two-wheeler, and pedestrian).

Refined class adjustments are used to resolve the intricacies and irregularities in object labelling within the dataset throughout the RF model's training process. For recognized objects that show a combination of "unknown" and a distinct class, this adjustment method actively eliminates contradictions by prioritizing precise, informative labels over the ambiguous "unknown" labels. By using this approach, a more uniform and representative training set is ensured, which improves the model's ability to recognize real underlying patterns and raises the predicted accuracy of the model for previously "unknown" cases.

Label Encoder is used to encode the target variable for these refined classes, while One Hot Encoder is used to encode categorical features such as "Object_type" and "Trajectory_type". Class labels are numerically transformed using Label Encoder so that the RF model can categorize them without proposing an ordinality. Contrarily, the One Hot Encoder transforms categorical data (such as object type and trajectory type) into a binary matrix. This approach prevents numerical order implications between categories and guarantees that each category is treated independently by the model—a crucial step in accurately representing the influence of each category on classification.

RF processes data internally as arrays, with each row frame denoting a data point and each column representing a feature. A random subset of features is considered at each tree split during the training phase when feature arrays are randomly selected (with replacement) to form subsets for training individual trees. The process of classifying new data point (x) for the reclassification issues, where there are N trees and classes as C, can be defined as follows:

$$\text{Predicted Class} = \underset{c \in C}{\text{argmax}} \sum_{i=1}^{N} I(\text{prediction of tree}_i(x) = c), \tag{5}$$

where I is an indicator function that evaluates whether the prediction of ith decision tree (tree_i) for the data point x matches the specific class c. If it is true, the indicator functions return 1, otherwise 0. Based on all the trees in the forest, this equation indicates that for data point x, the projected class has received the most votes. To further improve the RF model, it is common practice to check and evaluate

the model performance, and the model performance can be developed further by utilizing methods such as hyperparameter tuning. These processes are explained in the next sub-sections.

3.4.1. Evaluation metrics

To evaluate the performance of the RF model for the reclassification of unknown classes, several standard metrics are used to ensure that the models is acceptable and perform well for future unseen dataset. These metrics include accuracy, precision, recall, and F1-score. Each of these metrics offers unique insights into the model performance and its ability to accurately reclassified the unknown classes to known labels/classes such as car, truck, two-wheeler, or person. The following Table 5 summarizes the equations used to calculate these metrics.

Table 5. RF model evaluation metrics. Source: Breiman (2001).

Metrics	Description	Equation
Accuracy	Shows the proportion of correctly classified instances among the total instances of the RF trees	$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$, (6)
Precision	Indicate the ratio of true positive results to the total predicted positive results	$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$, (7)
Recall (Sensitivity)	Shows the model ability to identify all actual positive instances	$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$, (8)
F1-Score	Indicate the balances of precision and recall by calculating their harmonic mean	$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$, (9)

To further explain the context of the table above, true positives (TP) indicates the instances where a label is correctly predicted, in this research case is where the known classes are predicted back to those specific known classes (e.g., a car correctly classified back as a car). True negatives (TN) are the instances where the model correctly predict that an object does not belong to a specific known class (e.g., it is correctly identifying an object as not a car when the actual class is a truck, two-wheeler, or person). False positives (FP) are instances that are opposite to TP, where the model incorrectly classifies an object as a specific class (e.g., a truck incorrectly identified as a car). As for the last, false negatives (FN) are instances where the models are unsuccessful in identifying an object as a specific known class; instead, it is classifying it as another class (e.g., car incorrectly identified as a truck).

To complement these metrics, a confusion matrix is used in this RF model evaluation. The confusion matrix provides a detailed breakdown of TP, TN, FP, and FN for each known class. This detailed breakdown allows for the identification of specific areas where the model may be misclassifying instances, such as cars being misclassified as trucks. It is useful to better understand how the model is performing across different classes, and where it might be making mistakes.

For further evaluation and to ensure the robustness of the RF model, a k-fold cross-validation is conducted, specifically 10-fold, which is used to generate a learning curve. This method involves dividing the dataset into 10 equal subsets, training the model on 9 subsets, and testing it on the remaining one. This process is repeated 10 times, and the results are averaged to produce a model performance estimation. This can help in identifying whether the model performs consistently across different subsets of data and ensure that it generalizes well to new unseen data. The learning curve generated from this cross-validation process provide better visual insights into how the model accuracy improves with more training data and whether it is overfitting or underfitting, and visually, the generalization of the model can be seen better.

3.4.2. Hyperparameter tuning

Hyperparameter tuning is also considered in the reclassification, depending on the performance of the model and how well they can effectively adapt the model to accurately reclassify the unknown classes. Metrics such as the number of tree (n_estimators), maximum depth of the tree (max_depth), minimum sample split (min_sample_split), and the number of features (max_features) are considered. The detailed description of each of these parameters is as follows;

- n_estimators: This parameter defines the number of trees in the forest. Increasing the number of trees can improve model performance by reducing variance, but it also increases computational complexity and training time.
- max_depth: This parameter limits the maximum depth of each tree. Controlling the tree depth prevents overfitting by restricting the model from learning overly complex patterns that may not generalize well to new data.
- min_samples_split: This parameter specifies the minimum number of samples required to split an internal node. Setting a higher value can prevent the model from learning small, noisy patterns, thereby enhancing its generalization capability.
- max_features: This parameter determines the number of features to consider when searching for the best split. It influences the randomness and diversity of the trees in the forest, impacting model performance.

To determine the optimal combination of these hyperparameters, parameter grid search techniques are employed. Grid search involves systematically testing a predefined set of hyperparameters across various combinations and evaluating the model's performance for each set. This exhaustive search helps in identifying the hyperparameter values that result in the best performance metrics.

During the grid search, the model's performance is evaluated using cross-validation techniques, ensuring that the hyperparameters are chosen to provide robust results across different subsets of the data. This method helps prevent overfitting and ensures that the model generalizes well to unseen data. The use of cross-validation also provides a more accurate estimate of the model's performance by averaging results over multiple iterations. By meticulously tuning these hyperparameters, the Random Forest model can be effectively adapted to accurately reclassify unknown classes, improving its predictive accuracy and overall reliability.

3.5. Database processing

After local coordinate transformation, trajectory spatial analysis, and reclassification of the unknown objects are done, the enriched dataset is then uploaded into database. The database choice in this research is PostgreSQL with a PostGIS extension. Given the georeferenced nature of the dataset, which includes detailed spatial coordinates and trajectories, PostGIS is employed to enhance data management and retrieval. PostGIS, an extension of PostgreSQL, allows for efficient storage, querying, and manipulation of geographic information, making it an ideal choice for handling complex spatial datasets. This geospatial database facilitates the integration of data into the SUMO traffic simulation environment.

To enable interaction with the database, the SQLAlchemy library is utilized. SQLAlchemy is a comprehensive SQL toolkit and Object-Relational Mapping (ORM) library for python, which simplifies the interaction with the database. With the utilization of this library, the traffic simulation able to retrieves data directly which enables efficient data fetching and integration with the traffic simulation DT framework. This integration ensures that the simulation has access to accurate and up-to-date spatial information, which is essential for this research. This stage of database processing consist of both data

storing and data fetching of the enriched LiDAR datasets. The details of these processes are explained in the following sub-sections.

3.5.1. Data storing process

The processed .osef dataset is going under a systemic process to ensure the storing process in a database, utilizing SQL commands combined with additional python logic. This process involves generating the necessary database and schema if they are not already present and continuously update the database based on the new array of data points timestamps. Python scripts are used to check the existence of the database and schema, generate new one if not present, and ensure that new data entries are appended seamlessly. This ensures that the dataset remains current and organized for future retrieval and analysis.

Before the array of data points is stored in the database, a map-matching process is done in order to complete the information of object ID that is categorized as having incomplete trajectories from the spatial analysis stage. Due to the nature of SUMO, objects classified as incomplete trajectories cannot be properly loaded into the simulation. Therefore, it is important to transform them into complete trajectories. This process uses the present .gpkg data of the intersection as the base information. This approach adopts the sumolib map-match algorithm concept but instead of using road network XML data, it uses the .gpkg data information instead. The .gpkg data includes semantic information detailing the permissible movements within each lane, such as which lanes allow straight travel, left turns or right turns. This enhance the robustness of the map-matching process, which makes this approach hybrid, by adopting the original map-match algorithm from sumolib, incorporated with the semantic information of the .gpkg as extra logics.

By using this hybrid approach, the map-matching process significantly reduces potential errors that might happen from relying solely on the geometric proximity of the data points to the road edges, which is what the sumolib map-matching algorithm originally does. Relying only on the geometric proximity often results in misidentified paths, especially in complex intersections or when the data points are sparse. When the starting or ending of object ID trajectory sequences are missing, the connection information helps predict the most likely continuation or origin of a path. The inclusion of connection information able to limits the possible connection and filters out the incorrect paths in the proximity of the data points. This ensures the most likely path that is chosen for incomplete trajectories. This hybrid approach is especially important for ensuring continuity and logical consistency in the trajectory data, further enhancing the dataset's reliability.

Subsequently, the dataset is further enriched with weather information. This is accomplished by integrating with the OpenWeatherMap API, an open-source weather API. By using the geographic coordinates and timestamps of each point in the dataset, relevant weather data is retrieved and appended to the dataset. This additional weather information adds another layer of context and value to the data, providing insights into the environmental conditions at the time each data point was recorded; this also affect the configuration of the simulations, especially the speed limit, to further mimic the real-world traffic behaviour.

3.5.2. Data fetching process

The method for fetching data from the database to the SUMO traffic simulation involves multiple steps and two main approaches that has been mentioned before, which are XML-based and TraCI-based. XML Generally, the fetching process follows similar step, the connection to the PostGIS database is established using the specific credentials after the SUMO tools directory environment setup are ensured to be present. These credentials includes username, password, hostname, port, database name, schema and table. In this research localhost are used with duplicate port of 5433 in the local system. This connection

including the password is URL-encoded for security and a connection string is created using SQLAlchemy "create_engine" function. This allows for the interaction with the database more efficient.

A mapping dictionary is used to associate object classes (e.g., car, truck, motorcycle, and pedestrian) with their corresponding SUMO object types and colors. Additionally, a function to adjust lane information for non-pedestrian objects to ensure they follow realistic routing within the simulation is implemented. This involves replacing lane numbers based on specific exceptions (in SUMO lane number zero are for pedestrian), making sure that vehicles adhere to logical lane assignments.

The main function of the data fetching processes the routes by vehicle class. It starts by identifying the base departure time from the dataset timestamp. The data is then grouped by vehicle class and object ID. For each group, departure times are calculated, this includes departure and arrival road edges, as well as the positions are determined using the geom $(long, lat)$ information of the data points. Trajectories are analyzed to extract intermediate edges, ensuring duplicates are removed, and only valid paths based on the object ID trajectory information.

Following the previous processes, a function formats the XML output to ensure it is well-structured and readable. For pedestrians, specific attributes are set to account for walking paths (lane zero). For vehicle types, attributes such as departure speed, arrival speed, and acceleration are included. The XML files are then saved, providing complete and extensive route information that is ready for use in SUMO. As for TraCI-based approach, it is directly looking at the necessary data from the database to add the vehicles in the simulation itself, this is limited to the details of the departure points and arrival points in the road edges. However, this allows for more faster real-time interaction with the simulation without storing data locally, which is an advantage compared to XML-based approach. The configuration and integration details of this data fetching process, including the specific settings and parameters used for SUMO is explained in the next section.

3.6. Traffic simulation

In this traffic simulation stage, after the initial traffic simulations are developed, several what-if scenarios is generated to analyse the intersection traffic under various conditions. This section explains the integration and initial configuration of the SUMO traffic simulation with the enriched data stored in the database. It also covers the testing method of different what-if scenarios. Figure 13 on the next page shows the predefined process of this stage.

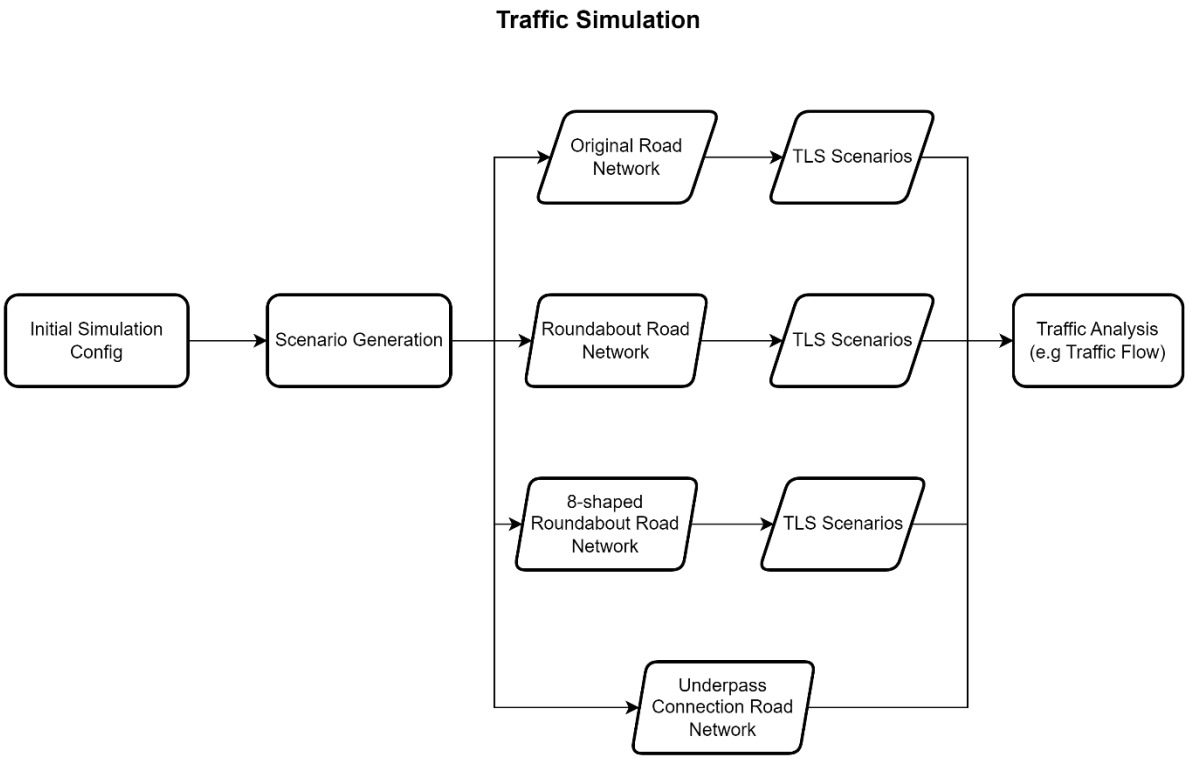


Figure 13. Traffic simulation predefined process. Source: Author (2024).

In this research, the focus is on analysing whether changing the junction into different shapes, such as roundabout, is significantly affect traffic parameters like traffic flow and count. The scenarios also explore other existing plan in the study area, such as transforming the intersection into an 8-shaped roundabout. Additionally, the scenarios include the situations where the intersections operates without a Traffic Light System (TLS) and with TLS. Each of the sub-sections that follows provides more explanation of these processes. This approach aims to understand the impact of different intersection design and configuration on traffic performance, providing valuable insights for traffic management and planning.

3.6.1. SUMO integration to DT and configuration

The traffic simulation platform SUMO runs entirely on XML as its foundational configuration. Therefore, the dataset is fetched from the database using the xml.tree libraries so that it can be directly exported into an XML format that the SUMO can handle. Two distinct processes comprise the traffic simulation workflow: a process that uses XML input for SUMO to initiate the traffic simulation, and a direct process that allows vehicles to be added to the simulation dynamically using TraCI libraries without requiring the data to be stored in XML format.

For XML based approach, data is directly fetched, transformed, and saved in the required XML format. This allows for the inclusion of complete information. Vehicles use the {<trip>} attribute, and pedestrians use the {<person>} attribute to define the XML configuration for vehicle input. In separate trips.xml files, every vehicle class is stored. The characteristics selected for the XML configuration are intended to match real-world behaviour as much as feasible while also considering the data that is contained in the database. The list below shows the details about each trips.xml configuration structure.

- Vehicle (car, truck, two-wheeler):

<vType id vClass accel decel sigma color**/>**

<trip id type depart from to via departLane departPos departSpeed arrivalLane arrivalPos arrivalSpeed**/>**

- Pedestrian (person):

<vType id vClass color**/>**

<person id type depart departPos**/>**

<walk edges speed arrivalPos**/>**

Depicting from the configuration structure above, the id uniquely identifies each object type, while vClass classifies objects into categories such as car, truck, person, or two-wheeler. The accel (acceleration) and decel (deceleration) are derived from speed changes over time, calculated from speed changes per timestamp. Vehicle driving behaviour is represented by sigma ranging from zero to one, where zero represents perfect driving behaviour, and one represents a high degree of randomness in the driving behaviour. As for object appearance, it is assigned by colour.

The depart attribute indicates the departure time, and the from and to attributes specify the starting and destination road edges, respectively. The departLane and arrivalLane attributes define the specific lanes for departure and arrival, while departPos and arrivalPos specify the exact positions within the simulation. As been mentioned before, for the target departLane and arrivalLane, the information is then fetched from the database in result of the map-matching process in the storing stage. Meanwhile, the input for departPos and arrivalPos are based on the data point position in the simulation space overlayed with the road network XML file. The departSpeed and arrivalSpeed attributes set the speeds at the start and end of the trip. Trajectory continuity for both vehicles and pedestrians is maintained through careful speed and position settings. Therefore, for (short) complete trajectories, the speed is set to "last" to follow the speed of the vehicle ahead, preventing simulation errors due to unrealistic acceleration or deceleration.

Dynamic approach using TraCI, on the other hand, does not store any data locally. This is because, for the duration of the simulation time step, the dynamic vehicle input function in TraCI only needs the base XML input, which, in this case, utilises the previous approach XML configuration. Following that, it adds vehicles continuously based on the timestamp. This approach has the advantage of enhancing performance by reducing the overhead associated with file or data-point handling and providing immediate feedback and control over the simulation environment. The creation of a road network in SUMO is a prerequisite for initiating the traffic simulation. Imported from OpenStreetMap, the base road network is modified to match the research area and the previously made road segmentation. This modification is imperative to complete the trajectories identified as incomplete by the spatial analysis, using the road network.

3.6.2. What-if scenario testing

The what-if scenario testing stage involves creating different junction structures to assess whether these long-term planning scenarios of changing junction shape affects the traffic flow. This type of analysis helps to further understand the potential impacts of major infrastructural changes. The scenarios reflect the potential effects well with the real-world condition due to the processed .osef data as an input, which consists of real-world traffic data recording in the study area. This ensures that the simulated scenarios mirror actual traffic conditions and able to show the possible outcomes.

Despite the nature of real-time data that typically follows its original trajectory that reflecting the real-world road structure, it is possible to test different scenarios in the SUMO traffic simulation. The .osef real-time data reflects the actual traffic conditions and movements, but the simulation can adapt even when the junction structure is changed and no longer mirrors the real structure of the intersection. This flexibility is made possible due to SUMO robust routing algorithms, namely Dijkstra and A* which has been explained in the 2.1.2 Traffic simulation models, on page 14, allows a dynamic calculation of routes for each object ID. When a junction is altered, these routing algorithms allow each vehicle to find its way through the modified network, if the key positions in the road network, such as the starting and ending segments, are specified. The algorithms can easily determine the optimal paths for the vehicles. This ensures that the simulation remains realistic and functional despite the significant changes in the road network.

In addition to modifying the junction structure, the scenarios includes a situation where each scenario implemented TLS function and what would happen if they is not implementing TLS. This approach gives better understanding if the traffic signals effects on different junction configurations. By comparing scenarios with TLS against those without it, the testing aims to provide insights into how traffic signals can influence traffic flow and congestion. Each scenario enable a comprehensive evaluation of the different junction shape including the existing proposed changes. Furthermore, this what-if scenario testing able to provide data-driven simulation that provide the potential benefits and drawbacks of each what-if scenarios, guiding to a better traffic management decision in a long-term planning.

To achieve the different scenarios of junction shapes, the process involves both the road network XML configuration and the use of NETEDIT, SUMO graphical network editor. First, using the road network XML configuration, the network layout, including roads, lanes, and junctions, is defined in the file. For instance, to convert the original intersection into a roundabout, specific changes are made in the XML file by defining a new type of junction and updating the connection edges and lanes to match the roundabout configuration. This involves specifying the roundabout junction and adjusting the necessary edges to ensure they connect properly to the newly changed junction type. As for incorporating TLS system, traffic light logic can be specified with various phases and timings to simulate different traffic signal operations. In this case, the default settings are used. The example of XML structure of each can be seen in the list below.

- Road network with changed junction shape:

```
<junction id="roundabout" type="roundabout" x="100" y="100">
        <request index="0" response="0"/>
</junction>
<edge id="edge1" from="A" to="B" />
<edge id="edge2" from="B" to="A" />
<!----……...……..---->
```

- TLS configuration:

```
<tlLogic id="junction1" type="static" programID="0" offset="0"/>
        <phase duration="31" state="GrGr"/>
        <phase duration="6" state="yryr"/>
        <phase duration="31" state="rGrG"/>
        <phase duration="6" state="ryry"/>
</tlLogic>
```

Alternatively, the graphical network editor NETEDIT can be utilized for a more visual approach. By loading the existing road network file into NETEDIT, the intersection intended for modification can be transformed into a different shape according to the user's needs. This approach helps better in creating more complex junction types that require detailed positioning, including geographical positioning and road connections. By employing these methods, different junction shapes and configurations necessary for scenarios testing can be modelled accurately within SUMO. For the detailed of the chosen scenarios in this research, please refer to the next sub-section.

3.6.2.1. Chosen what-if scenarios

The scenarios for different types of junction shapes that is explored and tested in this research covers standard roundabout junction shapes and two different existing proposals of the study area intersection changes. Roundabouts are circular junctions where traffic flows in one direction, counter clockwise, around a central island. Research has indicated that roundabouts can significantly reduce traffic delays and improve flow. According to study by the Insurance Institute of Highway Safety (IIHS), converting intersections with stop signs or traffic signals to roundabouts can reduce injury crashes by 72% to 80% and all crashes by 35% to 47% (IIHS, 2021). This improvement is largely due to the continuous movement allowed in roundabouts, which is able to minimise the stop-and-go conditions commonly applied in intersections. Furthermore, the roundabout is designed to lower vehicle speeds and reduce the severity of collisions, contributing to safer and smoother traffic conditions (IIHS, 2023).

In the context of this research, the roundabout scenarios are tested to observe these expected improvements in the traffic flow. Aside from the standard roundabout shape that is deployed as the first chosen scenario, different kinds of roundabout structures, such as the 8-shaped roundabout, is considered as the other scenario in this research. This 8-shaped roundabout has been proposed by the Sofia municipal councilor Borislav Ignatov. His proposal is to replace the study are intersection with double roundabout or 8-shaped roundabout. According to him, the current structure are not a very efficient and good thing, proven by numbers of crashes that often happen in the intersection of paradise mall center. By building this roundabout, the aim is to increase the level of safety and greatly reduce traffic congestion (Nikova, 2020). The proposed shape of this roundabout can be seen in Figure 14.

Figure 14. Proposed 8-shaped roundabout. Source: Nikova (2020), Арх. Игнатов предлага кръгово пред "Парадайс" в памет на Милен Цветков (СНИМКИ). Retrieved May 26, 2024, from https://stolica.bg/mestna-politika/arh-ignatov-predlaga-kragovo-pred-paradais-v-pamet-na-milen-tsvetkov-snimki.

As can be observed in the figure above, the proposed scenario consists of transforming the main junction into a double roundabout in an 8-shaped structure. This is shown with the existence of three islands in the main junctions. Another note is that there is a reduction of lanes that is visible in the north-east road where the inward edges, which originally had three lanes, are now reduced into two lanes. For the southern road edges, it is also visible that there is a reduction of lanes for each direction from three lanes to two lanes each. This is considered during the building of the scenario road network. Another scenario based on the existing proposal for Save Sofia (Спаси София) is explored as well. This proposal is not exactly located in the intersection of the study are of this research, however it is located in the neighbouring intersection, south of the study are intersections.

The proposed plan in this proposal involves creating an underpass for Blvd. "Todor Kableshkov", which allows east-west traffic to pass beneath the surface level, which basically involves the creation of an underpass. The surface-level intersection is still exist for north-south traffic on Blvd. "Cherni vrah". This structure ensures that the traffic that goes from east to west on "Todor Kbaleshkov" remains uninterrupted while accommodating the north-south traffic through the surface intersection (Спаси София, 2020). This design aims to reduce congestion in both directions. This proposal itself is adapted to the study area, by creating an underpass specifically in east-west traffic directions connecting "Hendrik Ibsen" street to "Srebarna" street. The aim of this is to see the potential of adding two-level intersections where one connection has a different elevation compared to the other and whether this type of scenario is possible to be done in the traffic simulation DT framework utilizing SUMO. The proposal for this two-level intersection can be seen in Figure 15 below. Point A shows the location of the study area intersection where this proposal is adapted, and point B shows the location of the original underpass proposal.

