COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

ANOMALY DETECTION IN CATENARY COMPONENTS OF RAILWAY SYSTEMS USING INSPECTION IMAGES AND DEEP LEARNING ALGORITHMS

Autor: Lucía Martínez Ruiz

Director: Miguel Ángel Sanz Bobi

Madrid

Mayo de 2026

Declaration of Originality

I hereby declare, under my own responsibility, that the Project submitted under the title **Anomaly Detection in Catenary Components of Railway Systems Using Inspection Images and Deep Learning Algorithms** to the School of Engineering – ICAI of Universidad Pontificia Comillas during the 2025/2026 academic year is my own work and has not been previously submitted for any other purpose. The Project does not constitute plagiarism, either in whole or in part, and all information taken from other documents has been duly referenced.

Use of Artificial Intelligence[1]

I hereby declare, under my own responsibility, that:

☐ I have not used Artificial Intelligence in the preparation of this document.

☑ I have used Artificial Intelligence in the preparation of this document and/or Annex B, always under the conditions permitted by Universidad Pontificia Comillas, applying Level 2 of the Escala de Evaluación de Perkins et al. (2024): "*AI may be used for pre-task activities such as brainstorming, outlining, and initial research. This level focuses on the use of AI for planning, synthesis, and idea generation, but assessments must emphasize the ability to develop and refine these ideas independently*". Specifically, Artificial Intelligence has been used as:

Artificial Intelligence has been used as a support tool during the development of this work, mainly for initial planning, organization of ideas, structuring of some sections, synthesis of technical information, and improvement of academic writing and clarity. It has also been used to review the coherence of the text and to help express certain explanations more precisely. In all cases, the analysis, interpretation of results, methodological decisions, and final conclusions have been developed independently by the author, who assumes full responsibility for the submitted work.

[1] This declaration refers to the use of generative Artificial Intelligence in the preparation of the Project documents, namely Annex B and the main report. It does not apply to Projects in which, due to their nature, Artificial Intelligence must be used as an integral part of the work, such as the application of machine learning techniques, neural networks, data analysis, and similar methods.


Signed (student): Lucía Martínez Ruiz
Date: 27 of May 2026

Authorization for the Submission Project

The Director of the Project	The co-Director of the Project (if applicable)
Firmado por SANZ BOBI MIGUEL ANGEL - ***6599** el día 08/06/2026 con un certificado emitido por AC FNMT Usuarios	
Signed: Miguel Ángel Sanz Bobi	Signed:
Date:	Date:

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

ANOMALY DETECTION IN CATENARY COMPONENTS OF RAILWAY SYSTEMS USING INSPECTION IMAGES AND DEEP LEARNING ALGORITHMS

Autor: Lucía Martínez Ruiz

Director: Miguel Ángel Sanz Bobi

Madrid

Mayo de 2026

Acknowledgements

I would like to express my sincere gratitude to my supervisor, **Miguel Ángel Sanz Bobi**, for his guidance, support, and availability throughout the development of this Master's Thesis. His advice and recommendations have been of great help in shaping this project and carrying it out with academic rigor.

ANOMALY DETECTION IN CATENARY COMPONENTS OF RAILWAY SYSTEMS USING INSPECTION IMAGES AND DEEP LEARNING ALGORITHMS

Autor: Martínez Ruiz, Lucía.
Director: Sanz Bobi, Miguel Ángel.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Este Trabajo Fin de Máster desarrolla un flujo de trabajo basado en deep learning para la segmentación semántica de nubes de puntos de catenaria ferroviaria y la detección de regiones anómalas. El pipeline propuesto abarca la preparación de los datos, el entrenamiento de modelos, la evaluación y el postprocesado orientado a anomalías. Se estudian distintas arquitecturas, siendo RandLA-lite la que ofrece el mejor equilibrio entre calidad de segmentación y eficiencia computacional. El sistema final permite tanto la clasificación de los principales elementos de la catenaria como la localización de regiones sospechosas para tareas de inspección.

Palabras clave: Nubes de Puntos Ferroviarias, Segmentación Semántica, Catenaria, Detección de Anomalías, Deep Learning, RandLA-Net, LiDAR.