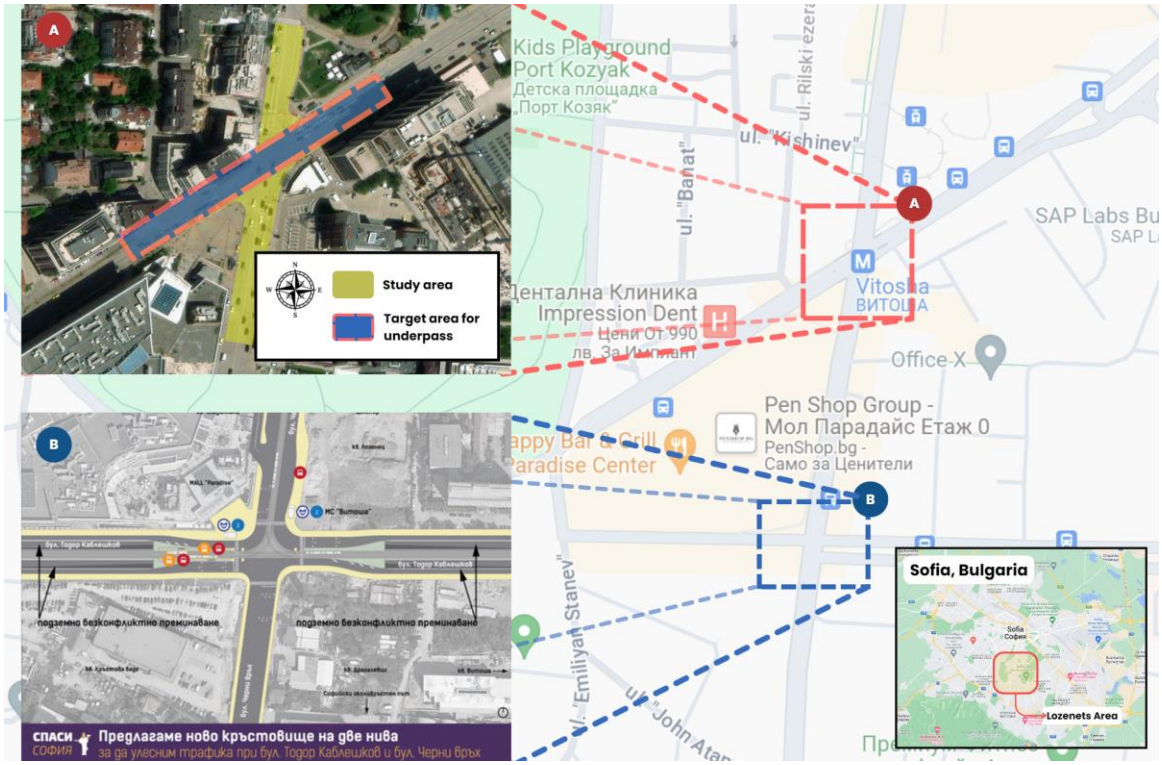


Figure 15. Two-level intersection proposal by "Спаси София". Source: Спаси София (2020), "Спаси София" предлага да се изгради ново кръстовище на две нива при МОЛ "Парадайс". Retrieved June 3, 2024, from https://www.novinite.bg/articles/195721/Spasi-Sofiya-predlaga-da-se-izgradi-novo-krastovishte-na-dve-niva-pri-MOL-Paradajs.

3.7. DT framework assessment

This stage focus on assessing the traffic simulation DT framework. One of the assessment methods is to validate the accuracy of the simulation towards real-world traffic; this includes checking the traffic volume or traffic count from the simulation compared to the real world. To achieve this, a comprehensive traffic count method is employed. The observation is conducted in the morning, when traffic is typically at its peak. Given the constraints of limited manpower and the complexity of the intersection, which has four road edges in total, each road edges have two different directions, with each consisting of more than two lanes. A strategic approach is adopted to ensure accurate and manageable data collection. Instead of attempting to count the traffic on all four road edges simultaneously, the observation is designed to take traffic counts for each road edge sequentially, with intervals of 10-minutes each.

The observation points for the survey are carefully chosen to match the positions of the induction loops in the SUMO network, ensuring that the real-world data corresponds with the simulation output. The illustration of the said observation points can be seen in Figure 16. For each road edge, the count is recorded for both directions; this includes vehicles that move inward towards the main junction of the intersections and those that move outward from the main junction. This simultaneous counting in both directions allows for an aggregated traffic count for each road edge during the 10-minute interval.

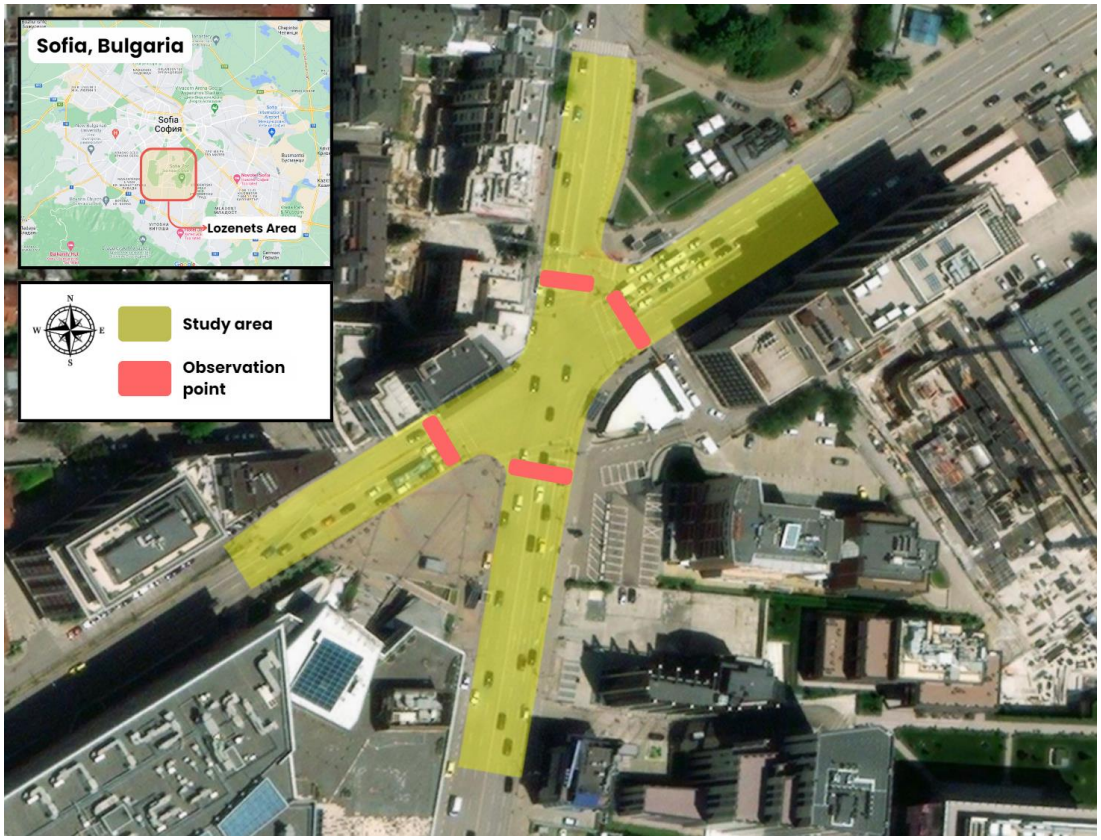


Figure 16. Traffic observation points. Source: Author (2024).

By aggregating the counts, for each of the road edge, the observation still able to provide a comprehensive overview of traffic volume, which can then be compared against the simulated traffic data to assess the accuracy of the traffic simulation DT framework. The real-world observation data is compared with the traffic count output of the SUMO after it has been properly collected. The simulation count is collected in the form of an additional XML format, the temporal factors is aligned with the

observation dataset collection timestamp, aggregated for each available four edges. This step is important to match the temporal and data format from the real-world traffic observations, ensuring consistency.

The accuracy of the traffic simulation is then assessed by comparing both datasets. The assessment part is done using commonly used statistical method, Root Mean Squared Error (RMSE). RMSE based on Farrag et al. (2020), measures the deviation of the simulation output from the observed data. The differences between the values predicted by the model and the observed values from the real-world observation is quantified in this process. By calculating the RMSE, the accuracy of the traffic simulation can be quantitatively evaluated, highlighting the effectiveness of the workflow in replicating real-world traffic conditions. The formula is as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\left(\frac{O - M}{O}\right)^2}. \tag{11}$$

where n is the number of observations, O is the observed value and M is the simulated value. In this case, the observation is done for each of the four road edges sequentially, with each road edge being observed for a 10-minute interval. Therefore, n in this context represents the total number of 10-minute intervals observed across all road edges. Since each road edge is observed for 10 minutes, the comparison timing from the simulation output must also match these intervals precisely.

Another aspect that is validated is the similarity of objects trajectory generated by the traffic simulation towards their true trajectory. This assessment is needed due to the SUMO traffic simulation requirement, where to load a vehicle, it needs to define the departure and arrival road segments and, optionally, a "via" attribute to indicate which road segments that the vehicle travels through. In comparison, the .osef dataset provides a detailed sequence of GPS trajectory data, with each frame showing the data points with precise timestamps, often down to the millisecond. This detailed level of data captures the exact path of the vehicle.

However, the input for SUMO is not designed for it to be that detailed compared to the .osef dataset. It is only requiring the names of the road segments for the XML attribute of departure and arrival; including the road segments in between arrival and departure attribute. The routing algorithm within SUMO uses this information to generate the vehicle trajectory. As long as the vehicles have the information on the specified departure edge, segments that they pass through, and the specified arrival edge, the traffic simulation able to create a trajectory sequence accordingly. This sequence might not exactly match the original trajectory from the .osef dataset due to the simplifications and assumptions made by the routing algorithm in SUMO.

Therefore, to evaluate the accuracy and similarity of the generated trajectories to the original .osef dataset recorded trajectories, cosine similarity is employed. This method check the vector of the generated trajectory from the simulation and determines the similarity with the actual object GPS trajectories. Cosine similarity is calculated as the cosine of the angle between two vectors in a multi-dimensional space, which gives a value between minus one and one. A cosine similarity of one means that the two vectors are identical, zero means they're orthogonal, and minus one means they are diametrically opposed. The equation are as follows, adopting from Manning et al. (2008) documentation,

$$\text{CoSim}(A, B) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}, \quad (12)$$

where:

- A_i and B_i are components of vectors A and B respectively.
- $A \cdot B$ is the dot product of vectors A and B.
- $|A|$ and $|B|$ are the Euclidean norms (magnitudes) of vectors A and B respectively.
- n is the number of dimensions.

3.8. Summary

This methodology chapter outlines the steps for developing the traffic simulation DT framework that integrates real-time LiDAR data into traffic simulations for effective traffic monitoring and assessment. The approach begins with acquiring OSEF real-time traffic data and network data from OSM, transmitted to a common database platform for continuous updates, data transformation, and filtering. The workflow involves reviewing existing literature, collecting and parsing the OSEF dataset into an appropriate CRS, and classifying object types. Trajectory spatial analysis is performed to classify moving objects' trajectories, providing features for the Random Forest model to reclassify unknown classes. Processed data is stored in a PostGIS database and integrated into SUMO traffic simulations using XML-based and dynamic TraCI methods. Various what-if scenarios are explored to evaluate different junction configurations and their impact on traffic flow. The digital twin framework accuracy is validated by comparing simulated traffic with real-world observations using RMSE and cosine similarity. This comprehensive methodology ensures a robust data processing and simulation pipeline, supporting real-time traffic monitoring and assessment.

4. RESULTS

This chapter presents the results of the research. The sections are ordered according to the research workflow and methodology. It begins with an overview of the parsed dataset, continued with the coordinate transformation process, followed by data integration and preprocessing. The simulation and modelling phase is then discussed, detailing the methods and findings. The chapter concludes with an evaluation of the traffic simulation DT framework, assessing the applicability of the research methodology.

4.1. Data parsing

Understanding the data structure is the primary step of this research. During the data parsing, the dataset is analysed to understand the overview of distribution and classification of tracked objects. As been mentioned before in the 1.4.2 Research datasets, on page 7, there is chances that the .osef dataset classified a tracked objects with multiple classes. Based on the analysis conducted using a sample of a two minute morning .osef dataset, it is found that there is approximately about 58,939 rows of data points present, with the maximum number of frames captured for an object ID being around 1726. Across these frames, a total of 516 objects is present. The detected classes within the dataset include car, person, truck, two-wheeler and unknown. These classes further divide the objects into three main groups: consistent objects, recognized objects, and unidentified objects. The ratio of these objects group can be seen in Figure 17.

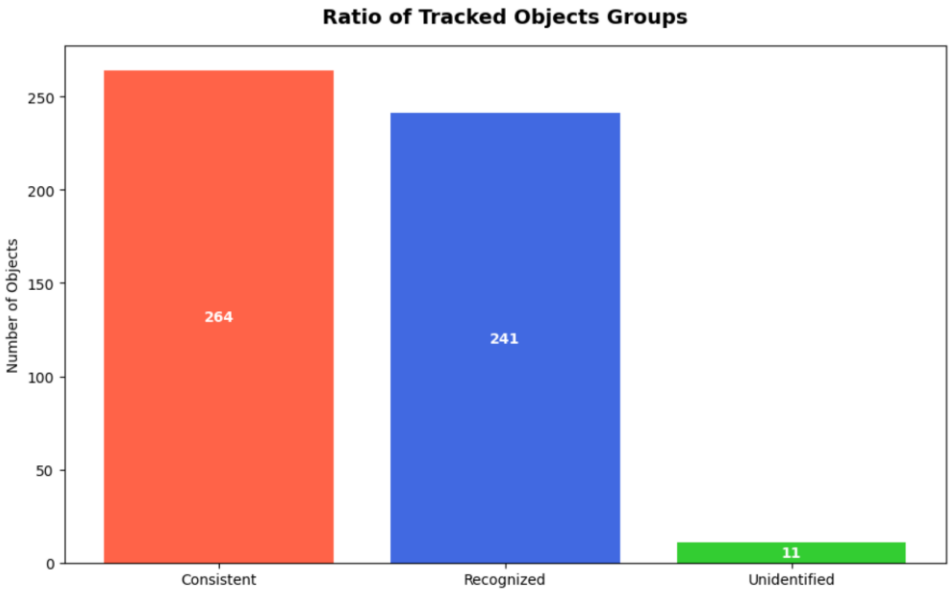


Figure 17. Ratio of tracked objects groups based on identified classes. Source: Author (2024).

Consistent objects, where an object ID has only one class throughout the frames, constitute as the largest group. It represents approximately 51% of total tracked objects in the sample data with 264 objects. Recognized objects group follows with 241 objects, accounting for around 47% of total tracked objects. This group consist of an object ID that has more than one classes with the other classes being unknown class. On the other hand, unidentified objects, where an object ID consist of more than two classes from the smallest category group, with approximately 2% of the total tracked objects. This

distribution indicates that the majority falls into the consistent and recognized categories. To further understand the classes distribution within these groups, especially consistent and recognized objects, Figure 18 provides an overview of the distribution.

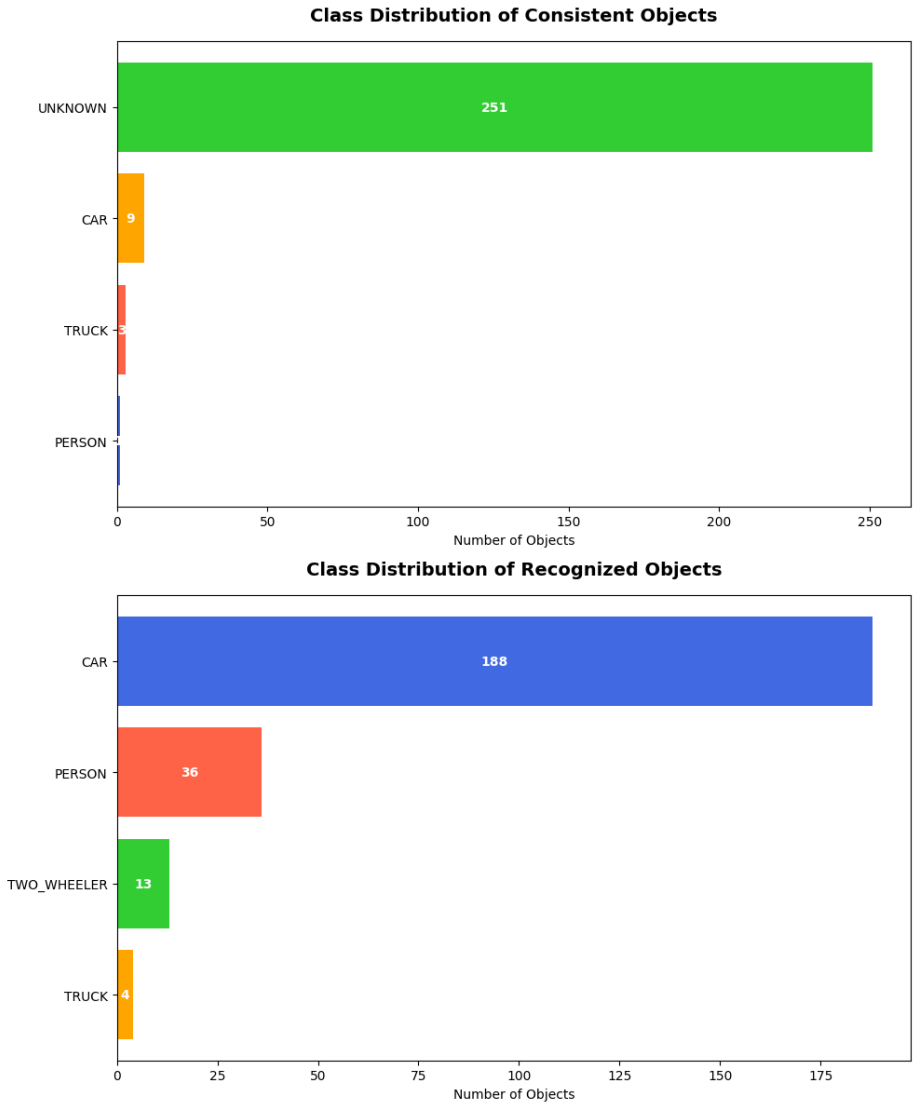


Figure 18. (Top) Consistent objects and (Bottom) Recognized objects class distribution. Source: Author (2024).

The figure above shows that the unknown class dominates in the consistent objects group. This indicate that majority of the tracked objects is not classified into specific known vehicle classes such as car and truck. The consistent objects in this sample data doesn't seems to be classified as two-wheeler. The prominence of unknown classifications, around 251 objects, suggest that an object ID that might be two-wheeler are classified as unknown instead. Depicting from the bottom figure, it can be seen that for recognized objects, majority of classification falls into car, with 188 instances. This is followed by person class with 36 objects, two-wheeler with 13 objects, and the truck class with 4 objects. For this group of classes, from the two classes they is assigned with, the other is unknown, therefore it is safe to assume that their actual classification is the counter part of the unknown class within that object ID.

In addition, the overview of the data parsing includes the information of unidentified objects group. The object ID categorized within this group (11 objects) have more than two classes, indicating inconsistencies in their classification process. Examples include objects with identification of 5324412,

5324431, and 5324597, which are classified in order from unknown class, two-wheeler, and then car or person. The detailed example of the overview can be found in Annex 2: .osef data overview sample. The structure of the datasets explained in this stage is used as the main foundation and consideration for the next analysis, such as the random forest reclassification. In the data parsing process, the next stage is to handle other known issues with the .osef dataset, such as the local coordinate that needs transformation to geo-coordinates, which are explained in the next sub section.

4.2. Local coordinate transformation

All of the arrays of data points had their local pose translation (x, y, z) are converted to WGS84 EPSG:4326 geo-coordinates. This process is done as part of the .osef data parsing algorithm. Where each frame's local coordinates are transformed into geo-coordinates, following the methodology section. The transformation is tested with the array of data points for static objects first, to determine the precision of the coordinate transformation. Afterwards, as the coordinate transformation process indicates a success, the local cartesian coordinate of the tracked objects is transformed into geo coordinates. The georeferenced data points of the static objects can be seen in Figure 19, visualized with Folium library.

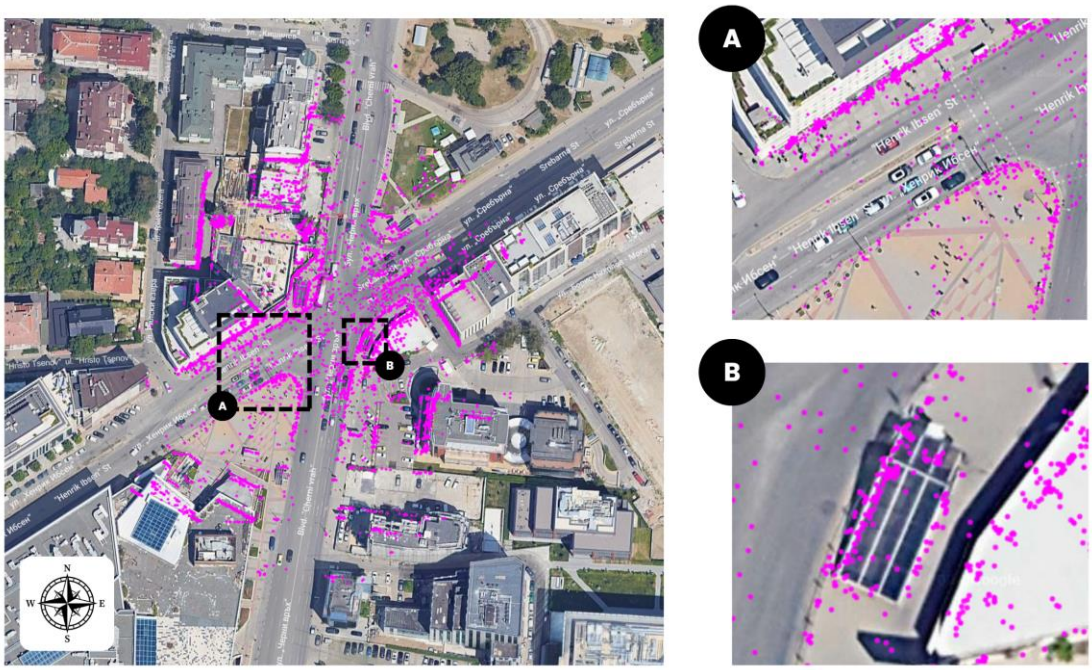


Figure 19. Static objects plotting of the geo-referenced data points. Source: Author (2024).

It can be seen in the figure above that the data points for the static objects is successfully transformed into geo-coordinates. The static objects data points aligns well with the structure of the intersection itself. In the example A, several data points clearly delineate the shape of the curb and road edges. Meanwhile, in example B, points align perfectly with an object in intersection that forms the shape of a rectangular building, specifically the metro train entrance.

Based on the successful transformation of the static object coordinates, the array of data points for tracked object, initially in cartesian coordinates, is then transformed into geographic coordinates. Table 6 on the next page shows the first five frames to demonstrate the transformation of the tracked objects data points by comparing the original local coordinates with the reprojected geographic coordinates (longitude, latitude, altitude). This conversion ensures that the data is accurately represented.

Table 6. Array of data points of reprojected coordinates. Source: Author (2024).

Object ID	Class	Speed/kmh	Volume/m3	Local Pose			Geo-coordinates		
				X	Y	Z	Longitude	Latitude	Altitude
5324219	truck	16.53	99.41	6.17698	-3.85191	1.75459	23.316452	42.659424	1.75459
5324294	unknown	2.58	8.16	-39.31805	-29.1799	1.45436	23.316156	42.65901	1.45436
5324224	car	0.02	17.94	-37.26698	-19.5937	1.384189	23.316113	42.659092	1.384189
5324304	unknown	7.76	7.76	-36.46281	-1.43824	1.421428	23.316	42.659232	1.421428
5324343	unknown	13.69	5.9	28.003428	2.333962	0.73473	23.316634	42.659577	0.73473
…..	…..	…..	…..	…..	…..	…..	…..	…..	…..

To further confirm that the tracked object points have been transformed successfully, the Plotly Express library's function called Mapbox is used. This function allows for advanced geospatial visualization, ensuring that the data points are accurately represented on a map. The array of data points must have precise geographic coordinates (longitude, latitude) for Mapbox to work properly. The geo-referenced plotting of tracked objects in Mapbox can be seen in Figure 20 below, indicating that the transformation of each tracked objects classes is successfully completed (the recording can be seen in footnote 1) .

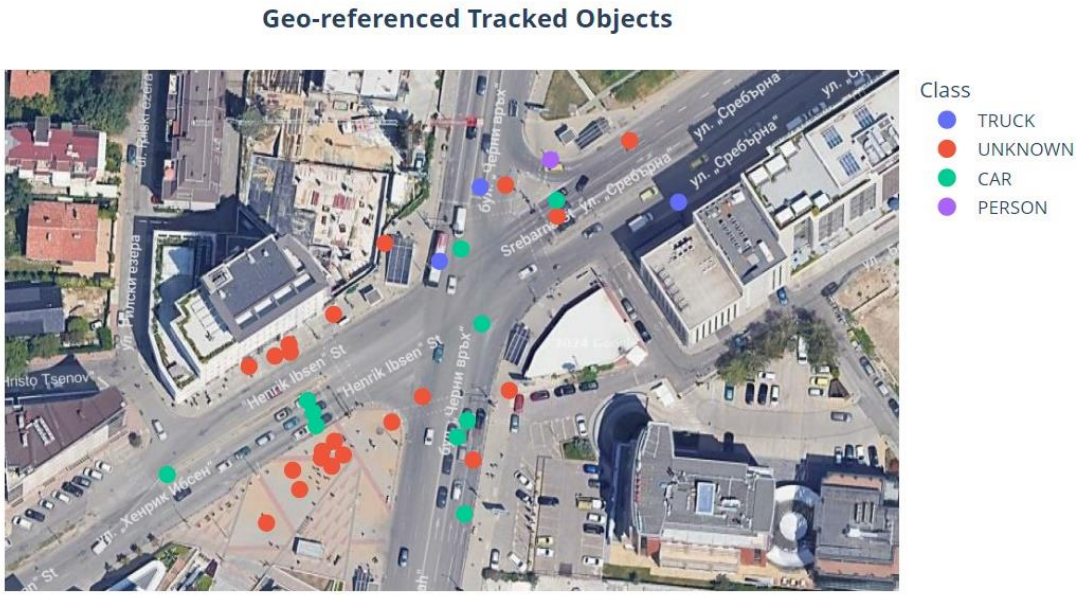


Figure 20. Tracked objects plotting of the geo-referenced data points[1]. Source: Author (2024).

4.3. Road segmentation

Road segmentation is created with lane and junction construction as its primary focus. The segment polygons are manually digitized and aligned with the real road structure as they exist in the satellite images (USGS Landsat 7 ETM+ C2 L1 with a resolution of 15-meter panchromatic blend and ESRI's world imagery service up to 1-meter resolution). The lanes' locations and positions in relation to the border determined their given names. Northern, northeastern, southern, and southwestern road segments are the four distinct road directions in the study area.

The northern road segment is divided into two lanes that travel outward and three lanes that travel inward toward the main intersection. The road segment in the northeast is made up of two lanes

[1] https://youtu.be/O5KluBLHygk

going outward and 3-4-3 lanes inward, which means the lane splits into four lanes at one point before returning to three lanes inward. There is a single turning lane connecting the northeastern road to the northern road's outbound lane, situated between the two routes, as shown in Figure 6.

Three inward and three outward lanes are present on the southern edges of the road. The southwestern road segment has 2-1 outward lanes (meaning that two lanes eventually merge into one lane) and 2-3 inward lanes (meaning that two lanes eventually become three lanes). Additionally, there is a single turning road segment that connects the southern outward lanes to the southern bidirectional outward lanes. In total, there are five junctions that constitute the road segmentation: four fork junctions and one main intersection junction. Fork junctions, for instance, facilitate division and merging in the case of a single turning route in the northeast that links two separate road segments. The semantic road segmentation details can be seen in Figure 21 and Table 7.

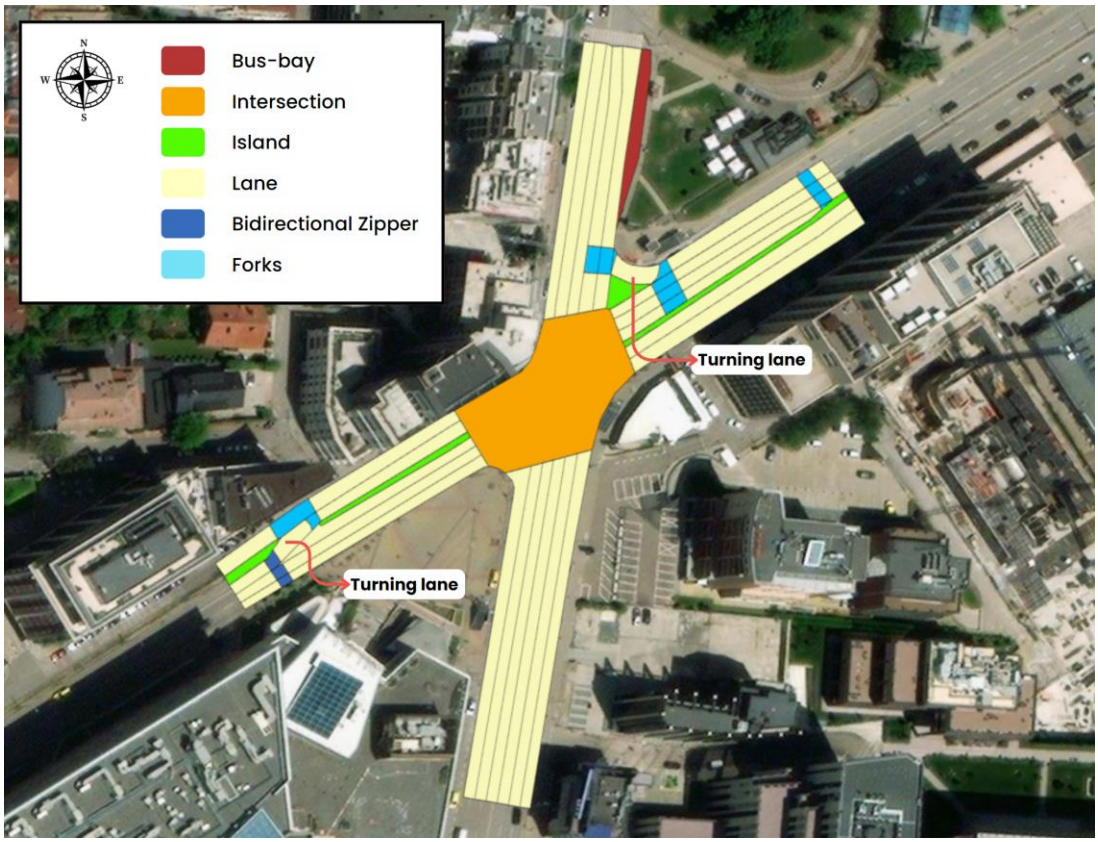


Figure 21. Road semantic information map. Source: Author (2024).

Table 7. Road segmentation details. Source: Author (2024).

Type	Segments	Count
Road	Inward North	3 lanes
	Inward North-East	4 lanes, 1 turning lane
	Inward South-West	3 lanes
	Inward South	3 lanes
	Outward North	3 lanes
	Outward North-East	2 lanes
	Outward South-West	2 lanes, 1 turning lane

	Outward South	3 lanes
Junctions	Main Junction	1
	Forks	4

In the road segmentation, the information about permissible connection based on the intersection condition is also included. This information shows the connection allowed from each lane to their target lane. Figure 22 displays the mapping of the allowed connections, indicated by the arrow in the lanes of each road segment. The strings visible in the main junction of the intersection represent the overall allowed connections between each lane. For example, consider Lane 1 of the road segment inward north, with the naming code "In_ne_1". In this lane, the permissible connections are either straight or left turn. Therefore, the allowed connection information for "In_ne_1" indicates that this lane connects to "Out_s_1" and "Out_sw_1". This information has been properly included in the intersection road semantic information and is used in the next stage of vehicle trajectory spatial analysis.

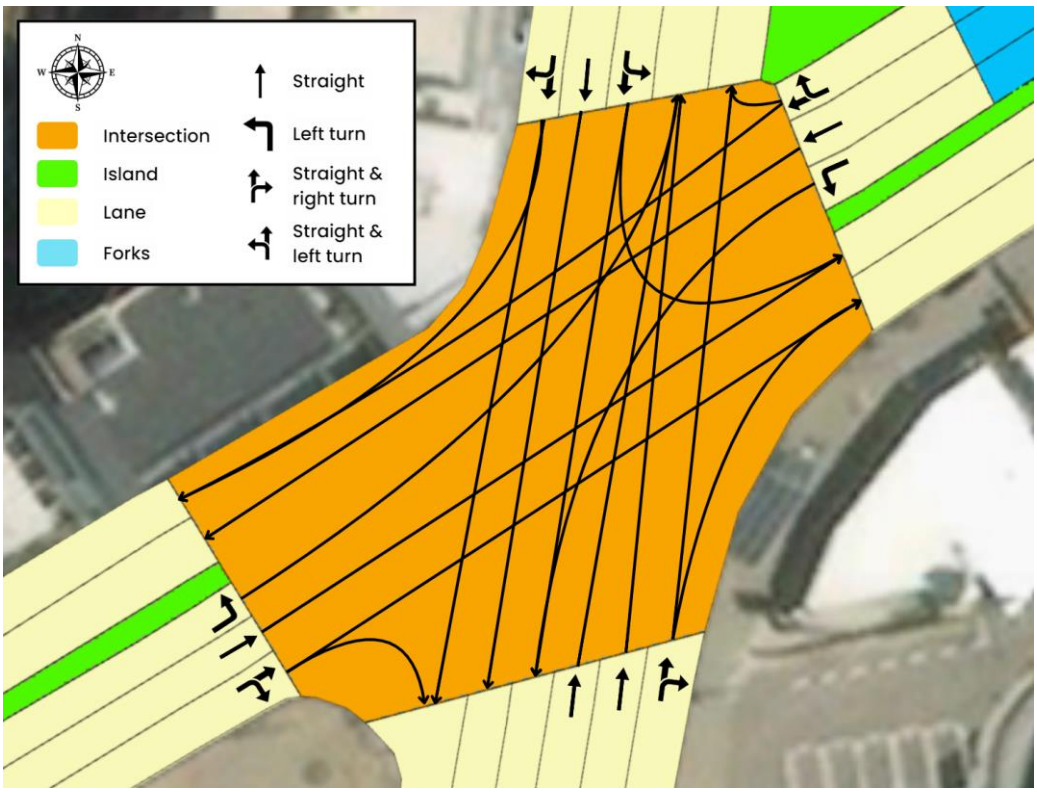


Figure 22. Allowed connection of each lane. Source: Author (2024).

4.4. Vehicle trajectory spatial analysis

Geospatial analysis is performed to determine object and trajectory categories using the results of CRS transformation and semantic road segmentation. The first step in the analysis is to identify the object type, whether the objects are possibly vehicles or non-vehicles, by intersecting the road segment with the geographical coordinates of each object. Here, the rational presumption is that an object is possibly a vehicle if its frame sequence begins and ends on a road segment. Otherwise, they are non-vehicles.

The term possible vehicle is sued because not all objects intersecting with road segment are exclusively vehicles, there is a possibility that a person class object ID might also be found within the road segments. On the other hand, any objects with a frame sequence outside of the road segment is considered as non-vehicle, specifically assumed as person class. Figure 23 clearly illustrates that object ID

categorized as possible vehicle include some that are classified as person (red line), and that non-vehicle objects shows a clear pattern of trajectory outside of the road segments, indicating a movement in the curb area of the road.

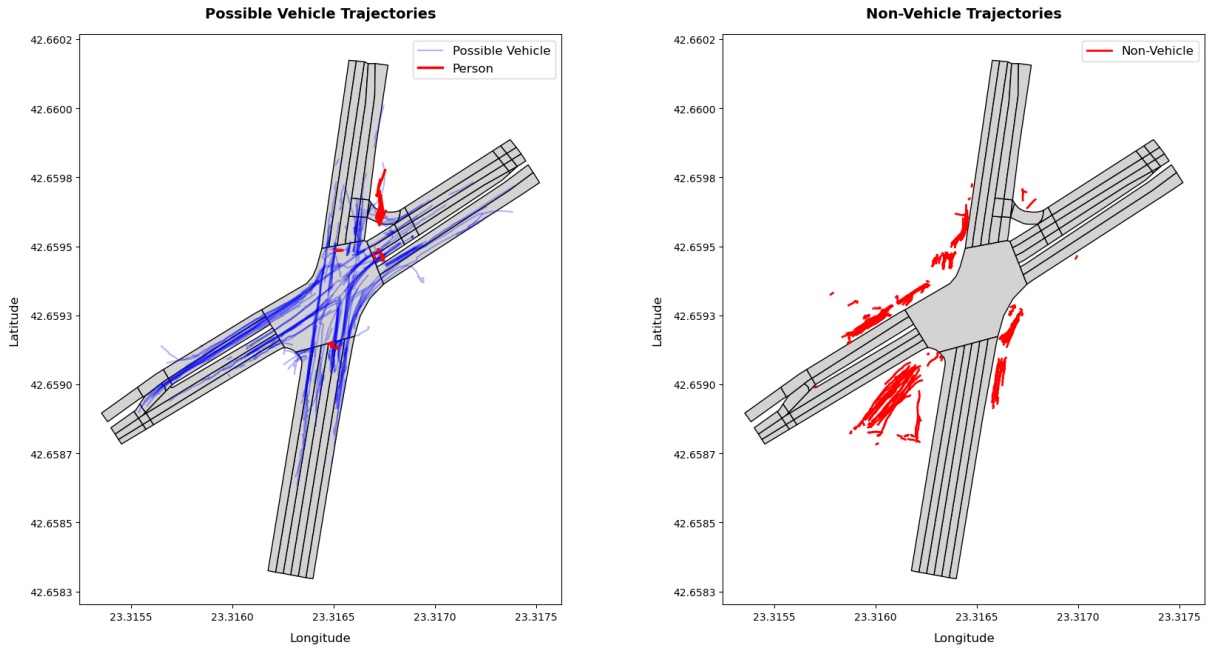


Figure 23. Object type classification plot. Source: Author (2024).

Figure 24 illustrates the distribution of object type categories. It shows that about 309 unique objects are classified as possible vehicles, although they are not exclusively vehicles since the category also contains the person and unknown objects. On the other hand, 194 objects which are later assumed to be pedestrians—are classified as non-vehicles. But there are anomalies as well. For example, four cars and four two-wheelers are classified as non-vehicles, which is probably the result of inconsistent labelling (when an object is associated with more than two classes).

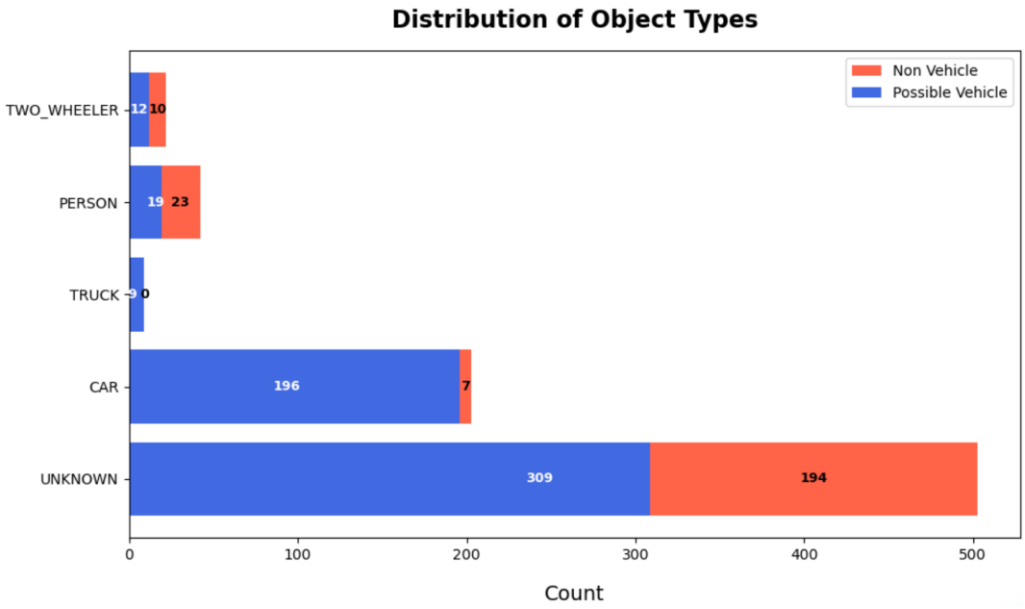


Figure 24. Object type classification distribution graph. Source: Author (2024).

The next step involves identifying the trajectories and analysing their patterns to understand movement behaviours or detect anomalies. According to the research goal of determining appropriate vehicle trajectories, a thorough categorization of trajectories is only done for possible vehicles; non-vehicles is not taken into consideration further. The methods section outlines the process for determining trajectories, and the spatial analysis result showed that 500 of the 516 recorded objects had incomplete trajectories. The analysis revealed that only 10 objects had complete trajectories, four cars had (short) complete trajectories, and two objects had violation trajectories due to illegal lane changes or U-turns. This is observed in the trajectory plot shown in Figure 25.

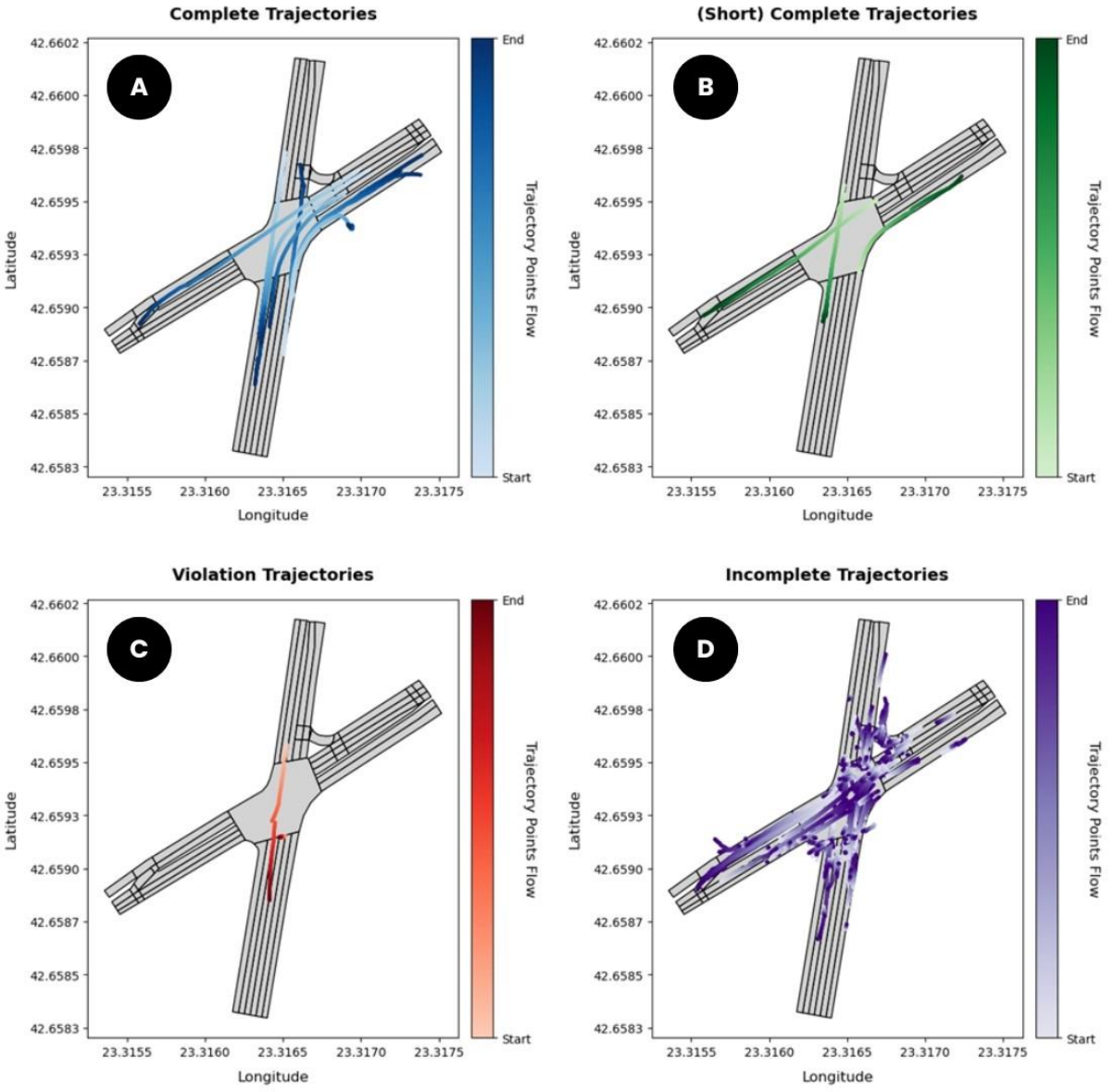


Figure 25. Trajectory type plot of each object ID; consist of complete (A), (short) complete (B), violation (C), and incomplete trajectories (D). Source: Author (2024).

The figure above comprises four trajectory plots, each categorizing different types of vehicle movements based on their completeness and legality. The color gradient in each plot indicates the flow of trajectory points, with lighter colors indicating the starting points and darker colors indicating the endpoints. Trajectory plot A displays complete trajectories, where objects have a full sequence of data points throughout the road segment. These paths are smooth and continuous, demonstrating that the

vehicles travel through the intersection without any missing data points. In contrast, Trajectory plot B illustrates (short) complete trajectories, which have starting or ending points that is close to the main intersection junction. These trajectories are nearly complete but are categorized as short because they begin or end abruptly very near the intersection main junction in proximity.

Trajectory plot C highlights violation trajectories, indicating illegal manoeuvres such as U-turns or other unauthorized turns. These paths deviate from the permissible routes and are clearly marked by their crossing over restricted segments. As for the last one, Trajectory plot D, it shows incomplete trajectories. In this plot D, objects have missing geo-location sequences in their data points. These trajectories typically start or end in the main intersection junction, this category shows a dominant ratio compared to the other categories. Information about the trajectory type distribution per frame, identified by their consistency, can be seen in Table 8 below.

Table 8. Trajectory type distribution per frame. Source: Author (2024).

Categories	**Trajectory Type**	**Classes**				
		car	truck	two-wheeler	person	unknown
Consistent	Complete	1	0	0	0	1
	(Short) Complete	0	0	0	0	0
	Violation	0	0	0	0	0
	Incomplete	8	3	0	1	250
Recognized	Complete	6	2	0	0	0
	(Short) Complete	4	0	0	0	0
	Violation	1	0	0	1	0
	Incomplete	179	2	18	39	0

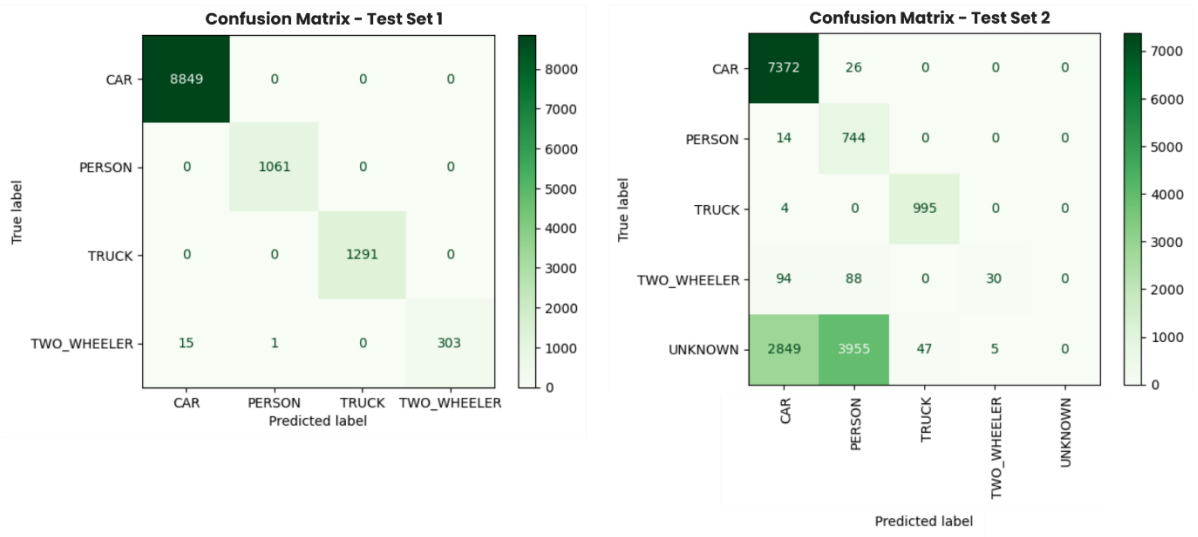
In the table above, 'Consistent' means that an object has a consistent class throughout, while 'Recognized' means that it has more than one class, including unknown, but is assumed as the other class. The object and trajectory types acquired in this phase improved the unknown object prediction in the training of the reclassification model, as shown in the following section.

4.5. Random forest unknown class reclassification

As indicated by the results of the data parsing analysis in the first sub-section of this chapter, the raw dataset's class classification is somewhat complex, which can add noise and reduce the RF model's ability to make accurate predictions. The osef dataset content overview shows three categories: recognized objects (one unknown among two classes, thus classified as the known class), consistent objects (with a single class throughout), and unidentified objects (changing classes more than twice, presumably producing noise in Random Forest training). The model is trained using two distinct .osef datasets, one from an afternoon session lasting four minutes and the other from a morning session lasting two minutes, comprising 108,156 frames. To reduce noise in the data, the training process did not directly use the class column as it is for the label. Instead, logic is applied to prioritize an identified class type over an unknown. Therefore, it allows the prediction to focus completely on unknown or unidentified objects with multi-classes.

The first training round is done with the default settings of reclassification model training. The trained model are then tested with the first test set (without unknown label) and second test set (with unknown label). The first test set shows the model performance towards unseen data without unknown label. In this first test set, a high accuracy of 0.998 is achieved. This indicates that the model is able to predict the known classes most of the times. For example, it reclassified cars back to car class, which affects the TP and TN rates, thereby resulting in high accuracy. On the other hand, in the second test set, which included the unknown class, the model accuracy dropped to 0.563. This decrease in accuracy is primarily due to the lower rate of TN, which occur when the object belonging to the unknown class is correctly identified as the other known classes (e.g., car, truck, person, or two-wheeler).

This low accuracy in the second test set does not necessarily indicate a bad model. As shown in Figure 26, the confusion matrix for the second test set aligns with the model purpose of reclassifying the unknown class into specific object classes. There are no instances of TP for the unknown class, which means the model did not predict any instances as belonging to the unknown class, this is supported by the rate of precision and recall of 0 for the unknown class.