1. Introducción

La inspección de infraestructuras ferroviarias depende cada vez más de herramientas basadas en datos capaces de procesar información sensorial compleja. En este contexto, las nubes de puntos LiDAR móviles proporcionan información tridimensional detallada sobre entornos ferroviarios, incluyendo componentes de la catenaria aérea como el hilo de contacto, el cable sustentador, las péndolas y los soportes. Sin embargo, su estructura irregular y gran volumen dificultan su análisis manual.

Este trabajo aborda este problema mediante segmentación semántica basada en deep learning, con el objetivo adicional de extender el marco a la detección de anomalías. El trabajo se basa en el conjunto de datos WHU-Railway3D y se centra en construir un flujo de trabajo reproducible para futuros sistemas inteligentes de inspección [1].

2. Definición del proyecto

El objetivo principal del proyecto es diseñar y evaluar un pipeline completo para la comprensión automática de nubes de puntos de catenaria ferroviaria. El trabajo incluye preprocesado de datos, organización de conjuntos de entrenamiento, validación y prueba, entrenamiento de modelos, evaluación cuantitativa y visualización de resultados.

Además de la segmentación semántica, el proyecto incorpora una etapa de análisis orientada a anomalías. Su finalidad es transformar las predicciones punto a punto en regiones interpretables que puedan indicar discontinuidades, incertidumbre, baja densidad o cambios geométricos bruscos, haciendo la salida más útil para inspección y mantenimiento.

3. Descripción del modelo/sistema/herramienta

El sistema propuesto se estructura como un flujo de trabajo completo para el análisis de nubes de puntos de catenaria ferroviaria. La primera etapa se centra en la preparación de los datos, incluyendo la organización del conjunto de datos, el preprocesado, la generación de particiones y la creación de estructuras derivadas para entrenamiento y evaluación.

La segunda etapa consiste en la segmentación semántica. El sistema evalúa diferentes arquitecturas de deep learning, entre ellas un MLP pointwise de referencia, PointNet [2], PointNet++, RandLA-lite [3] y un modelo de convoluciones dispersas basado en Minkowski. Su rendimiento se compara mediante métricas sensibles a las clases, prestando especial atención a la dificultad de segmentar elementos finos y desbalanceados. Entre todos ellos, **RandLA-lite** se perfila como el modelo más adecuado en los experimentos principales.

La etapa final extiende la segmentación semántica hacia la detección de anomalías. En lugar de limitar la salida a predicciones punto a punto, el sistema agrega los resultados en regiones de interés y las analiza mediante indicadores como incertidumbre, baja densidad de puntos, discontinuidades geométricas y saltos de altura. En la Figure 1 se muestra el flujo de trabajo general propuesto.

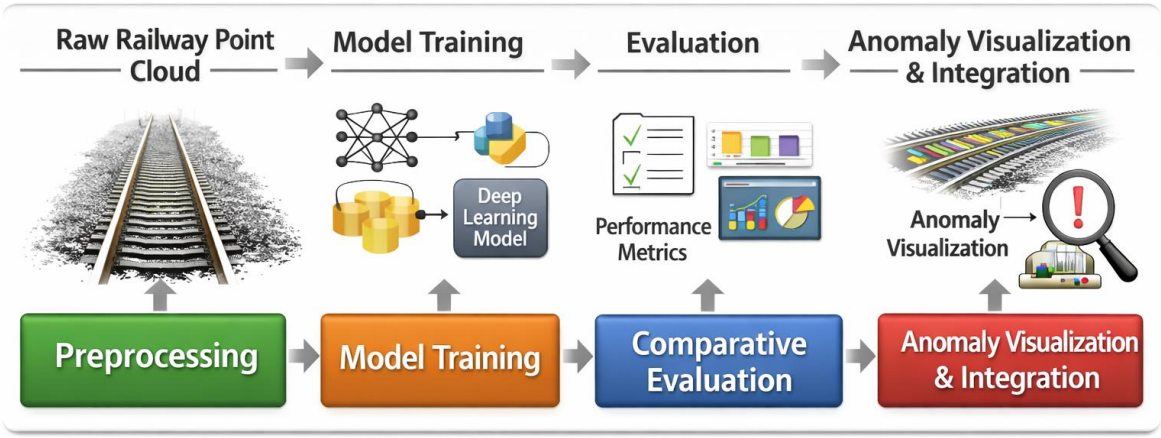

Figure 1: Flujo de trabajo general del sistema propuesto

4. Resultados

Los resultados muestran que la segmentación semántica proporciona una base eficaz para el análisis de nubes de puntos de catenaria ferroviaria. La comparación entre modelos confirma que la tarea es exigente debido al desbalanceo entre clases, la variabilidad de las escenas y la geometría fina de varios elementos de la catenaria.

Entre los modelos probados, el mejor rendimiento en los principales experimentos de validación se obtiene con **RandLA-lite**, que alcanza un valor medio de ***mIoU_present50*** de aproximadamente ***0.53***, manteniendo además un coste computacional más favorable que las alternativas más pesadas.

La Figure 2 muestra el contexto completo de un tile junto con una vista ampliada del intervalo detectado, ilustrando cómo el flujo de trabajo propuesto localiza regiones sospechosas de forma interpretable.

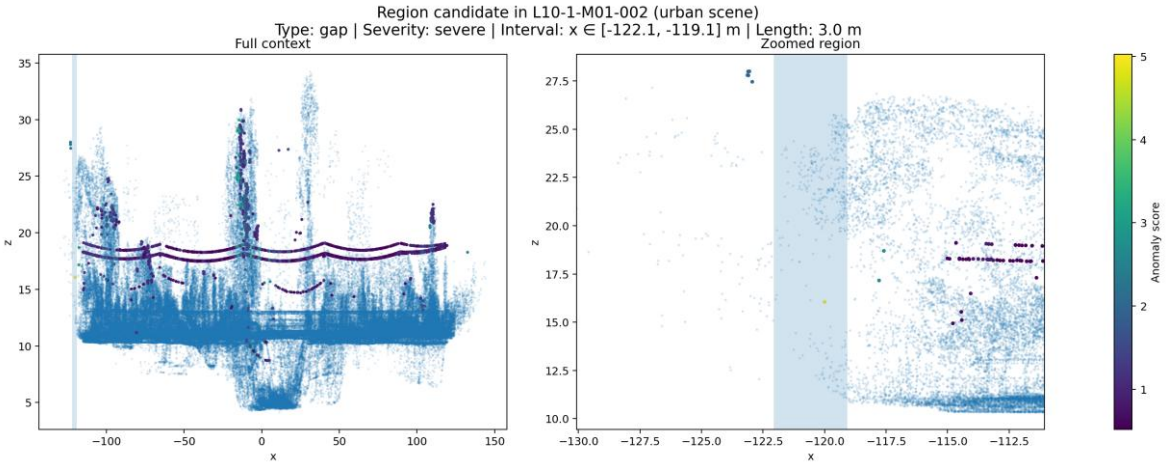

Figure 2: Ejemplo de una región urbana detectada tras la segmentación semántica y el postprocesado

La etapa de postprocesado demuestra además que la salida de la segmentación puede transformarse en información útil para inspección, al identificar intervalos anómalos asociados a discontinuidades, incertidumbre, baja densidad y cambios de altura.

5. Conclusiones

Este trabajo presenta un marco completo y reproducible para el análisis de nubes de puntos de catenaria ferroviaria basado en deep learning. Los resultados muestran que la segmentación semántica constituye una base válida para estructurar escenas ferroviarias complejas y aislar los elementos relevantes de la catenaria. El trabajo también demuestra que la ampliación de la segmentación mediante un postprocesado orientado a anomalías incrementa el valor práctico del sistema.

Como trabajo futuro, el marco puede mejorarse mediante una mejor generalización entre escenarios, arquitecturas más avanzadas y su validación sobre conjuntos de datos reales adicionales de inspección.

6. Referencias

[1] Qiu, B., Zhou, Y., Dai, L., Wang, B., Li, J., Dong, Z., Wen, C., Ma, Z., & Yang, B. (2024). *WHU-Railway3D: A diverse dataset and benchmark for railway point cloud semantic segmentation*. *IEEE Transactions on Intelligent Transportation Systems*.