Class	Precision	Recall	F1-Score	Support
CAR	1.0	1.00	1.00	8849
PERSON	1.0	1.00	1.00	1061
TRUCK	1.0	1.00	1.00	1291
TWO_WHEELER	1.0	0.95	0.97	319
accuracy	1.0	1.00	1.00	11520
macro avg	1.0	0.99	0.99	11520
weighted avg	1.0	1.00	1.00	11520

Class	Precision	Recall	F1-Score	Support
CAR	0.71	1.00	0.83	7398
PERSON	0.15	0.98	0.27	758
TRUCK	0.95	1.00	0.98	999
TWO_WHEELER	0.86	0.14	0.24	212
UNKNOWN	0.00	0.00	0.00	6856
accuracy	0.56	0.56	0.56	16223
macro avg	0.54	0.62	0.46	16223
weighted avg	0.40	0.56	0.45	16223

Figure 26. Evaluation metrics for initial RF model training. Source: Author (2024).

Since the accuracy for the first test set indicates close-to-perfect performance, which is unlikely. Therefore, further confirmation is done using the learning curve with a cross-validation of 10-fold. Figure 27 on the next page shows that training dataset size larger than 30,000 frames does not lead to a substantial improvement in the model; 30,000 frames is the ideal size. The training set line shows constant shape near 100% accuracy; normally, there should be a gradual improvement pattern from the starting point.

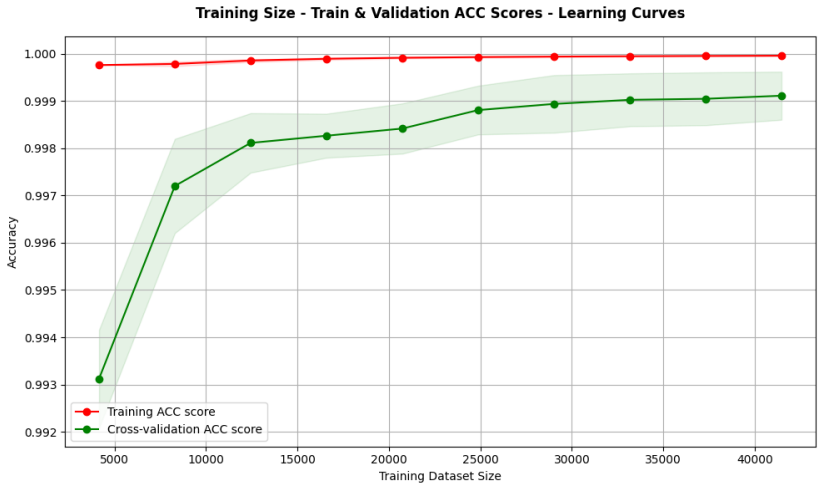


Figure 27. Random Forest ACC scores learning curves. Source: Author (2024).

Since the aim of this process is to reclassify the unknown labelled objects and generalize the classes as much as possible throughout the frames, the number of objects with multiple classes needs to be minimized. In this first attempt, there are still 88 objects that have multiple classes after predictions, and 11 of them consist of more than two classes.

Therefore, at the second attempt of the RF model training, hyperparameter tuning is conducted. In the first tuning attempt, only the n_estimators value is adjusted, setting it to 400. Additionally, the classes is substituted with 'Adjusted Class' labels to avoid multiple predictions. In this scheme, recognized objects take the known label, while unidentified objects take the majority class label. The accuracy of the second trained model increased slightly to 0.999, almost reaching 100%. As observed in Figure 28, at one point, the learning curve converges into one. This indicates a very ideal model. The multiclass prediction, in this attempt, is improving, as only 36 objects have multiple class predictions. Given that a near-perfect model is unlikely to be attainable, another iteration of hyperparameter tuning is conducted to improve the model's robustness and further reduce the multiple-class prediction.

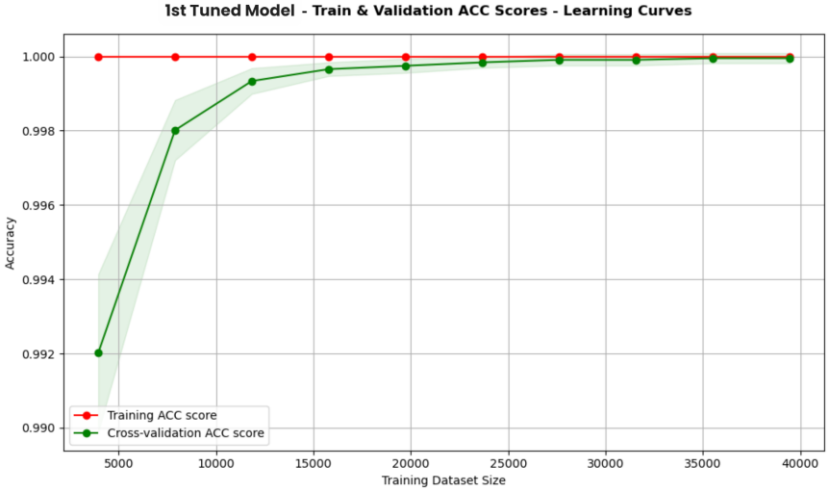


Figure 28. 1st Tuned model learning curves. Source: Author (2024).

The final iteration is performed using the parameter grid search method to obtain an optimal combination of hyperparameters. After several adjustments, the chosen hyperparameters are: 400 n_estimators, min_sample_split of 10, a max_depth of 9, and max_features set to 'none,' meaning that all

features are considered for each tree branch. Figure 29 below shows the learning curve of the second tuned model. It demonstrates a steady increase in accuracy, with the curves converging at the end, indicating good generalization.

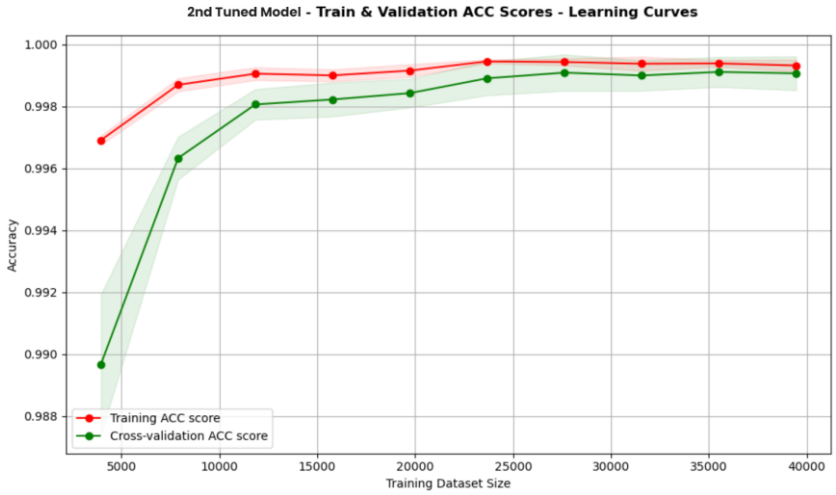


Figure 29. 2nd Tuned model learning curves. Source: Author (2024).

This pattern suggests that the model can predict new, unseen data more effectively and generalize better than the previous model. Although this model has a similar accuracy of 0.998 compared to the first model, its generalization is far better, reducing multiple class predictions for an object ID by 76%. Table 9 below shows the differences in each model's performance in minimizing multiple class predictions.

RF Model	Accuracy	Multiple Class Prediction	
		2 Classes	>2 Classes
Default model	0.998	88	11
1st Tuned model	0.999	36	4
2nd Tuned model	0.998	21	1

Table 9. Random forest model generalization performance, with accuracy based on the first test set. Source: Author (2024).

The 2nd tuned model is determined to be the ideal model for the purpose of this research. It is able to reduce the multiple class prediction more than the other trained mode. This 2nd tuned model shows that it only has one object ID that classified with multiple class, and in total there is 21 object ID overall with two classes. This final RF model directly incorporated into the final pipeline, where new, unseen data with an unknown label is reclassified. To handle the remaining multiple class object IDs, a logic to take the majority class per frame is implemented. Consequently, the entire dataset only have one true class for each object ID. The data overview of before and after prediction is displayed in Figure 30. After reclassification where all the object ID have 1 true class, the dataset is then uploaded to the database for the next processing phase.

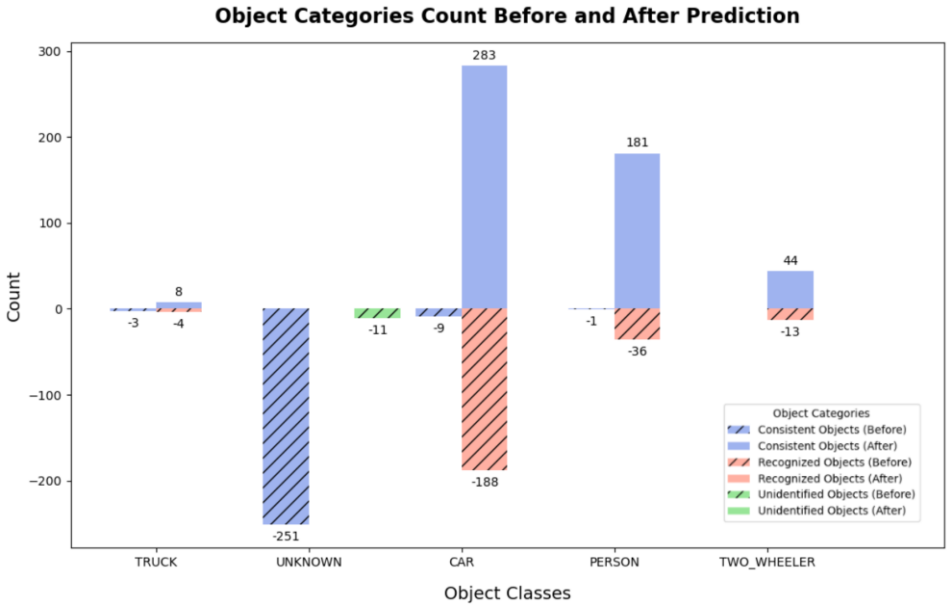


Figure 30. Object categories overview from reclassified dataset. Source: Author (2024).

4.6. Database overview

The database storing process is successfully completed, ensuring comprehensive and accurate data integration. The map-matching process to .gpkg files resulted in detailed information about the segments where the array of data points is located, effectively addressing all incomplete trajectory types of object ID. By comparing the incomplete trajectories with the missing sequences, this map-matching algorithm finds the edges and lanes that are closest to the end or starting point, as well as considering the permissible connections of the lanes. Additionally, the connection with the OpenWeatherMap API functioned seamlessly, allowing for the collection of weather data based on existing timestamps and geometry points. This integration enabled the retrieval of weather conditions, including temperature and humidity, providing a richer dataset for subsequent analysis and modeling. The detailed structure of the final PostGIS database can be seen in Table 10.

Table 10. Database structure overview. Source: Author (2024).

Attribute Name	Description	Data Type
Frame_number	Frame number of the recorded data	bigint
Date	Date of the recorded data	text
Timestamp	Timestamp of the recorded data	text
Object_id	Unique identifier for each object	bigint
Class	Classification of the object	text
Class_ID	Numerical ID for object class	integer
Speed_ms	Speed of the object in meters per second	double precision
Volume_m3	Volume of the object in cubic meters	real
Longitude	Longitude coordinate of the object	double precision
Latitude	Latitude coordinate of the object	double precision
Altitude	Altitude coordinate of the object	real
Speed_change	Change in speed over time	double precision
Volume_change	Change in volume over time	real

Average_speed_ms	Average speed of the object	double precision
Average_volume_m3	Average volume of the object	real
Reclassified_class	Updated classification of the object	text
Acceleration	Acceleration of the object	double precision
Deceleration	Deceleration of the object	double precision
Max_Acceleration/class	Maximum acceleration within the object class	double precision
Min_Deceleration/class	Minimum deceleration within the object class	double precision
Trajectory_type	Classification of the object's trajectory	text
Object_type	Classification of the object from spatial analysis	text
Segments	Road segments where the data point falls	text
Weather_conditions	Specific weather conditions based on timestamp	text
Temperature	Temperature at the time of recording	double precision
Precipitation	Precipitation at the time of recording	double precision
Geom	Geometric information (spatial data)	geometry (Point)

The PostGIS database contains a detailed set of fields for tracked objects, enabling precise geospatial analysis and advanced traffic simulations. Key temporal context is provided by Frame_number, Date, and Timestamp. Object identification and classification are handled by Object_id, Class, and Class_ID. Dynamic and spatial characteristics, such as Speed_ms, Volume_m3, Longitude, Latitude, and Altitude, ensure detailed representation.

On the other hand, Speed_change and Volume_change track dynamic behavior. Average_speed_ms, and Average_volume_m3 offer averaged metrics and Reclassified_class shows the final class of the objects that has been reclassified using the RF model during the storing process. Acceleration, Deceleration, Max_Acceleration/class, and Min_Deceleration/class measure traffic dynamics. Trajectory_type and Object_type provide classifications from the previous vehicle spatial analyses, and Segments indicates the road segment for each data point. Weather conditions, including Weather_conditions, Temperature, and Precipitation, are integrated from the OpenWeatherAPI. The Geom column holds geometric information using geographic coordinates.

This database structure further supports comprehensive geospatial and temporal analysis, allowing for easy data fetching for the integration in SUMO simulation, including scenario analysis. This database has been enriched and serves as a valuable resource for both research and operational purposes.

4.7. Traffic simulation

Before running the traffic simulation, the initial configuration of SUMO is created. This includes the adjustment of road network XML file based on the information fetched from OSM. OSM data fetching includes information about the road network in the intersection as well as the building shapes of the surrounding area. The raw road network file obtained from the OSM did not fully reflect the real-world conditions of the intersection; therefore, adjustments is made to align the shape of the SUMO road network with the road segments created in the road segmentation stage. The initial configuration of the road network is illustrated in Figure 31 (next page). For other initial configuration aside from the road networks, such as permissible connections and TLS settings, please refer to Annex 3: Permissible connection configuration and Annex 4: TLS system configuration.

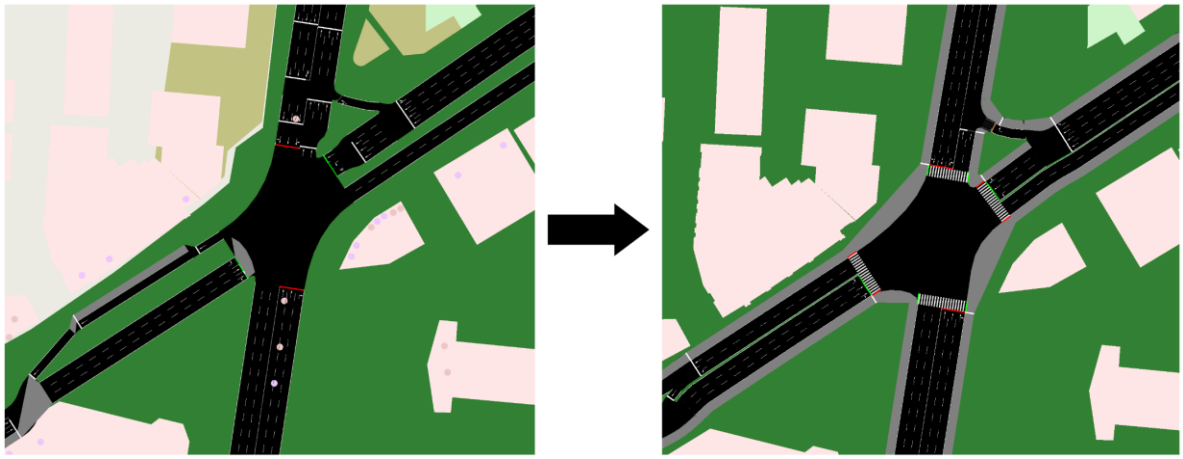


Figure 31. (Left) Raw OSM road network file and (Right) Adjusted road network file. Source: Author (2024).

Utilizing the initial configuration as the base, the traffic simulation are then run. To simulate traffic in real-time, the traffic simulation program SUMO fetches the parsed dataset from the database. As the methodology section explains, there are two types of simulation experiments: XML based and dynamic using TraCI. XML dataset is then generated considering the map-match logic. There are five XML files produced in the workflow: *"passenger (CAR).trips.xml"*, *"truck(TRUCK).trips.xml"*, "*motorcycle(TWO-WHEELER).trips.xml*", and *"pedestrian(PERSON).trips.xml"*. In contrast to the other XML formats, the pedestrian XML format only assigned to the pedestrian lane, denoted by lane 0 in the road network. Sumo-gui—a visualization tool for SUMO—then runs the simulation and opens the configuration automatically. Successful execution of the XML-based approach simulation results in an accurate representation of the tracked objects from the .osef data, as seen in Figure 32 (XML-based simulation recording can be seen in footnote 2).

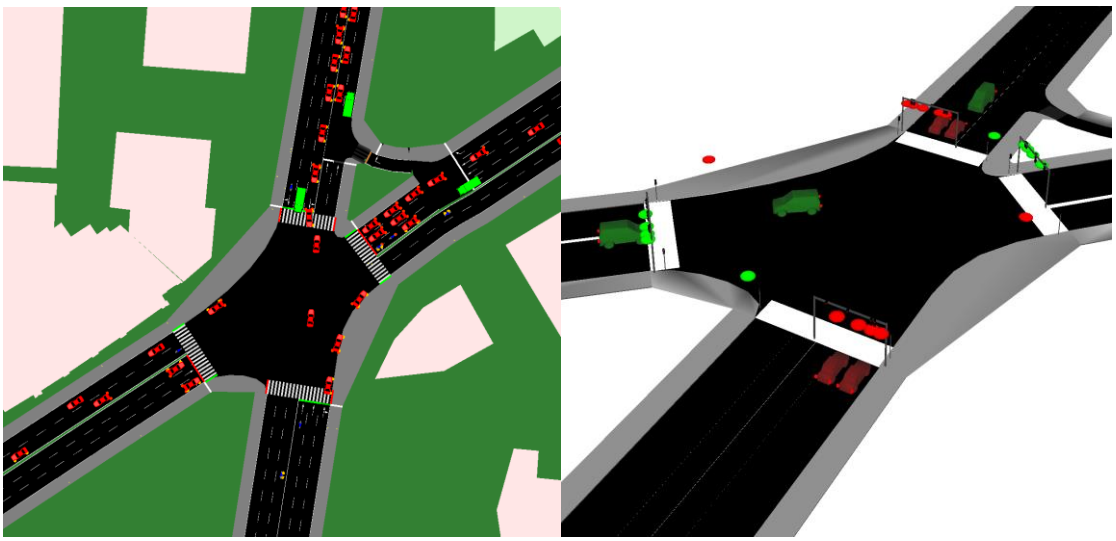


Figure 32. (Left) XML-based approach[2] and (Right) Dynamic approach with TraCI[3] for SUMO traffic simulation; red colour represent car, green represent truck, blue represent two-wheeler, and yellow represent person. Source: Author (2024).

[2] https://youtu.be/98qG7eiS9kY
[3] https://youtu.be/D7faWRh8lsU

On the other hand, in a dynamic approach with TraCI (recording can be seen in footnote 3), the database is fetched directly without the need to store data locally. This workflow employs TraCI as middleware. TraCI adds and removes vehicle IDs when they arrive at their destinations, updating them in real-time based on the timestep. The dynamic changes with TraCI (Figure 32 right image) have yet to be able to designate lanes like the XML-based approach does. Route distribution is necessary to identify certain lanes; nevertheless, considering the size of the dataset and the range of route distributions, this is a challenging process. However, the dynamic flow functions as intended, automatically determining the optimal lane for a vehicle.

4.8. What-if scenario testing

This section presents the results of the generated what-if scenarios, assessing how well the dataset performs under different settings during the simulations. The scenarios include a standard roundabout junction, an 8-shaped roundabout, and a two-level intersection with an underpass. Each scenario tests the adaptability and robustness of the dataset in varying traffic conditions and junction configurations.

For the transition from the original intersection to a standard roundabout, the dataset adapts seamlessly due to the similarity in the number of road segments to the original configuration, on that note it seems to adapt well due to the extra connections that has been made possible by roundabout structure that is not apparent in the original junction of the intersection. The roundabout's continuous flow and reduced stop-and-go conditions show improved traffic performance, indicating significant reductions in delays. Figure 33 below shows the simulation for this scenario (recording of the standard roundabout what-if scenario can be seen in footnote 4).

Figure 33. Roundabout scenario[4]. Source: Author (2024).

However, the 8-shaped roundabout scenario introduces complexity due to the removal of several lanes. Specifically, the north-east turning lane, one lane from the inward road in the north-east, and one lane each from the south road edges in both directions are removed. This reduction in lanes causes some

[4] https://youtu.be/IEIhqvyyJYs

vehicles, particularly those stopping at the northeast turning lane, to experience loading issues, which results in the simulation failing to run. This error can be seen in Figure 34.

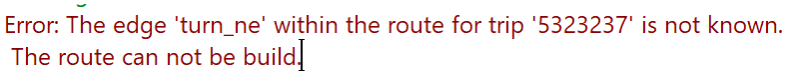


Figure 34. SUMO failed simulation attempt due to missing edges. Source: Author (2024).

The algorithm struggles with these adjustments since the ratio of vehicles with connections through the road segment "turn_ne" is insignificant. For the purposes of this 8-shaped what-if scenario simulation, they are manually edited out. Snapshot of the double roundabout can be seen in Figure 35 (recording of this scenario can be found in footnote 5).

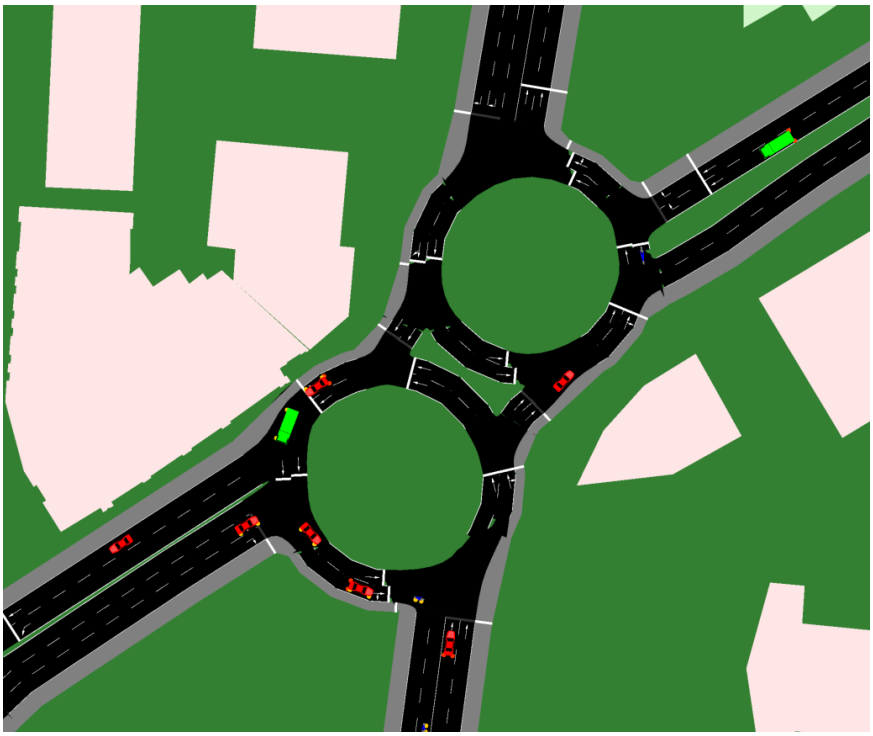


Figure 35. 8-shaped roundabout scenario[5]. Source: Author (2024).

Lastly, the two-level intersection scenario simulation, involving an underpass for east-west traffic, demonstrates the traffic simulations capability to manage multi-level traffic flows. This scenario, inspired by the Save Sofia proposal, integrates an underpass to maintain uninterrupted east-west traffic while allowing surface-level north-south traffic. The model successfully incorporates these changes, showcasing its flexibility in handling varied traffic structures. Due to the missing interchanges connections that originally exist in the intersections, the real-time data of .osef cannot be loaded completely because of the connection unavailability in certain roads, for example connection from north to east are lost due to this restructuring.

The road network for this scenario is illustrated in Figure 36. As shown, to adapt with the scenario concept, the elevation of the east-west road segment has been altered to be lower than actual ground elevation around 6-7m. However, in the 2D view, it is hard to discern the elevation difference that indicate the existence of underpass. This makes monitoring the traffic challenging due to the lack of visibility in the interchange spot. It can be seen that there are green trucks and red car moving through the

[5] https://youtu.be/gZhngg7oYJw

intersection, which indicate that the vehicle navigates well (recording of this two-level intersection what-if scenario can be found in footnote 6).