[2] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). *PointNet: Deep learning on point sets for 3D classification and segmentation*. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[3] Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2020). *RandLA-Net: Efficient semantic segmentation of large-scale point clouds*. En *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

ANOMALY DETECTION IN CATENARY COMPONENTS OF RAILWAY SYSTEMS USING INSPECTION IMAGES AND DEEP LEARNING ALGORITHMS

Author: Martínez Ruiz, Lucía.
Supervisor: Sanz Bobi, Miguel Ángel.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This Master's Thesis develops a deep learning workflow for the semantic segmentation of railway catenary point clouds and the detection of anomalous regions. The proposed pipeline covers data preparation, model training, evaluation, and anomaly-oriented post-processing. Several architectures were studied, with RandLA-lite achieving the best balance between segmentation quality and computational efficiency. The final system supports both the classification of key catenary elements and the localization of suspicious regions for inspection tasks.

Keywords: Railway Point Clouds, Semantic Segmentation, Catenary, Anomaly Detection, Deep Learning, RandLA-Net, LiDAR.

1. Introduction

Railway infrastructure inspection increasingly relies on data-driven tools able to process large and complex sensor data. In this context, mobile LiDAR point clouds provide detailed 3D information about railway environments, including overhead catenary components such as contact wire, messenger wire, droppers, and supports. However, the irregular structure and large size of point clouds make manual analysis difficult and inefficient.

This thesis addresses that problem through deep learning-based semantic segmentation, with the additional goal of extending the framework toward anomaly detection. The work is based on the WHU-Railway3D dataset and focuses on building a reproducible workflow that can support future intelligent inspection systems [1].

2. Project Definition

The main objective of the project is to design and evaluate a complete pipeline for the automatic understanding of railway catenary point clouds. The work includes data preprocessing, train/validation/test organization, model training, quantitative evaluation, and result visualization.

In addition to semantic segmentation, the project proposes an anomaly-oriented analysis stage. Its purpose is to transform point-wise predictions into interpretable regions that may indicate gaps, uncertainty, low-density behavior, or abrupt geometric changes. This makes the final output more useful from an inspection and maintenance perspective.

3. Model / System / Tool Description

The proposed system is designed as a complete workflow for railway catenary point cloud analysis. The first stage focuses on data preparation, including dataset

organization, preprocessing, split generation, and the creation of derived data structures for training and evaluation.

The second stage consists of semantic segmentation. The system evaluates different deep learning architectures, including a baseline pointwise MLP, PointNet [2], PointNet++, RandLA-lite [3], and a Minkowski-based sparse convolutional model. Their performance is compared using class-aware evaluation metrics, with special attention to the difficulty of segmenting thin and imbalanced catenary elements. Among them, **RandLA-lite** emerges as the most suitable model in the main experiments, as it offers the best balance between segmentation quality and computational efficiency.

The final stage extends semantic segmentation toward anomaly detection. Instead of limiting the output to point-wise class predictions, the system aggregates the results into regions of interest and analyzes them through indicators such as uncertainty, low point density, geometric discontinuities, and height jumps. This post-processing step makes the results more interpretable and useful for inspection purposes, since it highlights suspicious intervals that may require further review. In Figure 3 it is shown the general workflow proposed.

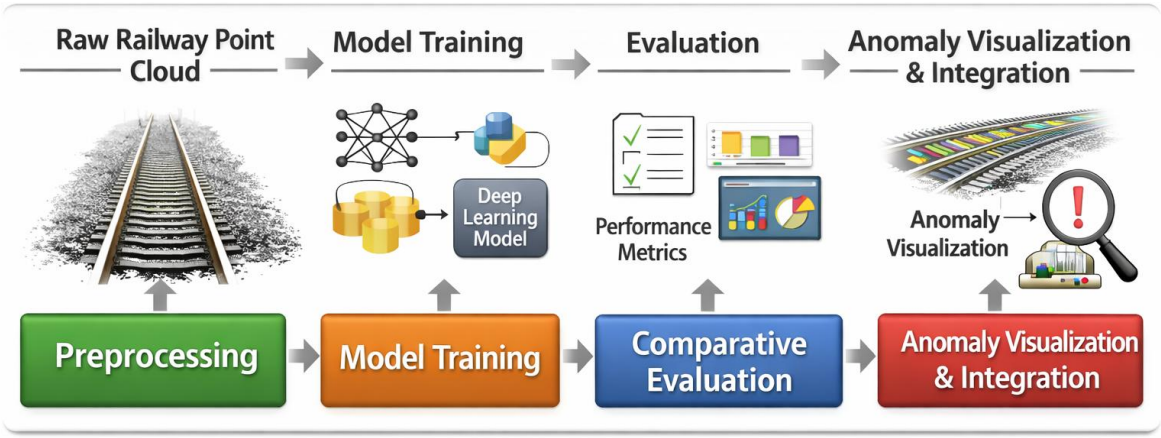

Figure 3: General workflow of the proposed system

4. Results

The results show that semantic segmentation provides an effective basis for the analysis of railway catenary point clouds. The comparison between the evaluated models confirms that the task is challenging due to class imbalance, scene variability, and the thin geometry of several catenary elements.

Among the tested models, the best performance in the main validation experiments is obtained with **RandLA-lite**, which reaches a mean ***mIoU_present50*** of approximately ***0.53*** while maintaining a more favorable computational cost than the heavier alternatives.

The following Figure 4 shows a full tile context together with a zoomed view of the detected interval, illustrating how the proposed workflow localizes suspicious regions in an interpretable way.

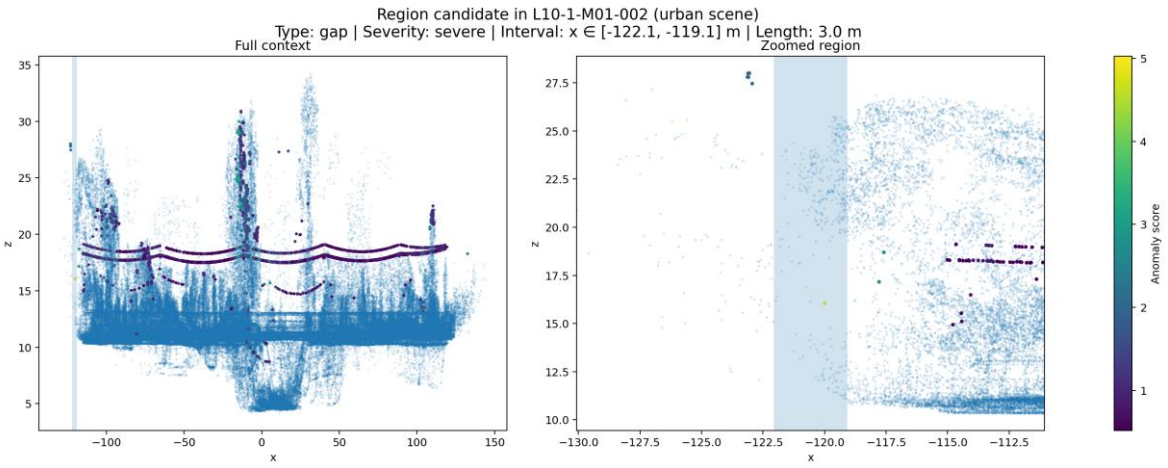

Figure 4: Example of an urban region detected after semantic segmentation and post-processing

The post-processing stage also shows that the segmentation output can be transformed into actionable inspection information. By combining prediction-based and geometric indicators, the system identifies anomalous intervals associated with gaps, uncertainty, low-density behavior, and height changes. The final visualizations highlight suspicious regions in a clear and interpretable way, which increases the practical value of the workflow for railway inspection support.

5. Conclusions

This thesis presents a complete and reproducible framework for railway catenary point cloud analysis based on deep learning. The results show that semantic segmentation is a valid basis for structuring complex railway scenes and isolating relevant catenary elements. The work also demonstrates that extending segmentation with anomaly-oriented post-processing increases the practical value of the system.

As for future work, the framework could be improved through better cross-scenario generalization, more advanced architectures, and validation on additional real inspection datasets.

6. References

[1] Qiu, B., Zhou, Y., Dai, L., Wang, B., Li, J., Dong, Z., Wen, C., Ma, Z., & Yang, B. (2024). *WHU-Railway3D: A diverse dataset and benchmark for railway point cloud semantic segmentation*. *IEEE Transactions on Intelligent Transportation Systems*.