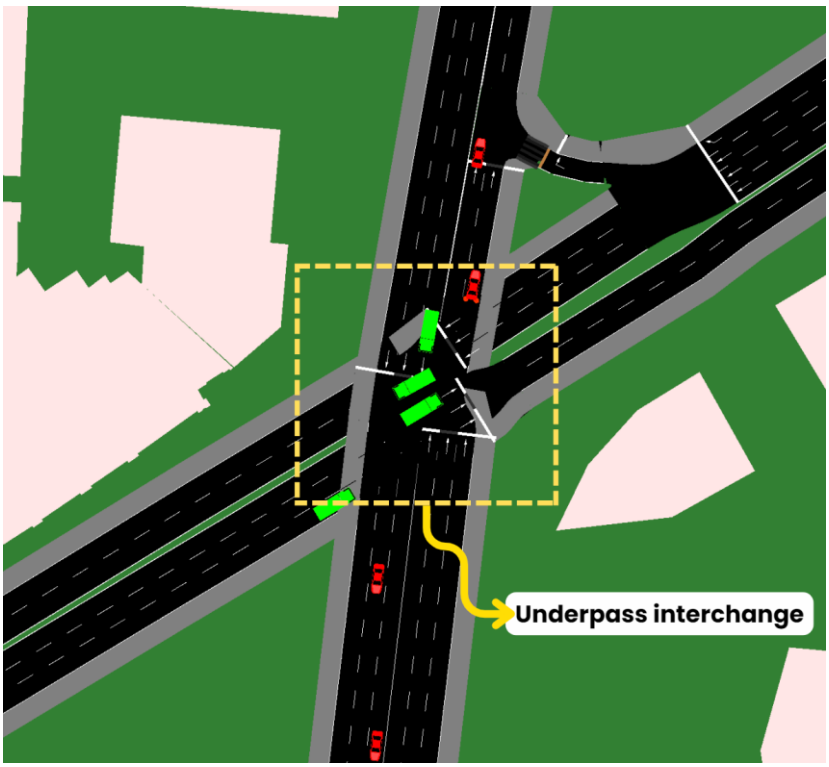


Figure 36. 2D View of the two-level intersection scenario[6]. Source: Author (2024).

For this scenario, a 3D view provides a much clearer representation of the traffic flow and the impact of the elevation changes. The lowered road segment for the underpass are more easily visible, allowing for a better assessment of how vehicles interact with the adjusted road structure. The illustration can be seen in Figure 37. The figure shows the different angle of the 3D view (recording can be found in footnote 7), highlighting the elevation difference and the movement of the vehicle in each level of the road (underpass and ground level road segments). The 3D view does not provide a perfect realism due to the limited configuration available in SUMO, however it is enough to indicate the different elevation in this two-level intersection scenario.

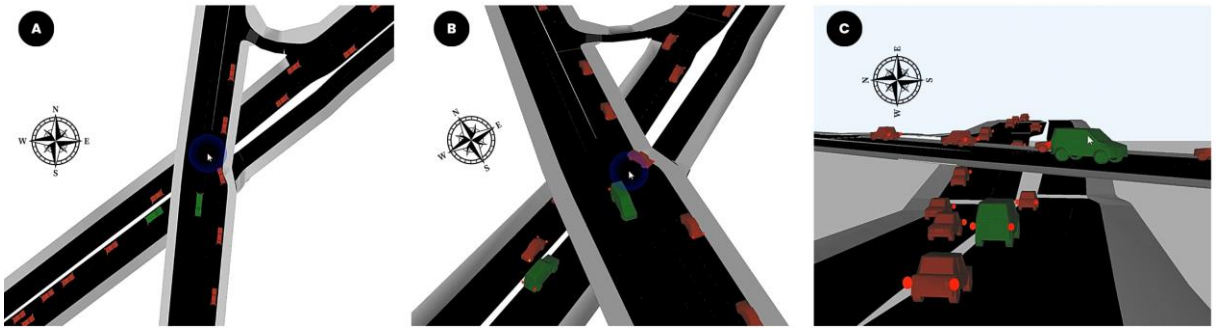


Figure 37. 3D view of Two-level intersection scenario[7]; (A) shows northern angle, (B) shows north-eastern angle, and (C) shows eastern angle. Source: Author (2024).

[6] https://youtu.be/MU_4o8fcCwU
[7] https://youtu.be/6LWDm4XxZkc

To further analyse which scenarios performs better in handling traffic, comparison are done for the traffic flow over time across different junction shapes. 10-minutes span available data is used to run the simulation in this case. Traffic flow over time, defined as the rate at which vehicles pass a point on the road, is an important indicator of a junction's efficiency. Higher traffic flow indicates smoother and more efficient movement, while lower traffic flow suggests congestion and delays. This comparison is made between the original junction shape, the standard roundabout, and the 8-shaped roundabout scenarios; each includes the TLS scenarios as well, in total there are 6 different scenarios being compared. The two-level intersection scenario is not included in this process, due to the missing connections between east-west and north-south road segments. This absence of the main junction prevents the proper loading of all traffic data, making it significantly different from the other scenarios. The results of this comparison are illustrated in the Figure 38 below, providing a clear visual representation of how each scenario impacts traffic flow.

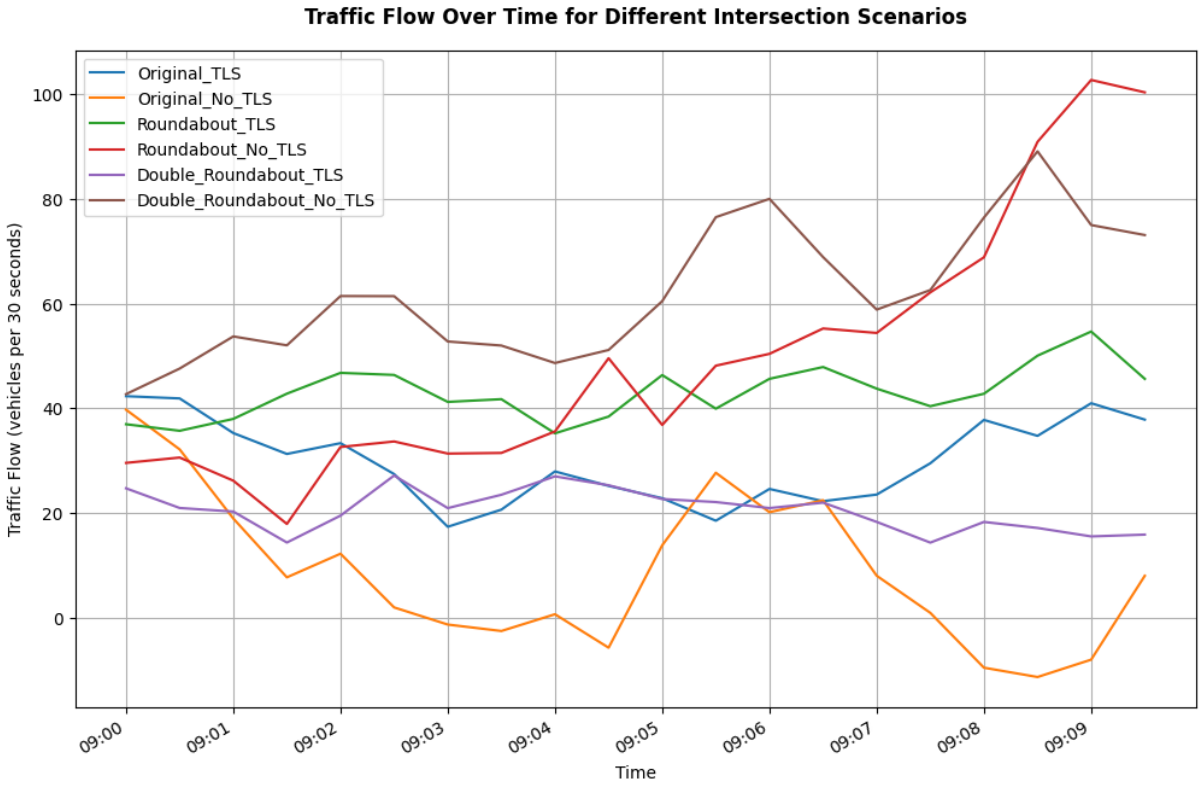


Figure 38. Traffic flow comparison of the scenarios. Source: Author (2024).

In the chart above, the traffic flow are measured with frequencies of 30 seconds, across six different intersection configurations over a 10-minute period. It is evident from the chart, that the 8-shaped or double roundabout with no TLS consistently shows the highest traffic flow, with a noticeable increase towards the end of the observed period. On the other hand, the original road network with no TLS configuration shows the lowest traffic flow, with a significant decline after the initial minutes, suggesting that at some point traffic congestions often happen in this scenario. The roundabout scenario with no TLS maintain a steady traffic flow, which slightly lower than 8-shaped roundabout at first but gradually became higher than the other scenarios by the end. In contrast, the scenarios involving TLS shows a moderate traffic flow, with distinct decrease in the roundabout and double roundabout road network compared to their results without TLS involved.

4.9. DT framework assessment

In this section, the validation results is shown. Starting from the accuracy of the simulation data to the real-world traffic observations. This observation is done on 2024-05-30, morning time 9.00 – 9.50 AM as explained in the methodology section. The result of the observation of 10 minutes each road edges can be seen in Table 11 below,

Table 11. Traffic observation data. Source: Author (2024).

Vehicle class	09:00 - 09:10		09:10 - 09:20		09:20 - 09:30		09:30 - 09:40	
	In_sw	Out_sw	In_ne	Out_ne	In_s	Out_s	In_n	Out_n
Cars	136	67	128	201	237	176	157	152
Truck	11	2	13	16	19	18	12	11
Two-wheeler	0	0	0	3	2	3	3	2
Total	147	69	141	220	258	197	172	165

This data is then used to compare it with the traffic count from the DT traffic simulation utilizing SUMO induction loop to count it. The result from SUMO is matched based on the temporal factors of the observation. The comparison are plotted as follows in Figure 39.

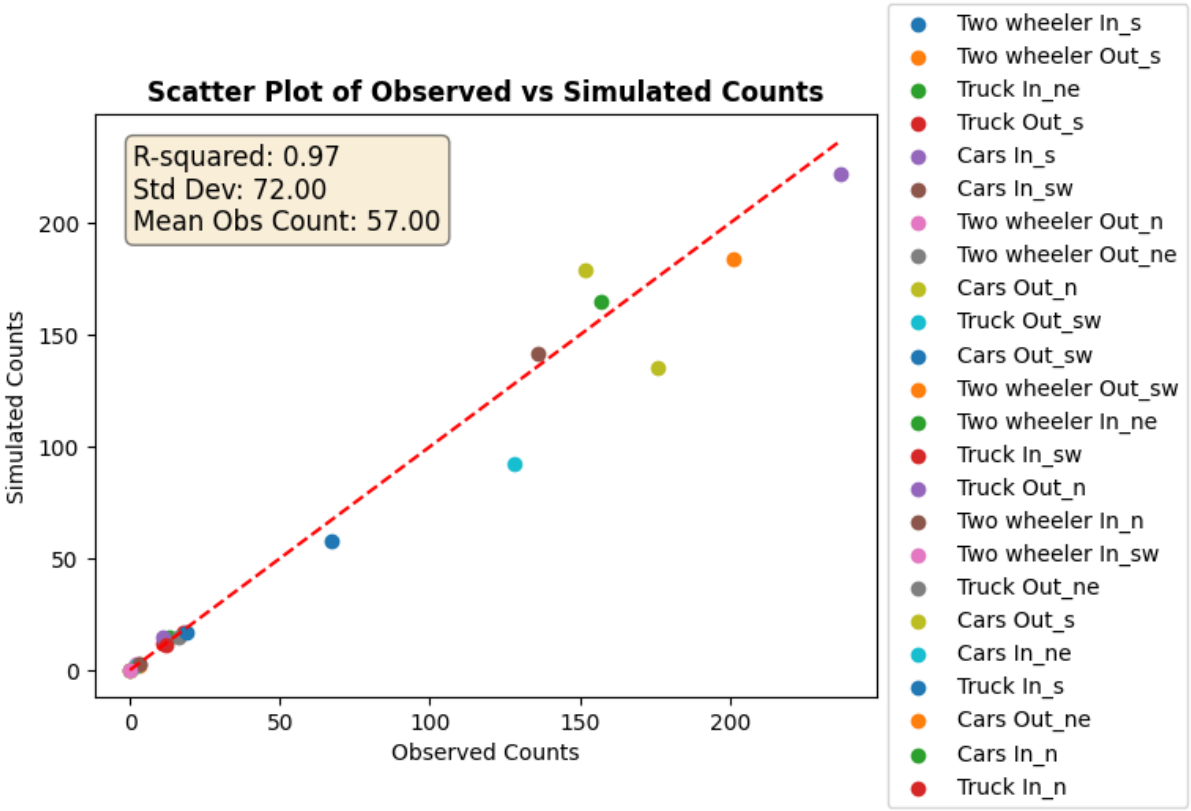


Figure 39. Scatter plot of observed traffic compared to simulated counts. Source: Author (2024).

With an overall RMSE of 11.34 and a mean observed count of 57.00, the relative error stands at approximately 19.89%. This indicates that, on average, the simulated data deviates from the observed data by about 11 to 12 units. The RMSE is calculated by comparing the predicted traffic counts from the simulation model against the actual observed counts. The R-squared value of 0.97 suggests that 97% of the variability in the observed traffic counts is explained by the simulated counts, demonstrating a very high degree of correlation. This high R-squared value indicates that the simulation model effectively

captures the overall patterns and trends observed in the real traffic data. The observed and simulated counts are compared for different vehicle classes, such as cars, trucks, and two-wheelers, as well as for different traffic directions, ensuring a robust evaluation of the model's performance.

The standard deviation of the simulated counts, at 72.00, is relatively high. However, this must be considered within the context of the specific traffic segments and their observed counts. The high standard deviation reflects the natural variability in traffic flow across different segments and times, which is typical in real-world traffic scenarios. Factors such as traffic congestion, road conditions, and time of day contribute to this variability and impact both the observed and simulated results. For each vehicle type and traffic direction, individual RMSE values is calculated to assess the accuracy of the simulated counts. For example, the RMSE for "Cars In_sw" is 6, indicating that the simulated counts for this category deviate from the observed counts by an average of 6 units. Similarly, the RMSE for "Cars Out_n" is 19, while for "Two-wheeler In_s," it is only 0.33, showing very little deviation.

The second evaluation method used in this section is cosine similarity (CoSim). This analysis measures the angle between two vectors representing trajectories, focusing on their directional similarity in a multi-dimensional space. The comparison is conducted between the real-world vehicle trajectories and the simulated trajectories. Only vehicles identified as having complete trajectories during the vehicle spatial analysis stage is included in this analysis. For the data sample of two minutes morning .osef data, there are 10 object ID that categorized as having complete trajectories. The simulated trajectories of these objects is extracted using SUMO in an XML format, which includes the geo-coordinates sequence of the trajectories of each time step. For the plot of these chosen objects, including both real-world trajectories and simulated trajectories, please refer to Annex 5: Real-world trajectory plot and Annex 6: Simulated trajectory plot. The result of the cosine similarity calculations can be found in Table 12.

Table 12. Cosine similarity between real-world and simulated trajectories. Source: Author (2024).

Object ID	Cosine Similarity	Euclidean Distance
5323299	0.9999999999530894	0.002455
5323769	0.9876543210987654	0.002934
5324553	0.9206119483729465	0.001728
5324711	0.9999999999622543	0.002548
5324579	0.8995432671840923	0.002394
5324857	0.9999999999756457	0.002017
5324767	0.9999999999876543	0.001980
5324878	0.9999999999699384	0.002045
5324944	0.9999999999843756	0.001956
5324926	0.9999999999827543	0.001923

It is important to note that cosine similarity measures the angle between trajectory vectors, indicating how similar their directions are, rather than comparing the exact shapes or positions of the trajectories. Therefore, a high cosine similarity indicates that the overall movement direction of the trajectories is similar, even if the exact paths differ slightly. Based on the Table 12, the CoSim values for most objects are very close to 1, suggesting that the simulated trajectories generally follow the same directional patterns as the real-world trajectories from .osef dataset. For example, object IDs 5323299, 5324711, 5324767, and 5324857 have cosine similarity close to 1, indicating near perfect alignment in their directional movement. Figure 40 on the next page shows the plot of object IDs 5324587 and 5323299 trajectories, it can be seen the form of their trajectories for both real-world and simulated are fairly similar, aligning with the high score of the CoSim.

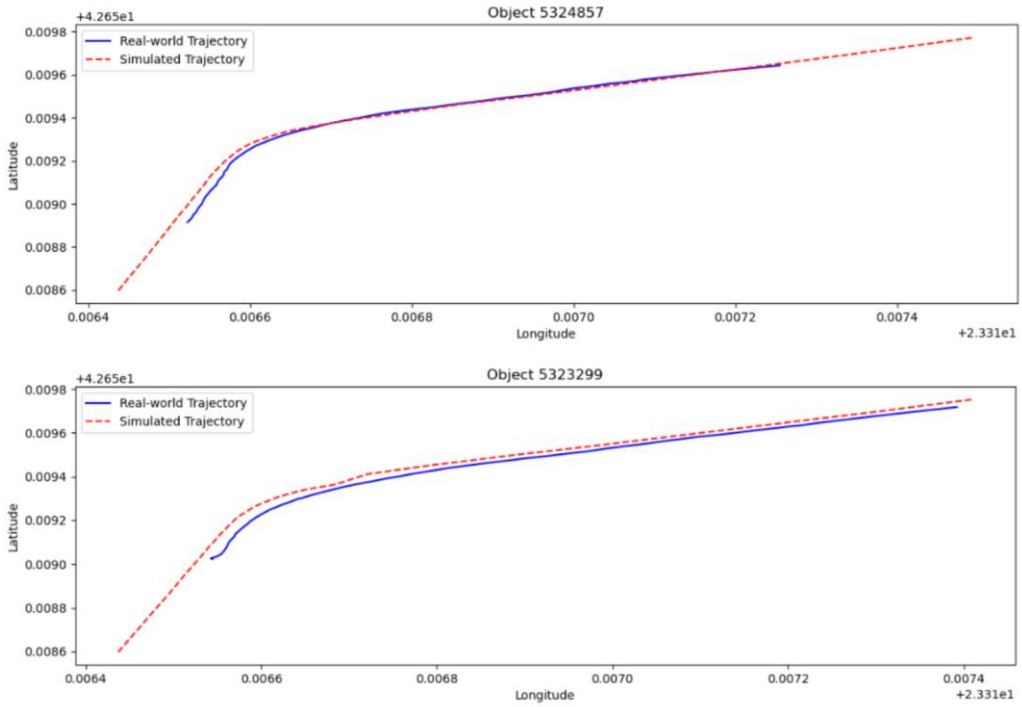


Figure 40. Trajectories comparison, real-world and simulated, object IDs; 5324587 and 5323299. Source: Author (2024).

For some objects, such as object IDs 5323769 and 5324553, the CoSim values are slightly lower than the rest of object IDs, around 0.92 to 0.98, however this is still considered high similarity for CoSim values. This slight difference indicates minor deviations in the directional patterns while it still represents a near perfect similarities between the real-world and simulated trajectories. To ensure comprehensive assessment, Euclidean distance help support the assessment by looking at the positional deviation. The Euclidean distance for object ID 5323769 for example, is around 0.0029, showing a slight positional deviation. This supports the observation that while the overall direction indicated by CoSim values are high, there are minor differences in the exact paths taken by the vehicle. Figure 41 highlights another example of this case, with specific example of object ID 532479. This object ID clearly has a visual discrepancy in trajectory shapes, which explains the lower CoSim values of 0.899. The Euclidean distance of this vehicle is 0.00239, further supports that there is a positional deviation in this object trajectory.

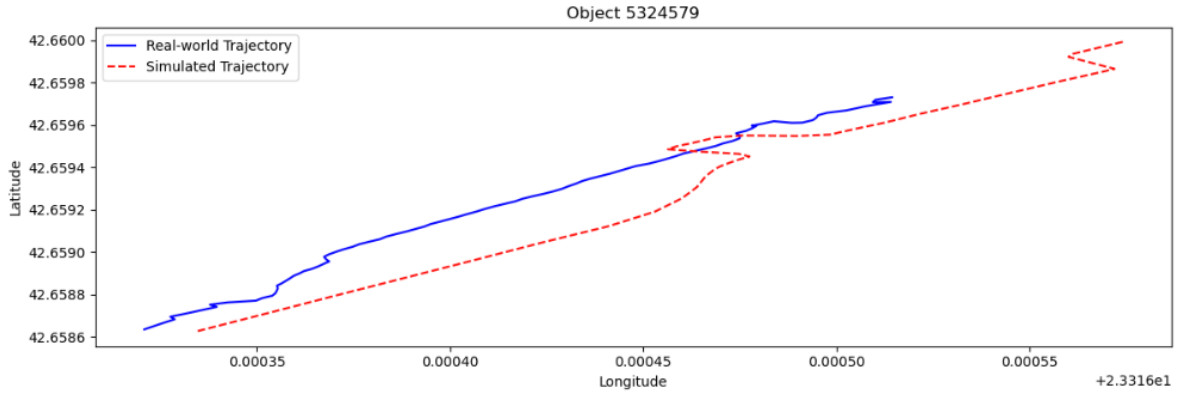


Figure 41. Trajectories plot for Object ID 532479. Source: Author (2024).

4.10. Summary

This results chapter presents the outcomes of data analysis, transformations, and traffic simulations. Initial data analysis of the .osef dataset revealed 58,939 data points across 516 tracked objects, categorized into consistent, recognized, and unidentified groups. Successful transformation of local coordinates into geo-coordinates enabled accurate mapping. Road segmentation accurately mapped lanes and junctions, crucial for spatial trajectory analysis. Most objects had incomplete trajectories, with only a few exhibiting complete paths. The Random Forest model reclassified unknown objects, significantly improving classification accuracy and reducing multiple-class predictions. The enriched PostGIS database supports advanced traffic simulations with comprehensive geospatial and temporal data.

Traffic simulations were conducted using SUMO with adjusted configurations to reflect real-world conditions. Various what-if scenarios, including standard and 8-shaped roundabouts and a two-level intersection, were tested. The standard roundabout without traffic lights provided the best traffic flow, while the 8-shaped roundabout posed challenges due to lane reductions. Validation against real-world observations showed high accuracy, with an R-squared value of 0.97. Cosine similarity analysis demonstrated that simulated trajectories closely followed real-world directional patterns. These results confirm the effectiveness of the digital twin framework and the integration of real-time LiDAR data into traffic simulations, offering valuable insights for traffic management and urban planning.

5. DISCUSSION

This chapter discusses the key findings from the results chapter, including the implications and limitations inherent in this research. The discussion of the findings follows the order of the analysis in the results chapter. Additionally, the chapter concludes with an elaborate discussion on possible future works, highlighting potential areas for further research and development.

5.1. Findings

This section discusses the findings from the results chapter, divided into subsections following the order of the analysis to ensure a logical flow from initial data parsing to the overall assessment of the DT framework.

5.1.1. Finding 1: OSEF data overview and parsing

The initial data parsing process of the .osef datasets on page 41 provides several significant insights into the classification and distribution of tracked objects in the study area. The dataset identification revealed a predominant number of objects with a single class (consistent objects) and objects that have two classes, with one of them being an unknown class (recognized objects). The consistent objects group is found to be the largest of the three group categories, as illustrated in Figure 17 on page 41. This suggests that the majority of the tracked objects in this .osef data captured by the LiDAR sensors, maintained a stable classification throughout the frames, thus highlighting a degree of performance in the data collection process. However, looking at the array of data points, the dominance of the unknown class within the consistent objects raises concerns about the accuracy of the classification system. This indicates that a substantial number of objects not properly classified into precise classes such as cars, trucks, or two-wheelers, implying potential limitations in the OSEF classification algorithm.

Looking at the recognized objects, which it is characterized as an unknown class at times but also assigned to known classes (cars, trucks, two-wheelers, or persons), highlights that the classification algorithm could identify these objects. However, there are instances of uncertainty or overlap with the unknown category at some points, which could stem from varying environmental conditions, object occlusions, or inherent limitations in the classification algorithm of .osef dataset. In the .osef datasets, there is a small proportion of unidentified objects that have more than two classes. This inconsistency shows that certain objects possess characteristics that make them difficult to discern and classify consistently by the algorithm; this might be due to their dynamic nature or the initial classification framework of the sensors.

5.1.2. Finding 2: Local coordinate transformation to geo-coordinates

Another issue with the .osef dataset is that its coordinate system, being cartesian coordinate systems, hinders the integration of the dataset into the DT framework. Therefore, the coordinate transformation from local to geo-coordinates is performed. The process of transforming local Cartesian coordinates into WGS84 EPSG:4326 geo-coordinates for the initial dataset provides insights into the importance of geospatial data representation. By having accurate geo-coordinates, it is possible to proceed further with the research. Accurate geospatial representations of the .osef dataset are deemed crucial for the traffic simulation. This ensures that the DT maintains a high level of fidelity and realism in comparison to its real-world counterpart. By mapping the dataset to accurate geospatial coordinates, the simulation can more precisely replicate real-world traffic patterns, leading to more valid and useful results.

The transformation is first validated using static objects, ensuring the accuracy of the coordinates transformation before applying it to the dynamic tracked objects. The successful alignment of the geo-referenced data points of the static object with actual geographical features, such as curbs and buildings, shows the effectiveness of the transformation methodology. The plotting in Figure 19 on page 43 demonstrates the precise alignment well, indicating that the process of preserving the spatial relationships and positions of objects within the dataset is accurate. Additionally, studies by Felus & Burtch (2008) focusing on coordinate transformations for converting GPS data to local coordinate systems highlight the possibility and importance of accurate transformations for maintaining spatial relationships within datasets. The above study does a reverse process compared to this research, showcasing similar findings of the successful attempt in transforming the GPS dataset coordinates into the desired coordinates.