[2] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). *PointNet: Deep learning on point sets for 3D classification and segmentation*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[3] Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2020). *RandLA-Net: Efficient semantic segmentation of large-scale point clouds*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Contents

List of figures

List of tables

Chapter 1. INTRODUCTION

Railway transport is one of the most demanding infrastructure domains from several perspectives such as safety, reliability, service continuity, and maintenance management. Unlike isolated engineered systems, railway networks operate as highly interconnected environments in which failures in one subsystem can quickly propagate into timetable disruption, increased operating costs, reduced punctuality, and, in extreme cases, safety-critical incidents. For that reason, railway infrastructure monitoring has progressively evolved from a mainly reactive and periodic activity into a data-intensive discipline that seeks to identify deterioration earlier, characterize asset condition more accurately, and support more informed maintenance decisions [8], [10].

Electrified railways present a particularly important and technically challenging case. Their performance depends not only on the condition of the track and civil infrastructure, but also on the correct operation of the traction power system and, more specifically, on the continuous interaction between the pantograph and the overhead contact infrastructure. This thesis is situated precisely at that intersection: it studies how modern point-cloud processing and deep learning can contribute to the monitoring of railway catenary environments and, ultimately, support anomaly-oriented inspection in electrified railway corridors [9], [11].

1.1 CONTEXT OF RAILWAY INFRASTRUCTURE MONITORING

Railway infrastructure monitoring has become a central issue in both research and industrial practice because railway systems must satisfy simultaneously high standards of safety, availability, operational efficiency, and cost control [8]. Traditional inspection strategies, although still indispensable, are increasingly limited when they are used alone. Large railway networks extend over long distances, including many interactive assets, and are exposed to environmental, mechanical, and operational stresses that evolve over time. As a result,

infrastructure managers increasingly require monitoring approaches capable of moving beyond fixed inspection intervals toward more condition-aware and data-driven maintenance strategies.

Among railway subsystems, the overhead contact system is especially relevant in electrified lines because it directly supports the energy transfer required for train operation. The overhead catenary system includes several interdependent elements, such as contact wires, messenger wires, droppers, poles or masts, and support devices [9], [11]. These components are structurally thin, spatially extended, and exposed to numerous degrading factors, including wind, ice, temperature fluctuations, and mechanical wear caused by repeated pantograph interaction. The following Figure 5 shows the different elements that exist in the catenary.