5.1.3. Finding 3: Road segmentation of the intersection

The road segmentation results on page 44, focusing on road segments, lane, and junction construction, provided a detailed representation of the road network following the real-world conditions of the study area. The detailed digitization of the road structure ensured that the segments were well-aligned. The segmentation captured the complex layout of the northern, northeastern, southern, and southwestern road segments, including the specific details of the lane configurations. Utilizing the satellite imagery and manually generating the polygon following the road structure has proven effective for this research. The inclusion of various semantic information in the attributes of the geopackage further enhances the analysis and process of establishing the traffic simulation DT framework. Research by Xing & Zhu (2021) highlighted a similar process of segmenting lane and road markers semantically. However, they used video as the basis for the segmentation using mask cropping. They mentioned the importance of clear and high-resolution images to generate better segmentation and more accurate information for the road structure, aligning with the methodology used in this research.

One of the findings in this approach is the utility of the segmentation semantic information, which relates to the vehicle trajectory spatial analysis, particularly in determining object type categories and trajectories of each object based on their data point locations within the segmented road network. The geopackage of the road network carries important information for this stage, such as permissible directions and lane configuration, which both have proven to be able to categorize object types and analyse their trajectory pattern. This segmentation facilitated the implementation of the hybrid map-matching method, which significantly improved the accuracy of the map-matching process. By integrating the semantic information of road segmentations, including the road segments and allowed connections (illustrated in Figure 22 on page 46), with the sumolib algorithm proximity-based approach, the hybrid method overcame the limitations when using sumolib alone. Specifically, the detailed permissible direction information from the road segmentation allows for more precise map-matching of objects with incomplete trajectories. This is achieved by filtering the array of data points' lane choices based on the permissible connection of each lane. This level of detail ensured that the map-matching algorithm could reliably place objects within the correct segments of the road network, leading to a significant reduction in matching errors and an overall improvement in the quality of the spatial data.

5.1.4. Finding 4: Vehicle trajectory spatial analysis

In the vehicle trajectory spatial analysis, during the initial classification of object types shown in Figure 23 on page 47, the visualization shows that possible vehicles often follow structured road paths within the road segments. In contrast, non-vehicle objects display movement patterns primarily outside these segments, indicating pedestrian movements along curbs and sidewalks at intersections. The distribution of the object types reveals that a significant portion of the tracked objects are possibly vehicles (309 in number), which may indicate the movement of vehicles or a person within the road segments. While most objects are assumed to be vehicles, the presence of a crossings in the intersection still leaves a

chance for tracked objects within the road segments to be classified as persons. Hence, the category is named possible vehicle. This categorization includes anomalies, such as four cars and four two-wheelers classified as non-vehicles. This is likely due to inconsistent labelling, classification errors, or the fact that some vehicles moving in a residential road network, which are not part of the study area, have been captured by the LiDAR sensors.

Aside from the object type categorization, the trajectory of each tracked object is analysed as well. These tracked objects' movement patterns are categorized into complete, (short) complete, violation, and incomplete trajectories. The results in Figure 25 on page 48 show that the dominant categories are incomplete trajectories (500 out of 516 records). This underscores that in the data collection process, sensor range or the duration of data recording might affect the trajectory information of an object. Categorization of incomplete trajectories helps the process of map-matching. This information allows the map-matching algorithm to focus only on incomplete trajectories, reducing the computation cost of the process itself. The detection of violation trajectories (two records), which indicate illegal movements such as U-turns, helps improving the traffic simulation framework. Knowing the tracked objects that have illegal movements helps with preliminary filtering before the objects themselves are loaded into the traffic simulations. SUMO does not have the capability of loading a vehicle that does not follow the proper road segment movements, such as U-turns. As for the (short) complete, this helps in the process of adjusting the necessary speed configurations in the traffic simulations by adjusting the speed based on the close proximity of the data points to the main intersection junction. The adjusted speed allows the tracked objects with this category to be loaded properly in the traffic simulations.

5.1.5. Finding 5: Random forest reclassification of unknown class

Categorization of object type and trajectory type has proven to improve the performance of the random forest model to reclassify the unknown class. These additional categorial features, aside from the original features from the .osef dataset, such as speed and volume, have improved the model accuracy. This is proven by the high accuracy of 0.998 achieved during the initial model training, along with high precision and recall scores, indicating the RF model's good generalization capability. Although in the initial model, the recall and F1-score are slightly lower than those of other classes, this is expected since the RF model could potentially reclassify a known class to another specific known class (e.g., two-wheeler to person or car). This happens because the RF model is robust and able to learn the pattern of each frame in detail, including speed, volume, categorical features (object type and trajectory type), and newly engineered features (speed change, max speed, min speed, acceleration, deceleration, etc.). Therefore, slight pattern changes that are more inclined toward a known class affect the reclassification process.

The study by Ramdani et al. (2022) has indicated that random forest is well-known for its high classification accuracy and robustness in handling outliers and noise in data. This aligns with the results of this research, that the RF model is able to handle the complex information provided by the .osef dataset and achieve high accuracy during the initial model training. Additionally, Zafar & Haq (2020), mentioned that random forest classification algorithm exhibited the highest prediction accuracy of 92% compared to other algorithms like XGBoost and KNN. This means that naturally, for classification purposes, RF models are a good option due to their robustness and ability to predict better than other machine learning classification algorithms. RF is specifically suitable for .osef dataset due to the detailed and rich information it provides for each tracked object. However, since the purpose of the RF model in this research is to reclassify the unknown class, it is important to check that the model did not predict an object to an unknown class. In the result section, this is done by testing the trained model in the second testing set. The accuracy significantly dropped to 0.563. However, this is not an issue because it means that the model properly predicted a label as specific classes, aside from the unknown. Since there is no TP rate of prediction from unknown back to unknown labels, there is bound to be a decrease in overall accuracy. This proves that the model is able to properly reclassify the unknown class.

On the other note, the issue of object IDs with multiple classes still persists. Based on the results, hyperparameter tuning is chosen to minimize this issue. Although the initial RF model has shown high accuracy, when the model is used on new unseen data, it still predicted 88 objects as having two classes and 11 objects as having more than two classes. The aim of this reclassification methodology is to have one specific class for each tracked object so that it can be utilized and simulated well in the SUMO traffic simulation. To achieve that, the RF model needs to be able to reclassify the classes with a minimum error, including the possibility of multiple class predictions. Hyperparameter tuning is chosen in this research to handle this problem. By tuning the RF model, the goal for the model is to be able to specifically predict one class label for an object.

The first tuning method, shown in Figure 28 on page 51, demonstrates a significant reduction in multiple class predictions: from 88 objects having two classes, it is reduced to 36 objects having two classes and four objects having more than two classes. The second tuning method, shown in Figure 29 on page 52, however, shows a significant improvement from the previous model. In this 2nd tuned RF model, it shows that there is only one object having more than two classes and 21 objects with two classes. This means that the model able to generalize better compared to the other version. For example, object 5324303 is predicted with more than 2 classes in the initial RF model and the 1st tuned RF model. This particular object shows a pattern of a motorcycle or a bike with a speed of more than 10 km per hour and a volume lower than 8 m^3. In the initial model and 1st tuned model, this object 5324303 is predicted as either person, car, or two-wheeler at the same time; however, the 2nd tuned model is able to generalize it and predict this object as two-wheeler throughout the array of data points. A study by Probst et al. (2019) emphasized that although random forest models generally perform well out of the box, fine-tuning the hyperparameters can lead to significant performance gains; this further supports the decision of the hyperparameters tuning in this research to improve the RF model ability to properly reclassify the unknown class. This result is crucial for subsequent data analysis and applications, ensuring that each object ID has a single, reliable class label.

5.1.6. Finding 6: Database processing

The previous findings ensure a comprehensive and enriched data-storing process in this research. The database storing process signifies an important part of the DT framework. This process ensures that the dataset issues have been addressed before uploading to the database. The addressed issues mainly include the incomplete trajectory map-matching using the road segmentation results and the unknown class reclassification process with the trained RF model. The processes are implemented in the framework, which means that for new .osef data stored in the database, the framework effectively fills gaps in the data. This results in a more complete and accurate dataset, which is crucial for the traffic simulation and subsequent analysis in this research. The integration of weather data using OpenWeatherMap API also enriched the dataset with additional contextual information. By linking weather conditions, such as temperature and humidity, the dataset now allows for analysis that consider the impact of weather on traffic patterns. Furthermore, it helps on improving the realistic representations of the traffic simulations, by applying logic based on the weather conditions that adjust the speed limit in the intersection. Nguyen et al. (2022) have also suggested that integrating online services such as weather API can provide real-time data on weather conditions, which can affect traffic patterns, speed limit and altitude, thereby improving the specificity and accuracy in traffic simulations, especially route planning. This aligns with the concept that leveraging weather API can contribute more to realistic traffic simulations.

The PostGIS database structure, detailed in Table 10 on page 53, supports extensive geospatial and temporal information. The detailed structure allows for precise tracking of object behaviours over time and across different conditions. The inclusion of dynamic attributes such as speed changes and acceleration provides a nuanced understanding of traffic dynamics while also smoothening the process of XML configuration for the traffic simulations in SUMO since all the information needed is available in the

database. The enriched dataset is not only useful for the .osef data integration to the traffic simulation in SUMO, but it has the potential to be used for various geospatial and temporal analyses if needed.

5.1.7. Finding 7: Traffic simulation configuration

In preparing for traffic simulations, the initial configuration of the SUMO environment is established. Adjustments to the raw OSM road network data are necessary to reflect the real-world conditions more accurately. OSM road network data in itself did not show the correct shape of the road network in the intersection, as seen in Figure 31 on page 55. However, OSM provides accurate information such as road names and speed limits for each road segment, which is essential for the traffic simulation. The alignment of the SUMO road network with the segmented road data ensured that the simulations would be realistic and reliable. A study by Meng et al. (2022) supported this statement; they discussed the importance of preserving the network topology when simplifying OSM data for simulations in SUMO. This highlights that the adjustment is important for achieving accurate representation of traffic flow and integrity within the simulation environment.

The traffic simulations conducted using both XML-based and dynamic approach with TraCI demonstrated the versatility of SUMO as traffic simulations. The XML-based approach ensured a structured and detailed simulation, down to the lane number. This method allowed for precise assignments of lanes and accurate representation of the vehicle movements. The dynamic approach on the other hand, fetched data directly from the database in real-time, managing the simulation updates dynamically, lowering the computational cost and the storage cost since this approach did not save the simulation input and output locally. However, it has disadvantages in lane designation, as mentioned before, underscoring the complexity of real-time route distribution. Overall, the DT traffic simulation framework, shows that the integration and connection between the raw .osef dataset, stored in the PostGIS database, shows a good result. The majority of the tracked objects are simulated without many issues, mainly due to the detailed and specific methodology employed in this research, particularly in handling the .osef dataset.

The traffic simulations with SUMO allow for real-time interactions with the simulations. These interactions include closing a lane in real-time to observe the effects, following specific vehicles to provide a third-person view for better understanding of object movements throughout the road network, removing certain vehicles when they get stuck, and identifying the real geo-location of vehicles in the simulation environment when the road network is geo-referenced. The benefits of these real-time interactions are significant: they improve the understanding of traffic dynamics and vehicle behaviours, by allowing for immediate adjustments and observations. This also facilitate the identification and resolution of potential issues as they occur. In addition to these real-time interaction features, SUMO also enables the testing of what-if scenarios. This allows for the examination of different conditions that cannot be identified with real-time data, such as the long-term planning of changing road structures and its impact on traffic conditions. These capabilities provide a robust DT framework for traffic management and urban planning, offering valuable insights into both immediate and future traffic scenarios.

5.1.8. Finding 8: What-if scenario testing

The what-if scenarios chosen in this research include a standard roundabout, an 8-shaped roundabout, and a two-level intersection with an underpass. Each of them is designed to assess the adaptability of the dataset and the simulation model in varying traffic conditions and junction configurations. The transition from the original intersection to a standard roundabout has no difficulties, with the dataset adapting well due to the similarity in the overall road segments. Furthermore, the additional connections provided by the roundabout structure give more varied connection options for the vehicles in the traffic. The roundabout had continuous traffic flow and reduced stop-and-go conditions,

indicating a significant reduction in delays and smoother traffic movement compared to the original intersection. The 8-shaped roundabout on the other hand, introduced complexity due to the removal of several lanes, including the north-east turning lane and inward lanes from the north-east and south road segments. The result highlights the challenges of adapting to more complex junction modifications. The removal of certain lines does affect the congestion and traffic flow in the road segments; however, it seems that the existence of a double roundabout junction helps in mitigating this expected congestion and therefore makes the traffic flow better. This scenario shows that while roundabouts can enhance traffic flow, in theory, their design must account for sufficient lane capacity to prevent congestion.

The two-level intersection is the final what-if scenario tested in this research. This scenario features an underpass for east-west traffic, inspired by the Save Sofia proposal, maintaining uninterrupted east-west traffic while allowing surface-level north-south traffic. Due to its distinct structure compared to the other designs, this scenario can only accommodate vehicles with trajectory sequences from southeast to northeast road segments and vice versa, or from northern to southern road segments and vice versa. These are the only possible connections in this case, due to the separation between east-west and north-south traffic. However, the highlight of this scenario is its ability to demonstrate the traffic simulation's flexibility in handling varied traffic structures. The 2D view in this case did not provide perfect realism and visibility. In the 2D view, the interchange or crossover of the multi-level intersections appears jumbled and overlapping, without a clear distinction between levels. This hinders the user's ability to understand and monitor the traffic in this scenario, as it is difficult to discern which roads are below ground (underpass) and which are at ground level. Therefore, the 3D view provided a much clearer representation of the elevations, properly showing which roads are underpasses and which are at ground level by highlighting the elevation differences in 3D space. This emphasizes the importance of using a 3D view in certain scenarios.

To determine which scenarios performed better in handling the real-world traffic data from the .osef dataset, a comparison of traffic flow over time across different junction shapes is conducted. A multi-level intersection scenario is not included due to the vast differences in traffic conditions. The comparison included junctions with and without Traffic Light Systems (TLS). Measurements are taken over 10-minute periods, with a frequency of 30 seconds. The results showed that the 8-shaped or double roundabout without TLS consistently had the highest traffic flow, indicating efficient movement and minimal delays. Conversely, the standard roundabout without TLS, initially followed but gradually surpassed the 8-shaped roundabout, indicating that this scenario, over time, has a better chance of mitigating congestion and delays, resulting in high traffic flow.

An important observation, based on Figure 38 on page 58, is that both roundabout scenarios have significantly lower traffic flow when a TLS system is implemented. This highlights that a TLS system is not suitable for roundabout designs. The design of roundabouts inherently considers several traffic conditions to prioritize reducing delays and congestion. Therefore, adding an extra TLS system disrupts the traffic dynamics, leading to inefficiencies. Atapauccar et al. (2022) support these findings; their study highlights that poorly designed roundabouts, including the addition of unnecessary traffic lights, can result in inefficient traffic self-regulation, leading to congestion and increased travel times. This suggest that the incorporation of TLS systems in roundabouts may not be suitable and can have adverse effects on traffic management.

Another point to note is that the original road network structure of the study area shows opposite results. The results indicate that the original road network without traffic lights has the lowest traffic flow, with significant declines compared to when traffic lights are implemented, suggesting frequent traffic congestion. This means the original road network of the intersections might not handle traffic well without the addition of TLS systems, highlighting the importance of traffic lights at intersections. This aligns with a study by Z. Li et al. (2017), where they highlight the critical role of TLS in mitigating

congestion and reducing emissions. The absence of it in a busy intersection can cause unregulated traffic resulting in a high congestion rate.

5.1.9. Finding 9: DT framework assessment

The overall DT framework is assessed at the end of this research. This assessment includes validation of the simulated data and comparing the trajectories from the simulation with the real-world data. The scatter plot comparing observed traffic counts to simulated counts in Figure 39 on page 59 shows a strong alignment between the simulation and real-world traffic patterns. Overall, the high R-squared value and relatively low RMSE indicate that the traffic simulation closely mirrors real-world traffic conditions. The high R-squared value indicates that 97% of the variability in observed traffic counts is explained by the simulated counts, demonstrating a high degree of correlation. For each of the road segments, according to the overall RMSE of 11.34, there is a difference in vehicle counts of around 11 to 12 traffic units. This implies that while the simulation closely matches the observed traffic, there are still minor deviations. These deviations can be attributed to inherent variability in real-world traffic, errors during traffic observation, or inaccuracies in the reclassification process with the RF model, meaning that some objects might not be reclassified properly into their respective classes based on real-world conditions. Despite the noted variability, the overall similarity rate between the simulation and the observed traffic data is acceptable. This high degree of correlation signifies that the simulation is a reliable representation of real-world traffic conditions.

The other assessment involves trajectory comparison between the simulated trajectories and the real-world trajectories. The use of cosine similarity (CoSim) in this research provides a deeper understanding of the routing algorithm's performance within the traffic simulation. High CoSim values closer to 1 for most object IDs indicate that the simulated trajectories follow similar directional patterns as the real-world trajectories, reinforcing the simulation accuracy in capturing vehicle movement directions. However, the slightly lower CoSim values for some objects suggest minor position deviations supported by the observed Euclidean distance values. The slight differences in shape do not significantly affect the CoSim values. However, object IDs such as 5324579, illustrated in Figure 41 on page 62, may exhibit visual discrepancies in trajectory shapes. This still indicates that the overall directional movements are similar to those of real-world trajectories. This assessment highlights areas where the simulation model can be further refined to improve its precision in replicating exact vehicle trajectories. For example, employing other metrics, such as Dynamic Time Warping (DTW), which measures the minimum distance between trajectories while accounting for possible shifts and distortions in the time dimension, can provide a different perspective on trajectory similarity and further identify areas for improvement.

5.2. Implications

The overall traffic simulation digital twin framework developed in this study shows a successful attempt at integrating the .osef dataset into a detailed and comprehensive traffic simulation model. This framework demonstrates a robust capacity for handling detailed traffic data and transforming it into usable simulations, proving its potential adaptability to other intersections with similar characteristics. The methodology applied throughout this research ensures that the DT framework can be effectively used to other areas with similar features and characteristics to the Paradise Center Mall intersection. However, the road segmentation, which is a crucial part of this framework, needs to be configured accurately in order to represent the target intersection characteristics. On that note, as long as the segmentation method aligns with the approach used in this research, the framework functions effectively.

The DT framework's ability to integrate the .osef dataset into traffic simulations signifies a major advancement in urban traffic management tools. This framework not only shows potential in the context of the current study area but also shows potential for broader application such as further traffic analysis

using the enriched databases. The advantages of these traffic simulations lie in the ability to simulate real-world traffic, identify potential congestion and changes in traffic flows, and evaluate the impacts of various traffic management strategies before their implementation. This capability can improve overall urban mobility and enhance the decision-making process in urban traffic management.

The random forest reclassification model trained in this research proves to be a valuable tool for handling new, unseen .osef data. The DT framework supported by this trained machine learning model addresses gaps in dataset completeness by reclassifying unknown classes into specific object classes (car, truck, two-wheeler, or person). Although there is room for improvement, the current model acts as a reliable method for improving the dataset quality and ensuring accurate traffic simulations. This is particularly important for maintaining the integrity and reliability of traffic data, which forms the foundation for effective traffic management strategies. This reclassification model ensures that the traffic simulation in the DT framework remains accurate to the real-world conditions, giving detailed insights of the traffic conditions in the intersection.

The enriched PostGIS database developed in this research allows for more extensive analysis beyond traffic simulations. The simulation framework can provide a more realistic model of traffic conditions by integrating real-time environmental data, such as weather conditions. This comprehensive dataset supports a wide range of analyses, from understanding the impact of weather on traffic patterns to planning for emergency responses and optimizing public transportation schedules. Incorporating diverse data types, enhances the model's utility and ensures that it can address complex urban mobility challenges.

The what-if scenario testing conducted in this study reveals an important insight into the effectiveness of different intersection designs. The comparison of standard roundabouts, 8-shaped roundabouts, and original intersections provides a comprehensive evaluation of potential traffic management solutions. The findings suggest that roundabouts, particularly those without TLS, offer the most effective solutions for improving traffic flow and minimizing congestion. Both the standard roundabout and the 8-shaped roundabout scenarios show good traffic flow with minimal congestion. However, it is essential to note that a longer observation period might be necessary to fully understand the implications of these scenarios. A 10-minute simulation may not capture all the characteristics of intersection traffic, and extended simulations could provide more comprehensive insights. This approach allows traffic managers, transportation planners, or urban planners to test their road design structure and see how it is affecting the traffic flow in the area.

The multi-level intersection scenario, while it demonstrates a realistic design and construction feasibility towards the study area, presents challenges due to the presence of an underground metro station. Although this scenario, in theory, is able to manage traffic flow, constructing an underpass for east-west traffic is not recommended due to potential conflicts with the metro infrastructure. Instead, constructing a fly-over road for north-east traffic may provide a more feasible and effective solution. This consideration shows the importance of contextualizing traffic management solutions within the specific characteristics and constraints of the study area. By tailoring it to the unique characteristics of the intersection, urban planners can develop more effective and sustainable traffic management strategies.

Overall, this validated traffic simulation DT framework shows potential as a valuable tool for urban planning. Urban planners can leverage the model to simulate various traffic scenarios, test different intersection designs, and evaluate the impact of proposed infrastructure changes before implementation. This proactive approach allows for the identification and mitigation of potential issues, ensuring that urban infrastructure projects are optimized for efficiency and effectiveness. The traffic simulation DT framework aligns well with the objectives of Sofia's Sustainable Urban Mobility Plan (SUMP) for 2019-2035. The SUMP aims to create an integrated transportation system by advancing green transportation options, digitalizing city transportation, and encouraging sustainable transportation modes (Modijefsky,

2023). By providing a comprehensive tool for analysing and optimizing traffic flow, the DT framework supports the SUMP goals of reducing the negative effects of transportation development, improving the urban environment's attractiveness, and raising living standards. The framework's emphasis on accurate data-driven simulations and its potential for real-time traffic management align with the SUMP focus on data-driven shared mobility and environmentally friendly urban transportation.

5.3. Limitations

Despite the successful implementation of the DT framework and promising results in this research, several limitations exist that needs to be addressed in future research and development. One of the primary limitations is based on the .osef dataset's inherent challenges. The nature of the dataset that only has local coordinates contains multiple classes within an object ID, and a high ratio of unknown class types has been the first limitation of this research. However, these issues are solved through the methodological approach in this research. The local coordinates issues, for example, have successfully transformed into the respective geo-coordinates using the WKT information. As for the class problem, this research successfully addressed these issues using machine learning approaches of random forest. The RF model has properly reclassified the unknown type of object class and minimized the prediction of multiple classes in an object ID. Although there is still room for improvement, it has shown an acceptable performance and generalization.

Currently, the direct streaming of the .osef dataset is restricted to a private network, making it challenging to access the data directly for real-time analysis. While the parsing method used in this research is adaptable for streaming data, it requires adjustments, such as incorporating network credential information. In the future, developing the OSEF network into an accessible API system would significantly improve its usability and facilitate broader access for public and research purposes. The time span of the dataset used in this research is not extensive due to the substantial size of the .osef dataset when stored locally. Analysing datasets larger than one hour is not computationally viable with the current local setup. Consequently, this research relied on sample data, which, while appropriate for initial analysis, limits the ability to observe long-term traffic dynamics comprehensively. A larger dataset would enable more detailed observations and validations, potentially solidifying the performance and reliability of the traffic simulation DT framework.

Although the overall Digital Twin (DT) framework has shown a successful attempt at integrating the .osef dataset and creating detailed traffic simulations, several limitations exist that should be considered. Due to the complexity of the road network, slight misalignments between the road segmentation and the SUMO road network can result in some vehicles not being properly loaded. This issue is evident in object IDs categorized as having violation trajectory types. Another example is when the position of data points is slightly outside the defined road segments, causing them not to be properly loaded into the simulation. Currently, SUMO does not support loading vehicles that have violation types of trajectories or those that do not follow the permissible allowed connections. While these issues are not major in this research, they are worth considering during the initial configuration of the DT framework to ensure accurate and comprehensive simulations.

Finally, the approach in this research is not fully automated for now, particularly concerning road segmentation. The need for detailed semantic information, including lane permissible connections, makes it difficult to automate this process fully. Currently, there is no suitable methods that can handle the level of detail required for road segmentation in this research. Developing automated tools for road segmentation that can incorporate detailed semantic information is crucial for scaling and enhancing the efficiency of the DT framework.

5.4. Future works

This research opens to extensive possibilities for future research and development to improve its performance, scalability, and applicability. Future work should focus on establishing the overall DT framework in different intersections to further test its adaptability. This includes extending the application to a bigger scope, such as including a residential road in the observation. This consideration can help validate the DT framework further regarding its versatility and effectiveness across diverse urban settings. While the current RF model effectively handled the issues with the object classes in the .osef dataset, where it is able to reclassify unknown classes and minimize multiple class predictions, it is not perfect. Future research should explore advanced methods to further improve this model. This could involve hybrid artificial intelligence approaches or deep learning techniques to improve classification results and handle multiple class predictions more effectively close to zero. These improvements ensure that the model can manage better the .osef dataset with greater precision.

Since this research has proven the usability of the traffic simulation DT framework, to gain a more comprehensive understanding of traffic patterns, future studies may utilize the DT framework for extended observation periods, such as a week or more. Observing traffic dynamics over longer timespans helps to identify congestion patterns, peak hours, and the effects of various unexpected conditions (such as an eventful national day) on traffic flow. This provides deeper insights into the temporal variations in traffic behaviour, enabling more effective traffic management strategies.