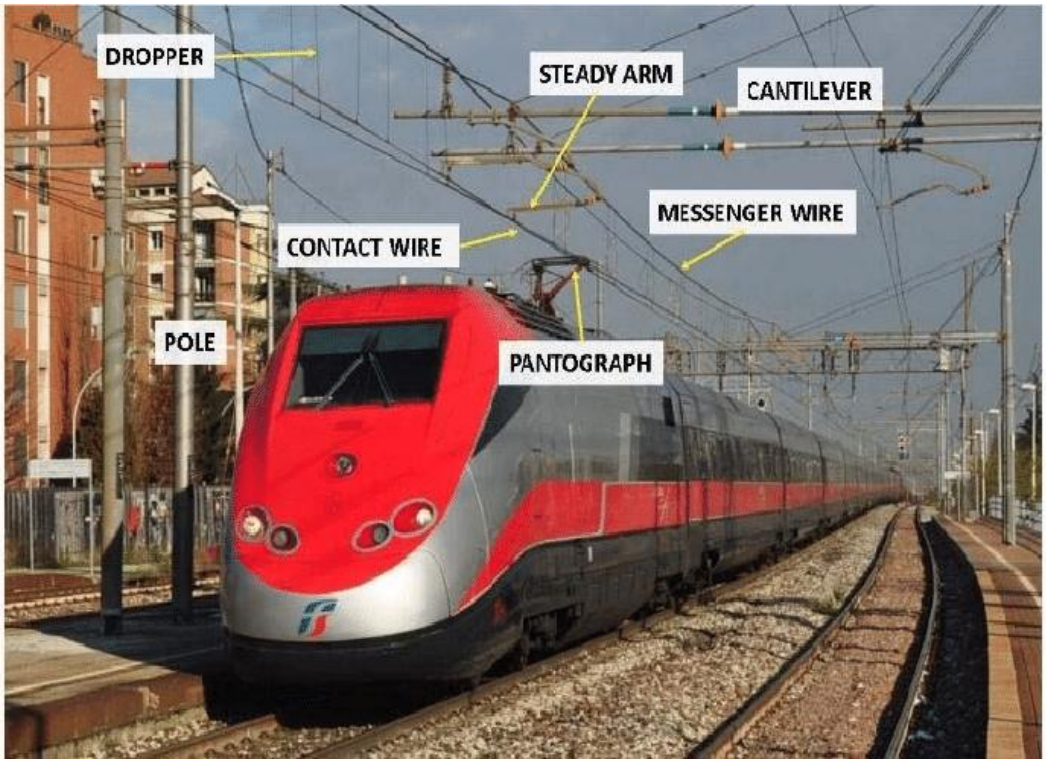

Figure 5: Overhead contact system [12]

As shown in Figure 5**¡Error! No se encuentra el origen de la referencia.**, the catenary system is not a single wire but a structured assembly of elements with distinct mechanical and functional roles. This is important from a data-analysis perspective because the thesis does not operate on a generic notion of "overhead infrastructure." Instead, it is concerned with identifying and separating semantically meaningful catenary-related elements that later become relevant for anomaly-oriented analysis [11].

Historically, a large proportion of railway overhead inspection has depended on manual procedures or on expert-assisted review of collected measurements. However, literature on OCS recognition makes it clear that manual methods suffer from low accuracy and efficiency, particularly as high-speed rail mileage and operating hours continue to increase [2], [11]. Additionally, traditional manual inspection is no longer adequate for modern operations and intelligent identification techniques are needed to improve effectiveness and safety. This shift in perspective is one of the main reasons why automated recognition from sensor data has become an active research topic in railway engineering.

This industrial need is also visible in real infrastructure monitoring practice. Network Rail's New Measurement Train, for example, monitors and records track condition information at speeds of up to 125 mph, helps identify faults before they become safety or performance issues, covers 115,000 miles per year, and captures approximately 10 TB of image data every 440 miles. The significance of this example lies in what it reveals about the state of the field: railway operators already possess sophisticated sensing platforms capable of collecting infrastructure data at scale [6]. The increasingly difficult part is therefore the interpretation of those data in a way that is fast, reliable, repeatable, and sufficiently detailed to support maintenance actions.

One of the sensing modalities that has become especially valuable in this context is LiDAR-based or mobile-mapping point-cloud acquisition. Point clouds provide a dense three-dimensional representation of railway corridors and capture the geometric arrangement of rails, masts, wiring, supports, vegetation, buildings, and surrounding terrain [4], [5]. This makes them particularly suitable for infrastructure analysis, because many railway assets are

better described by spatial structure than by texture alone. In the case of catenary systems, this is especially important: thin elongated wires and droppers may be difficult to interpret from conventional imagery under varying lighting conditions, whereas 3D data preserve their geometry more directly.

Nevertheless, raw point clouds are not yet actionable maintenance information. Their value emerges only when they can be transformed into semantically meaningful representations [4], [5]. This is the purpose of semantic segmentation: assigning a class label to each point so that the railway scene is decomposed into interpretable infrastructure categories [5]. In railway environments, this can include rails, masts, wiring, droppers, traffic lights, signals, and other surrounding elements. A representative example is provided by the work of Lamas et al. [5], who present a workflow for semantic segmentation of complex railway environments and explicitly classify relevant elements such as rails, masts, wiring, droppers, traffic lights, and signals over a large 90 km railway scenario. Therefore, this is the right moment to make the transition from physical infrastructure to data-driven infrastructure understanding.