In addition, establishing a proper pipeline directly from the .osef streams when it is accessible for public will improve the DT framework process. For now, the PostGIS database containing detailed and robust information of the traffic conditions in the intersections allows for more potential future works, where exploration of this database for extensive analysis can be done. The possibility of integrating the traffic simulation DT framework into a broader city model could be an exciting future direction as well. This would involve dynamic simulations that interact with various urban elements, such as public transportation systems and pedestrian flows. Exploring the platforms and approaches that can handle SUMO traffic simulations and dynamic object data will be crucial for this. This can contribute further to the development of smart cities.

5.5. Summary

This discussion chapter analyzes the DT framework's integration of the .osef dataset for detailed traffic simulations. The framework shows robust accuracy and adaptability, enhancing urban traffic management and planning. Key findings include effective coordinate transformation, accurate vehicle trajectory analysis, and the positive impact of what-if scenarios like roundabouts on traffic flow. The enriched PostGIS database supports extensive geospatial analysis and simulation. The DT framework aligns with Sofia's Sustainable Urban Mobility Plan (SUMP) to improve urban mobility and living standards. However, limitations include the complexity of the .osef dataset, challenges with direct streaming, and the need for longer dataset time spans. Future work should enhance the RF model, extend simulation scalability, incorporate advanced AI methods for better object classification, and integrate the DT framework into broader city models for comprehensive urban traffic management.

6. CONCLUSION

This concluding chapter draws comprehensive conclusions based on the formulated objectives and research questions that guided this research. It evaluates how effectively the research has addressed the initial goals and inquiries. By reflecting on the outcomes, this chapter provides a concise concluding statement of the research contributions to the urban and transportation field.

The research began with a review of the relevant literatures, including tools, models, and methods for traffic simulations. This review highlighted best practices and notable case studies, emphasizing the importance of microscopic simulation for detailed modelling of individual vehicle movements at intersections. SUMO emerged as the most suitable tool due to its extensive capabilities, flexibility, and open-source nature.

Processing and enriching the LiDAR dataset was the next focus. The OSEF dataset contained detailed information about objects at intersections but had issues such as Cartesian coordinates and many "unknown" classes. The methodology included transforming local coordinates to geo-coordinates and generating semantic information about road segments. This information was used to train a Random Forest model to reclassify unknown classes, ensuring smooth integration of the OSEF LiDAR data into the digital twin framework. In this research, the PostGIS database was used as middleware for this integration. It facilitated data storage and fetching, ensuring seamless data flow to the traffic simulation.

The development of the traffic simulation digital twin framework involved using SUMO for intersection simulations. Real-world road information was fetched from OpenStreetMap, with adjustments made to enhance realism. Using the PostGIS database as middleware, the traffic simulation could fetch data directly for model input. The approach adopted involved XML-based methods and dynamic interaction using TraCI. Running various what-if scenarios revealed that in a 10-minute simulation, an 8-shaped roundabout exhibited better, and more stable traffic flow compared to other scenarios. However, the standard roundabout showed superior traffic flow over the total observation period. Additionally, the application of Traffic Light Systems to roundabout scenarios significantly reduced traffic flow, whereas for intersections, TLS improved traffic flow, highlighting a contrast in effectiveness between the two configurations.

Regarding the type of digital twin suitable for traffic simulation, it was determined that 2D digital twins are generally more effective for broader traffic observation. However, there are specific cases where 3D digital twins are advantageous. For instance, in the what-if scenario of a two-level intersection, the 2D view was insufficient to fully capture the interchange where ground-level roads and an underpass intersected. Conversely, the 3D view provided better visualization of elevation differences and angles, enhancing the accuracy and detail of observations.

Finally, the traffic simulation DT framework accuracy and similarity to real-world traffic were assessed. The framework demonstrated high accuracy in traffic generation and strong similarity in traffic trajectories, evidenced by a high R-squared value of 0.97. While minor deviations were observed, the overall RMSE of 11.34 and a relative error of approximately 19.89% indicate general accuracy. The simulated trajectories closely matched real-world directional patterns, with cosine similarity values close to 1 for most object IDs. These findings underscore the potential of integrating advanced simulation tools and enriched datasets to enhance traffic management and planning, which open possibilities for more efficient and realistic traffic simulations in future research.

7. ETHICAL CONSIDERATIONS

This research, conducted in collaboration with The Big Data for Smart Society Institute (GATE), reflects the institute's vision and intentions by leveraging their proprietary .osef dataset, which is not available for public use. The dataset, owned by GATE, is used under mutual agreements, ensuring ethical handling and data security. The study intentionally chose LiDAR as the method because it does not capture license plates of vehicles or the faces of people, thereby safeguarding privacy and confidentiality. The focus was solely on tracked objects, with no involvement of human subjects. Rigorous data security measures are implemented, and all research activities adhered to the highest ethical standards, ensuring transparency, integrity, and alignment with institutional guidelines.

LIST OF REFERENCES

Alcaras, E., Parente, C., & Vallario, A. (2020). The Importance of the Coordinate Transformation Process in Using Heterogeneous Data in Coastal and Marine Geographic Information System. *Journal of Marine Science and Engineering*, *8*(9). https://doi.org/10.3390/jmse8090708

AlRajie, H. (2018). *Investigation of using microscopic traffic simulation tools to predict traffic conflicts between Right-Turning vehicles and through cyclists at signalized intersections*. https://doi.org/10.22215/etd/2015-11179

Atapauccar, S., Mellado, P., Silvera, M., & Campos, F. de. (2022). *Transit Signal Priority Strategies Applied in Roundabouts to Reduce Conflicts and Vehicular Travel Times*. https://doi.org/10.18687/laccei2022.1.1.195

Barcelo, J. (2010). *Fundamentals of traffic Simulation*. https://doi.org/10.1007/978-1-4419-6142-6

Barcelo, J., & Casas, J. (2005). Dynamic network simulation with AIMSUN. In *Simulation Approaches in Transportation Analysis* (Vol. 31, pp. 57–98). https://doi.org/10.1007/0-387-24109-4_3

Ben-Akiva, M., Koutsopoulos, H., Toledo, T., Yang, Q., Choudhury, C., Antoniou, C., & Balakrishna, R. (2010). Traffic simulation with MITSIMLab. *Fundamentals of Traffic Simulation*, *145*, 233. https://doi.org/10.1007/978-1-4419-6142-6_6

Berbar, A., Gastli, A., Meskin, N., Al-Hitmi, M., Ghommam, J., Mesbah, M., & Mnif, F. (2022). Reinforcement Learning-Based control of signalized intersections having platoons. *IEEE Access*, *10*, 17683–17696. https://doi.org/10.1109/access.2022.3149161

Bessa Jr, J., Magalhães, V., & Santos, G. (2021). CALIBRATION AND VALIDATION OF a VOLUME-DELAY FUNCTION USING a GENETIC ALGORITHM. *Journal of Urban and Environmental Engineering*, *15*(2), 173–179. https://doi.org/10.4090/juee.2021.v15n2.173179

Boonyotsawad, C., Hernandez, J. M. P., & Wee, V. (2022). *Strategies for Recovery: COVID-19 and Urban Transport Policy in Asia.* Asian Development Bank Institute.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Brockfeld, E., Kühne, R., & Wagner, P. (2005). Calibration and validation of microscopic models of traffic flow. *Transportation Research Record*, *1934*, 179–187. https://doi.org/10.3141/1934-19

Burghout, W., Koutsopoulos, H., & Andréasson, I. (2005). Hybrid Mesoscopic-Microscopic traffic Simulation. *Transportation Research Record*, *1934*, 218–255. https://doi.org/10.3141/1934-23

Cameron, G. D. B., & Duncan, G. I. D. (1996). PARAMICS—Parallel microscopic simulation of road traffic. *The Journal of Supercomputing*, *10*(1), 25–53. https://doi.org/10.1007/BF00128098

Chang, G. L., Mahmassani, H. S., & Herman, R. (1985). A macroparticle traffic simulation model to investigate peak-period commuter decision dynamics. *Transportation Research Record*, *1005*, 107–121. https://trid.trb.org/view/270405

Chen, D., Zhu, M., Yang, H., Wang, X., & Wang, Y. (2023). Data-driven Traffic Simulation: A Comprehensive review. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2310.15975

Clemente, M. L. (2022). Building a real-world traffic micro-simulation scenario from scratch with SUMO. *SUMO Conference Proceedings*, *3*, 215–230. https://doi.org/10.52825/scp.v3i.109

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms, third edition*. http://portal.acm.org/citation.cfm?id=1614191

de By, R. A., Georgiadou, P. Y., Knippers, R., Kraak, M. J., Sun, Y., Weir, M. J. C., & van Westen, C. J. (2004). *Principles of geographic information systems (Chapter 4.2 on spatial referencing)* (R. A. de By, Ed.; 2nd Edition, Vol. 1). International Institute for Geo-Information Science and Earth Observation. https://kartoweb.itc.nl/geometrics/Coordinate%20transformations/coordtrans.html

Digital Twin Geohub. (2023). *Digital Twinning for Urban and Rural Environmental Modelling*. https://www.utwente.nl/en/digital-society/research/digitalisation/digital-twin-geohub/#vision

Dijkstra, E. W. (2022). A Note on Two Problems in Connexion with Graphs. In *ACM eBooks*. https://doi.org/10.1145/3544585.3544600

Elks, S. (2021). *How the pandemic has changed our use of public transport*. https://www.weforum.org/agenda/2021/05/report-new-initiatives-needed-to-increase-public-transport-usage/

Espejel-Garcia, D., Saniger-Alba, J. A., Wenglas-Lara, G., Espejel-Garcia, V. V., & Villalobos-Aragon, A. (2017). A Comparison among Manual and Automatic Calibration Methods in VISSIM in an Expressway (Chihuahua, Mexico). *Open Journal of Civil Engineering*, *07*(04), 539–552. https://doi.org/10.4236/ojce.2017.74036

European Commission. (2022). *Bulgaria's recovery and resilience plan*. https://commission.europa.eu/business-economy-euro/economic-recovery/recovery-and-resilience-facility/country-pages/bulgarias-recovery-and-resilience-plan_en

Fan, H., Zipf, A., Fu, Q., & Neis, P. (2014). Quality assessment for building footprints data on OpenStreetMap. *International Journal of Geographical Information Science*, *28*(4), 700–719. https://doi.org/10.1080/13658816.2013.867495

Fang, X., & Tettamanti, T. (2021). Change in Microscopic Traffic Simulation Practice with Respect to the Emerging Automated Driving Technology. *Periodica Polytechnica. Civil Engineering/Periodica Polytechnica. Civil Engineering (Online)*. https://doi.org/10.3311/ppci.17411

Farrag, S. G., El-Hansali, M. Y., Yasar, A.-U.-H., Shakshuki, E. M., & Malik, H. (2020). A microsimulation-based analysis for driving behaviour modelling on a congested expressway. *Journal of Ambient Intelligence and Humanized Computing*, *11*(12), 5857–5874. https://doi.org/10.1007/s12652-020-02098-5

Fellendorf, M., & Vortisch, P. (2011). *Microscopic traffic flow simulator VISSIM* (pp. 63–93). https://doi.org/10.1007/978-1-4419-6142-6_2

Felus, Y. A., & Burtch, R. (2008). On Symmetrical Three-Dimensional Datum Conversion. *GPS Solutions*, *13*(1), 65–74. https://doi.org/10.1007/s10291-008-0100-5

Fernandez, C. (2023). *This city has the worst traffic in the U.S.—and it's actually a good thing: "Congestion shows the economy is moving."* https://www.cnbc.com/2023/08/17/north-america-cities-highest-traffic-delays-inrix-report.html

FHWA. (2023, October 17). *Safety at FHWA*. U S Department of Transportation Federal Highway Administration. https://highways.dot.gov/safety/about-safety

Gavric, S., Erdagi, I. G., & Stevanovic, A. (2024). Environmental assessment of incorrect automated pedestrian detection and common pedestrian timing treatments at signalized intersections. *Sustainability*, *16*(11), 4487. https://doi.org/10.3390/su16114487

Giuffrè, O., Granà, A., Tumminello, M. L., Giuffrè, T., Trubia, S., Sferlazza, A., & Rencelj, M. (2018). Evaluation of Roundabout Safety Performance through Surrogate Safety Measures from Microsimulation. *Journal of Advanced Transportation*, *2018*, 1–14. https://doi.org/10.1155/2018/4915970

Google Street View. (2023, November 18). https://www.google.com/maps/

Gu, J., Jiang, Z., Fan, W. D., Qin, W., & Zhang, Z. (2024). Short-term trajectory prediction for individual metro passengers based on multi-level periodicity mining from semantic trajectory. *Engineering Applications of Artificial Intelligence*, *133*, 108134. https://doi.org/10.1016/j.engappai.2024.108134

Hao, R., & Ruan, T. (2024). Advancing Traffic Simulation Precision and Scalability: a Data-Driven approach utilizing deep neural networks. *Sustainability*, *16*(7), 2666. https://doi.org/10.3390/su16072666

Hart, P., Nilsson, N., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, *4*(2), 100–107. https://doi.org/10.1109/tssc.1968.300136

Hristov, P. O., Petrova-Antonova, D., Ilieva, S., & Rizov, R. (2022). ENABLING CITY DIGITAL TWINS THROUGH URBAN LIVING LABS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLIII-B1-2022*, 151–156. https://doi.org/10.5194/isprs-archives-xliii-b1-2022-151-2022

Huang, Y., Kockelman, K. M., & Truong, L. T. (2021). SAV operations on a bus line corridor: travel demand, service frequency, and vehicle size. *Journal of Advanced Transportation*, *2021*, 1–15. https://doi.org/10.1155/2021/5577500

IIHS. (2021, August 9). *Unusual design slashes injury crashes for Roundabout City*. Insurance Institute for Highway Safety; Insurance Institute for Highway Safety. https://www.iihs.org/news/detail/unusual-design-slashes-injury-crashes-for-roundabout-city

IIHS. (2023, June). *Roundabouts*. Insurance Institute for Highway Safety. https://www.iihs.org/topics/roundabouts

Ilyas, S., Ulusoy, B. E., Kofteci, S., & Albayrak, Y. (2024). Association of Vehicle Count Data Obtained Via Image Processing Techniques Compared with Microsimulation Program Analysis Results. *Research Square (Research Square)*. https://doi.org/10.21203/rs.3.rs-3960480/v1

Intelligent Transport. (2015). *Sustainable development of Sofia public transport*. https://www.intelligenttransport.com/transport-articles/17724/sustainable-development-of-sofia-public-transport/

International Trade Administration. (2022). *Bulgaria transport sector*. https://www.trade.gov/market-intelligence/bulgaria-transport-sector

Khare, M. (2024). Simulation of Traffic Flow under Heterogeneous Traffic Conditions using VISSIM. *Current Trends in Civil & Structural Engineering*, *10*(3). https://doi.org/10.33552/ctcse.2024.10.000736

Kitajima, S., Shimono, K., Tajima, J., Antona-Makoshi, J., & Uchida, N. (2019). Multi-agent traffic simulations to estimate the impact of automated technologies on safety. *Traffic Injury Prevention (Online)/Traffic Injury Prevention*, *20*(sup1), S58–S64. https://doi.org/10.1080/15389588.2019.1625335

Krauß, S., für Luft- und Raumfahrt, D. Z., & und Systemtechnik, H. M. (1998). *Microscopic Modeling of traffic flow: Investigation of collision free vehicle dynamics* (Issues 98–08, p. 115). Deutsches Zentrum für Luft- und Raumfahrt. https://sumo.dlr.de/pdf/KraussDiss.pdf

Kušić, K., Schumann, R., & Ivanjko, E. (2023). A digital twin in transportation: Real-time synergy of traffic data streams and simulation for virtualizing motorway dynamics. *Advanced Engineering Informatics*, *55*, 101858. https://doi.org/10.1016/j.aei.2022.101858

Li, W., Wu, G., Boriboonsomsin, K., Barth, M. J., Rajab, S., Bai, S., & Zhang, Y. (2017). Development and evaluation of High-Speed Differential Warning Application using Vehicle-to-Vehicle Communication. *Transportation Research Record*, *2621*(1), 81–91. https://doi.org/10.3141/2621-10

Li, Z., Shahidehpour, M., Bahramirad, S., & Khodaei, A. (2017). Optimizing Traffic Signal Settings in Smart Cities. *Ieee Transactions on Smart Grid*, *8*(5), 2382–2393. https://doi.org/10.1109/tsg.2016.2526032

Liaw, A., & Wiener, M. (2002). *Classification and regression by RandomForest: Vol. Vol. 2/3*. https://journal.r-project.org/articles/RN-2002-022/RN-2002-022.pdf

Lin, X., Wang, M., & Van Arem, B. (2017). Realistic Car-Following models for microscopic simulation of adaptive and cooperative adaptive cruise control vehicles. *Transportation Research Record*, *2623*(1), 1–9. https://doi.org/10.3141/2623-01

Ma, J., & Fukuda, D. (2014). A Note on Route Planning in Transportation with Open Sources. *Cictp 2014*. https://doi.org/10.1061/9780784413623.028

Maciejewski, M. (2010). A comparison of microscopic traffic flow simulation systems for an urban area. *DOAJ (DOAJ: Directory of Open Access Journals)*. https://doaj.org/article/e167acd4bbb64f8fb309b8c29e5de1b4

Mahmud, S. M. S., Ferreira, L., Hoque, Md. S., & Tavassoli, A. (2019). Micro-simulation modelling for traffic safety: A review and potential application to heterogeneous traffic environment. *IATSS Research*, *43*(1), 27–36. https://doi.org/https://doi.org/10.1016/j.iatssr.2018.07.002

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Medapati, N., Rao, D. M., & Patnaikuni, C. K. (2022). A study on pedestrian safety, vehicular fuel consumption, and emissions using GIS and PTV VISSIM software. *Innovative Infrastructure Solutions*, *7*(5). https://doi.org/10.1007/s41062-022-00909-6

Meng, Z., Du, X., Sottovia, P., Foroni, D., Axenie, C., Wieder, A., Eckhoff, D., Bortoli, S., Knoll, A., & Sommer, C. (2022). Topology-Preserving Simplification of OpenStreetMap Network Data for Large-Scale Simulation in SUMO. *Sumo Conference Proceedings*, *3*, 181–197. https://doi.org/10.52825/scp.v3i.111

Ministry of Regional Development and Public Work of Bulgaria. (2023). *Tables of persons registered by permanent and current address in municipality of Sofia*. General Directorate of Civil Registration and Administrative Services. https://www.grao.bg/tna/isnt41nm-15-06-2023-2.txt

Ministry of Transport and Communications of Republic of Bulgaria. (2017). *Integrated Transport Strategy for the period until 2030*. https://www.mtc.government.bg/sites/default/files/integrated_transport_strategy_2030_eng.pdf

Modijefsky, M. (2023). *Sofia's SUMP 2019-2035: Addressing Urban Mobility Challenges*. https://www.eltis.org/resources/case-studies/sofias-sump-2019-2035-addressing-urban-mobility-challenges

Mohan, R., & Ramadurai, G. (2013). State-of-the art of macroscopic traffic flow modelling. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, *5*(2–3), 158–176. https://doi.org/10.1007/s12572-013-0087-1

Mynuddin, M., & Gao, W. (2020). Distributed predictive cruise control based on reinforcement learning and validation on microscopic traffic simulation. *IET Intelligent Transport Systems*, *14*(5), 270–277. https://doi.org/10.1049/iet-its.2019.0404

Nguyen, T., Kriesten, R., & Chrenko, D. (2022). Concept for Generating Energy Demand in Electric Vehicles With a Model Based Approach. *Applied Sciences*, *12*(8), 3968. https://doi.org/10.3390/app12083968

NHTSA. (2023, April 20). *NHTSA estimates for 2022 show roadway fatalities remain flat after two years of dramatic increases*. National Highway Traffic Safety Administration. https://www.nhtsa.gov/press-releases/traffic-crash-death-estimates-2022

Nikova, A. (2020). *Арх. Игнатов предлага кръгово пред "Парадайс" в памет на Милен Цветков (СНИМКИ)*. https://stolica.bg/mestna-politika/arh-ignatov-predlaga-kragovo-pred-paradais-v-pamet-na-milen-tsvetkov-snimki

Outsight. (2022, October 14). *osefTypes.h*. ROS Documentation. https://docs.ros.org/en/melodic/api/outsight_alb_driver/html/osefTypes_8h_source.html

Panayotova, E. (2022). *Sofia's challenges and solutions in urban mobility, air pollution and urban green areas*. https://www.iurc.eu/wp-content/uploads/2022/01/IURC-Sofia-Semarang-Meeting-13_01_2022.pdf

Park, B., & Schneeberger, J. D. (2003). Microscopic Simulation model calibration and validation: Case study of VISSIM simulation model for a Coordinated Actuated Signal system. *Transportation Research Record*, *1856*(1), 185–192. https://doi.org/10.3141/1856-20

Paulsen, M., Rasmussen, T. K., & Nielsen, O. A. (2022). Including Right-of-Way in a joint Large-Scale Agent-Based dynamic traffic assignment model for cars and bicycles. *Networks and Spatial Economics*, *22*(4), 915–957. https://doi.org/10.1007/s11067-022-09573-w

Pishue, B. (2023). *2022 Global Traffic Scorecard: Congestion is Up Despite High Oil Prices - INRIX*. https://inrix.com/blog/2022-traffic-scorecard/

Probst, P., Wright, M. N., & Boulesteix, A. (2019). Hyperparameters and Tuning Strategies for Random Forest. *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, *9*(3). https://doi.org/10.1002/widm.1301

Qi, H.-S., & Ying, Y.-Y. (2023). A stochastic two-dimensional intelligent driver car-following model with vehicular dynamics. *Chinese Physics B/Chinese Physics B*, *32*(4), 44501. https://doi.org/10.1088/1674-1056/ac8f3c

Ramdani, N., Prasetyowati, S. S., & Sibaroni, Y. (2022). Performance Analysis of Bandung City Traffic Flow Classification With Machine Learning and Kriging Interpolation. *Building of Informatics Technology and Science (Bits)*, *4*(2), 694–704. https://doi.org/10.47065/bits.v4i2.1972

Rangelov, T. (2023, November 17). *Феново пред мол Paradise нападнаха Емили :)*. https://www.youtube.com/watch?v=jFmi3rBElQk

Rashid, A., Umair, M., Hasan, O., & Zaki, M. H. (2020). Toward the formalization of macroscopic models of traffic flow using Higher-Order-Logic theorem proving. *IEEE Access*, *8*, 27291–27307. https://doi.org/10.1109/access.2020.2971661

Rashkov, A. (2023, November 18). *Задръстване мол Парадайс*. https://www.youtube.com/watch?v=_uAGMuzPeLI

Reza, I., Ratrout, N. T., & Rahman, S. M. (2016). Calibration protocol for PARAMICS microscopic traffic simulation model: application of neuro-fuzzy approach. *Canadian Journal of Civil Engineering*, *43*(4), 361–368. https://doi.org/10.1139/cjce-2015-0435

Rodrigues, M., Teoh, T., Ramos, C., De Winter, T., Knezevic, L., Marcucci, E., Lozzi, G., Gatta, V., Antonucci, B., Cutrufo, N., Marongiu, L., & Cré, I. (2021). *Research for TRAN Committee - Relaunching Transport and tourism in the EU after COVID-19*.

Sofia City Council, Bulgarian Swiss Cooperation Programme, B I M Consulting, & Infraproject Consult Ltd. (2019). *Sofia sustainable urban mobility plan (SUMP)*.

The Sofia Globe. (2022a, March 29). *EC: Bulgaria had second-highest road fatality rate in EU in 2021*. https://sofiaglobe.com/2022/03/28/ec-bulgaria-had-second-highest-road-fatality-rate-in-eu-in-2021/

The Sofia Globe. (2022b, June 2). *Bulgaria's road death toll in first five months of 2022 is 175*. https://sofiaglobe.com/2022/06/01/bulgarias-road-death-toll-in-first-five-months-of-2022-is-175/

TomTom International BV. (2023). *Sofia traffic report | TomTom Traffic Index*. https://www.tomtom.com/traffic-index/sofia-traffic/

Treiber, M., & Kesting, A. (2013). *Traffic flow dynamics*. https://doi.org/10.1007/978-3-642-32460-4

Vincent, K. (2023, February 23). *What is a 3D LiDAR Preprocessor?* https://www2.outsight.ai/insights/whats-a-3d-lidar-preprocessor?ref=lidar-insighter.com

Wang, D., Wang, X., Chen, L., Yao, S., Jing, M., Li, H., Li, L., Bao, S., Wang, F.-Y., & Lin, Y. (2023). TransWorldNG: Traffic Simulation via Foundation Model. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2305.15743

Wang, J., Kong, Y., Fu, T., & Stipancic, J. (2017). The impact of vehicle moving violations and freeway traffic flow on crash risk: An application of plugin development for microsimulation. *PloS One*, *12*(9), e0184564. https://doi.org/10.1371/journal.pone.0184564

Wang, L., Guo, Y., Li, H., & Liu, Y. (2024). Mixed traffic flow characteristics analysis based on simulation approach. *Seventh International Conference on Traffic Engineering and Transportation System (ICTETS 2023)*. https://doi.org/10.1117/12.3015694

Warchol, S., Chase, T., & Cunningham, C. (2017). Use of microsimulation to evaluate Signal-Phasing schemes at diverging diamond interchanges. *Transportation Research Record*, *2620*(1), 10–19. https://doi.org/10.3141/2620-02

Wei, F., Guo, Y., Liu, P., Cai, Z., Li, Q., & Chen, L. (2020). Modeling Car-Following Behaviour of Turning Movements at Intersections with Consideration of Turning Radius. *Journal of Advanced Transportation*, *2020*, 1–9. https://doi.org/10.1155/2020/8884797

WHO. (2019, June). *Road safety*. World Health Organization. https://www.who.int/health-topics/road-safety#tab=tab_1

Williams, K., Olsen, M. J., Roe, G., & Glennie, C. L. (2013). Synthesis of Transportation applications of Mobile LIDAR. *Remote Sensing*, *5*(9), 4652–4692. https://doi.org/10.3390/rs5094652

World Bank. (2020). *Best Practice in City Public Transport Authorities' Responses to COVID-19: A Note for Municipalities in Bulgaria*. https://doi.org/10.1596/34051

Wu, J., Radwan, E., Abou-Senna, H., for Advanced Transportation Systems Simulation, C., of Civil Environment Construction Engineering, D., & of Central Florida, U. (2012). *PEDESTRIAN-VEHICLE CONFLICT ANALYSIS AT SIGNALIZED INTERSECTIONS USING MICRO-SIMULATION*. https://www.diva-portal.org/smash/get/diva2:926089/FULLTEXT01.pdf

Xing, G., & Zhu, Z. (2021). Lane and Road Marker Semantic Video Segmentation Using Mask Cropping and Optical Flow Estimation. *Sensors*, *21*(21), 7156. https://doi.org/10.3390/s21217156

Xu, Y., Xie, Z., Feng, Y., & Chen, Z. (2018). Road Extraction from High-Resolution Remote Sensing Imagery Using Deep Learning. *Remote Sensing*, *10*(9), 1461. https://doi.org/10.3390/rs10091461

Yedavalli, P., Kumar, K., & Waddell, P. (2021). Microsimulation analysis for network traffic assignment (MANTA) at metropolitan-scale for agile transportation planning. *Transportmetrica*, *18*(3), 1278–1299. https://doi.org/10.1080/23249935.2021.1936281

Zafar, N., & Haq, I. U. (2020). Traffic Congestion Prediction Based on Estimated Time of Arrival. *Plos One*, *15*(12), e0238200. https://doi.org/10.1371/journal.pone.0238200

Zeb, A., Khattak, K. S., Ullah, M. R., Khan, Z. H., & Gulliver, T. A. (2023). HETRoTraffSiM: A macroscopic heterogeneous traffic flow Simulator for road bottlenecks. *Future Transportation*, *3*(1), 368–383. https://doi.org/10.3390/futuretransp3010022

Zhang, K. (2020, May). *How Urban Transport is Changing in the Age of COVID-19*. https://news.climate.columbia.edu/2020/07/10/urban-transport-changing-covid-19/

Zhao, B., Lin, Y., Hao, H., & Yao, Z. (2022). Fuel Consumption and Traffic Emissions Evaluation of Mixed Traffic Flow with Connected Automated Vehicles at Multiple Traffic Scenarios. *Journal of Advanced Transportation*, *2022*, 1–14. https://doi.org/10.1155/2022/6345404

Zheng, J., Suzuki, K., & Fujita, M. (2012). Modelling a vehicle's speed fluctuation with a cellular automata model. *WIT Transactions on the Built Environment*. https://doi.org/10.2495/ut120321

Zheng, Y. (2015). Trajectory data mining. *ACM Transactions on Intelligent Systems and Technology*, *6*(3), 1–41. https://doi.org/10.1145/2743025

Zhong, Z., Rempe, D., Xu, D., Chen, Y., Veer, S., Che, T., Ray, B., & Pavone, M. (2023). Guided Conditional Diffusion for Controllable Traffic Simulation. *2023 IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/icra48891.2023.10161463

Ziemska-Osuch, M., & Osuch, D. (2022). Modeling the Assessment of Intersections with Traffic Lights and the Significance Level of the Number of Pedestrians in Microsimulation Models Based on the PTV Vissim Tool. *Sustainability*, *14*(14), 8945. https://doi.org/10.3390/su14148945

Спаси София. (2020, December 9). *"Спаси София" предлага да се изгради ново кръстовище на две нива при МОЛ "Парадайс."* https://www.novinite.bg/articles/195721/Spasi-Sofiya-predlaga-da-se-izgradi-novo-krastovishte-na-dve-niva-pri-MOL-Paradajs

Annex 1: Study Area WKT information

```
wkt = 'DERIVEDPROJCRS["",BASEPROJCRS["WGS 84 / Pseudo-Mercator",BASEGEOGCRS["WGS 84",ENSEMBLE["World
Geodetic System 1984 ensemble",MEMBER["World Geodetic System 1984 (Transit)"],MEMBER["World Geodetic System
1984 (G730)"],MEMBER["World Geodetic System 1984 (G873)"],MEMBER["World Geodetic System 1984 (G1150)"],
MEMBER["World Geodetic System 1984 (G1674)"],MEMBER["World Geodetic System 1984 (G1762)"],MEMBER["World
Geodetic System 1984 (G2139)"],ELLIPSOID["WGS 84",6378137,298.257223563,LENGTHUNIT["metre",1]],
ENSEMBLEACCURACY[2.0]],PRIMEM["Greenwich",0,ANGLEUNIT["degree",0.0174532925199433]],ID["EPSG",4326]],
CONVERSION["Popular Visualisation Pseudo-Mercator",METHOD["Popular Visualisation Pseudo Mercator",ID["EPSG"
1024]],PARAMETER["Latitude of natural origin",0,ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8801]],
PARAMETER["Longitude of natural origin",0,ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8802]],PARAMETER
["False easting",0,LENGTHUNIT["metre",1],ID["EPSG",8806]],PARAMETER["False northing",0,LENGTHUNIT["metre",
1],ID["EPSG",8807]]]],DERIVINGCONVERSION["Affine",METHOD["Affine parametric transformation",ID["EPSG",
9624]],PARAMETER["A0",-3716438.1093590977,LENGTHUNIT["metre",1],ID["EPSG",8623]],PARAMETER["A1",0.
6158540384148486,SCALEUNIT["coefficient",1],ID["EPSG",8624]],PARAMETER["A2",0.40263073503547436,SCALEUNIT
["coefficient",1],ID["EPSG",8625]],PARAMETER["B0",-2194507.2465994842,LENGTHUNIT["metre",1],ID["EPSG",
8639]],PARAMETER["B1",-0.40263073503547436,SCALEUNIT["coefficient",1],ID["EPSG",8640]],PARAMETER["B2",0.
6158540384148486,SCALEUNIT["coefficient",1],ID["EPSG",8641]]],CS[Cartesian,2],AXIS["easting (X)",east,ORDER
[1],LENGTHUNIT["metre",1]],AXIS["northing (Y)",north,ORDER[2],LENGTHUNIT["metre",1]]]'
crs_wgs84 = CRS.from_epsg(4326)  # EPSG:4326 for WGS 84 long lat ellipsoid, EPSG:3857 alternative for
mercator -> 2d plane
```

Figure 42. WKT information of the study area intersection. Source: Author (2024).

Annex 2: .osef data overview sample

```
FIRST RECORD: Date 2023-05-30 Timestamp 11:43:58.1045
LAST RECORD: Date 2023-05-30 Timestamp 11:45:24.4901
TIME DIFF: 0 day 0 hour 02 minutes 26.0385 seconds
FRAMES DETECTED: 1726
FPS: 19.98
TRACKED OBJECTS: 516
DETECTED CLASSES: CAR, PERSON, TRUCK, TWO_WHEELER, UNKNOWN

CONSISTENT OBJECTS: 264
---
TRUCK      :   3
UNKNOWN    : 251
CAR        :   9
PERSON     :   1

RECOGNIZED OBJECTS: 241
---
CAR        : 188
TRUCK      :   4
PERSON     :  36
TWO_WHEELER:  13

UNIDENTIFIED OBJECTS: 11
---
5324412: UNKNOWN, TWO_WHEELER, CAR
5324431: UNKNOWN, PERSON, TWO_WHEELER
5324597: UNKNOWN, TWO_WHEELER, CAR
5324894: UNKNOWN, TWO_WHEELER, CAR
5324926: UNKNOWN, CAR, TRUCK
5324944: UNKNOWN, CAR, TRUCK
5324972: UNKNOWN, PERSON, TWO_WHEELER
5324976: UNKNOWN, PERSON, TWO_WHEELER
5325108: UNKNOWN, PERSON, TWO_WHEELER
5325172: UNKNOWN, TWO_WHEELER, CAR
5325213: UNKNOWN, PERSON, TWO_WHEELER
```

Figure 43. .osef sample data overview. Source: Author (2024).

Annex 3: Permissible connection configuration

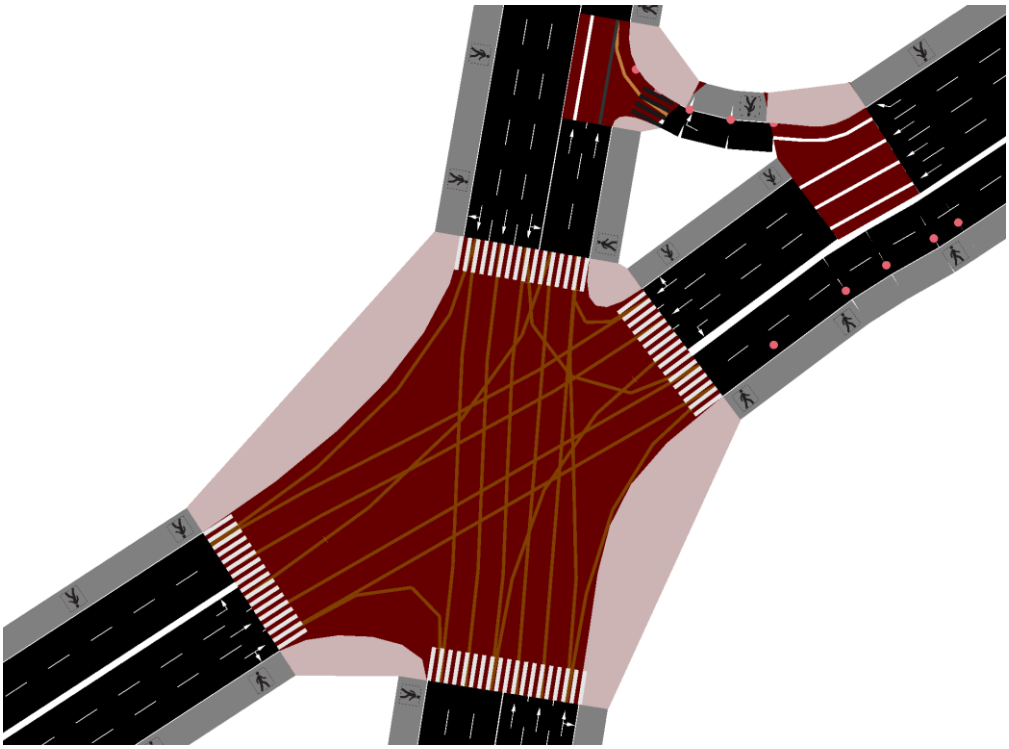


Figure 44. Permissible connection SUMO configuration. Source: Author (2024).

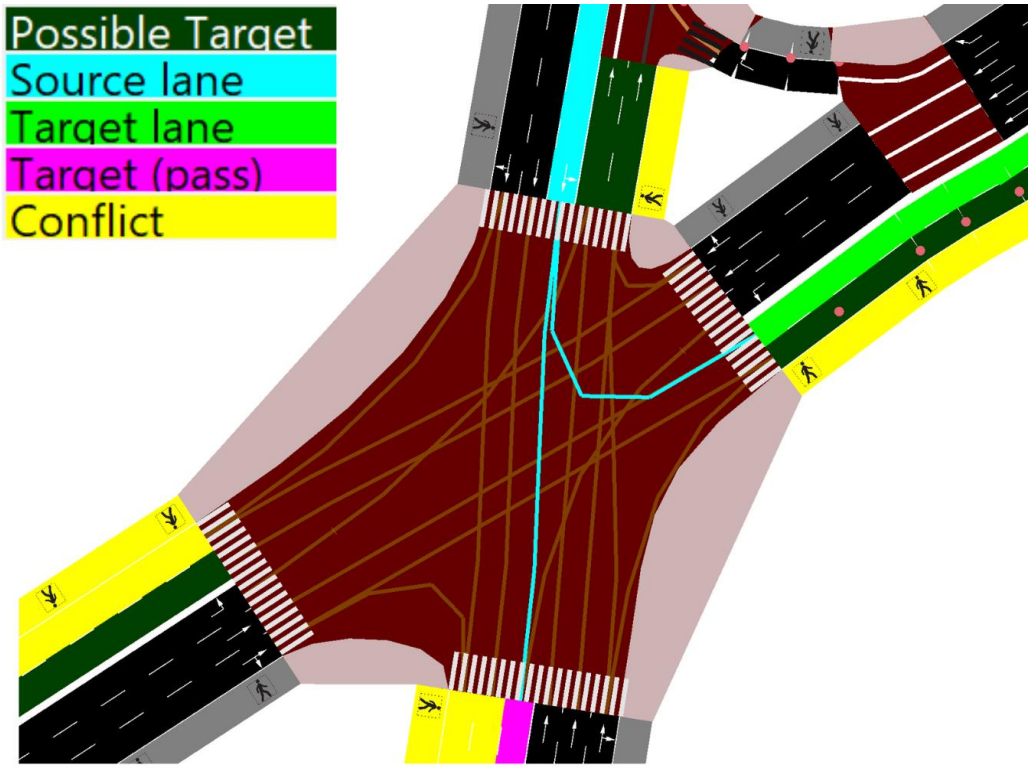
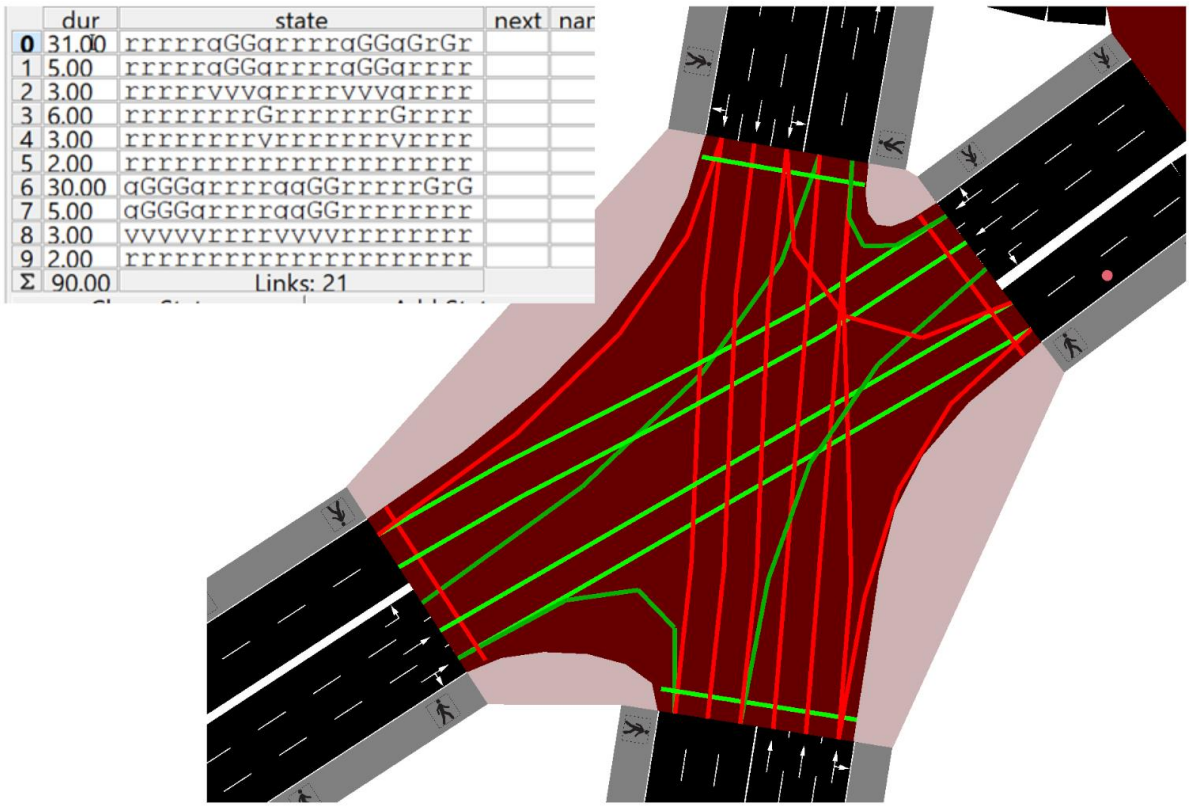


Figure 45. Permissible connection configuration details. Source: Author (2024).

Annex 4: TLS system configuration



	dur	state	next	nar
0	31.00	rrrrraGGarrrraGGaGrGr		
1	5.00	rrrrraGGarrrraGGarrrr		
2	3.00	rrrrrvvvarrrrvvvarrrr		
3	6.00	rrrrrrrrGrrrrrrrGrrrr		
4	3.00	rrrrrrrrvrrrrrrrvrrrr		
5	2.00	rrrrrrrrrrrrrrrrrrrrr		
6	30.00	gGGGarrrraaGGrrrrrGrG		
7	5.00	gGGGarrrraaGGrrrrrrrr		
8	3.00	vvvvvrrrrvvvvrrrrrrrr		
9	2.00	rrrrrrrrrrrrrrrrrrrrr		
Σ	90.00	Links: 21		

Figure 46. TLS systems configuration in SUMO. Source: Author (2024).

Annex 5: Real-world trajectory plot

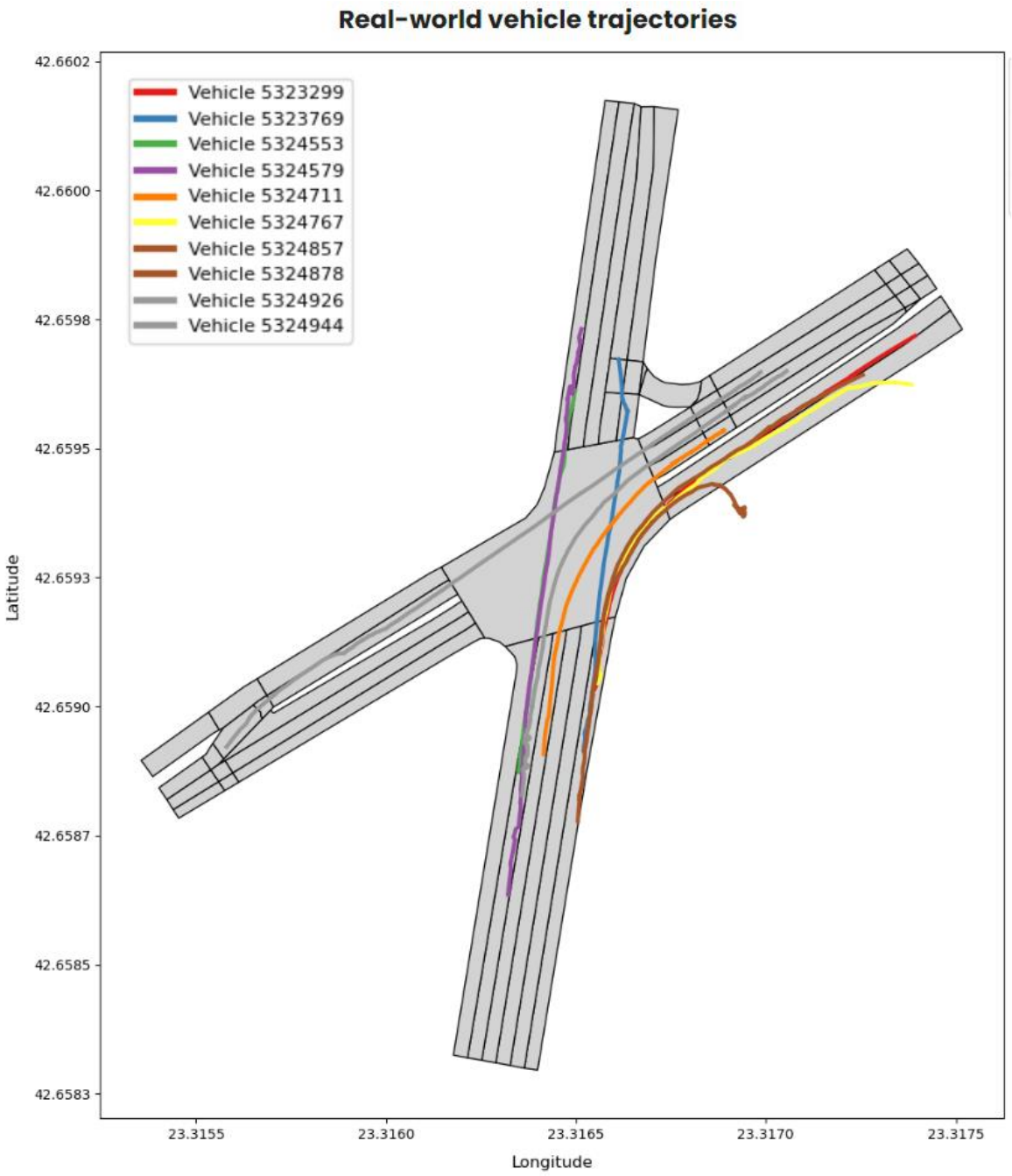


Figure 47. Real-world vehicle trajectories. Source: Author (2024).

Annex 6: Simulated trajectory plot

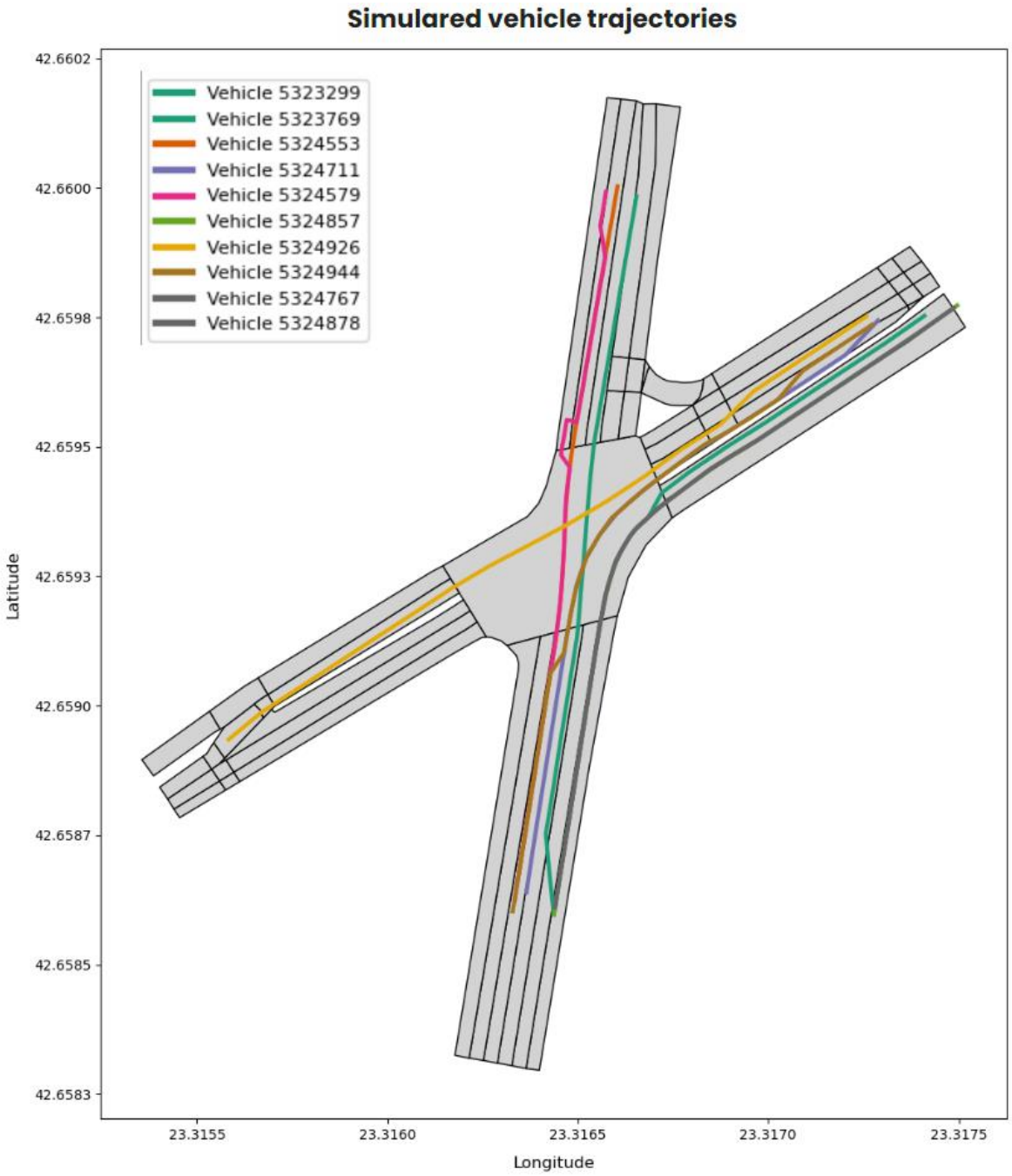


Figure 48. Simulated vehicle trajectories. Source: Author (2024).