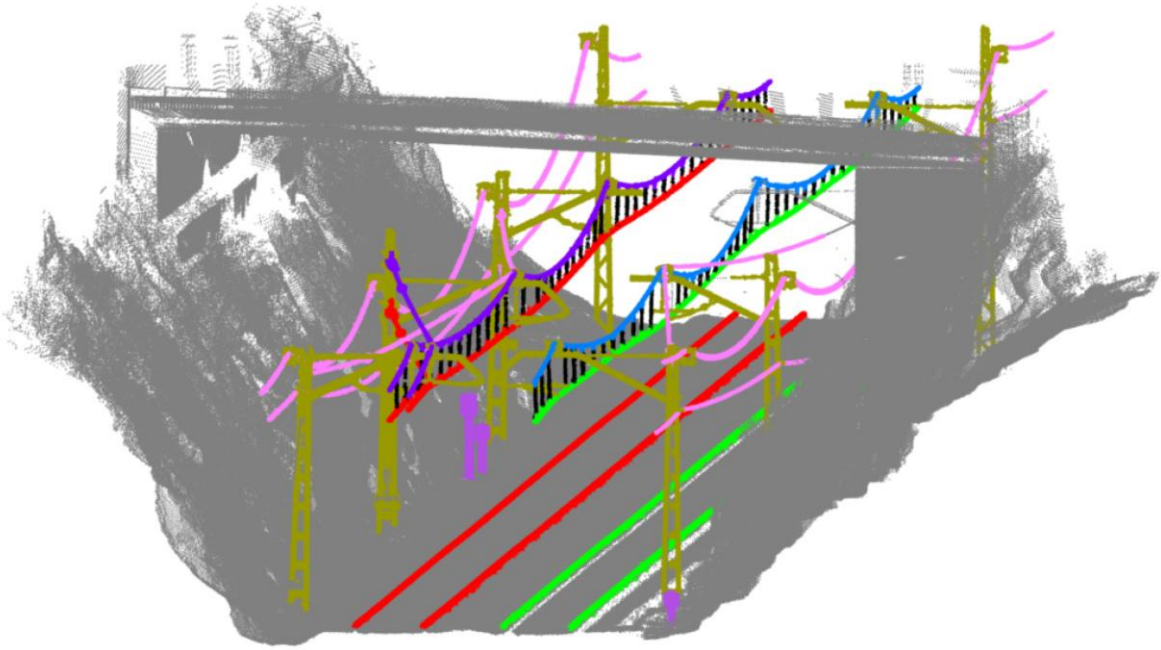

Figure 6. Railway point cloud after semantic segmentation [5]

¡Error! No se encuentra el origen de la referencia. Figure 6 illustrates this transition clearly. Instead of observing an unstructured cloud of points, it can now be seen a railway environment organized into semantic classes [5]. Once the scene has been segmented, the point cloud is no longer just a geometric record of the corridor; it becomes a machine-readable representation of railway assets. Such a representation is much more useful for downstream tasks such as component extraction, geometry checking, corridor inventory, digital railway modelling, and anomaly-oriented inspection.

Another important contextual factor is the recent emergence of large-scale railway point-cloud benchmarks. For many years, research in this area was constrained by the scarcity of railway-specific annotated datasets. More recently, however, datasets such as WHU-Railway3D [7] have significantly improved the experimental basis of the field. According to the project repository and associated publication, WHU-Railway3D spans approximately 30 km, contains about 4.6 billion annotated points, includes 11 semantic classes, and covers urban, rural, and plateau railway environments. The benchmark positions itself as a resource

for railway point-cloud semantic segmentation and for advancing digital-twin-oriented railway digitization.

The availability of such datasets is highly significant. It means that railway infrastructure monitoring can now be studied under conditions that are much closer to real deployment scenarios, with diverse scenes, specialized railway classes, and large-scale data volumes [3], [7]. This makes it possible not only to test whether a model works in principle, but also to compare architectures, analyze class-specific behavior, and investigate whether segmentation quality is sufficient for downstream engineering tasks such as anomaly detection.

1.2 PROBLEM STATEMENT AND SCOPE

Despite the progress described above, the automated understanding of railway catenary environments remains a difficult problem. Overhead contact systems are geometrically complex and contain many thin, elongated, sparsely represented, and partially occluded elements [2], [11]. Their components are embedded within broader railway corridors that also contain terrain, vegetation, buildings, fences, poles, rails, and other infrastructure. As a result, the point-cloud segmentation problem is characterized by strong class imbalance, structural delicacy, and the need to preserve fine geometric detail. OCS segmentation states that identifying and detecting overhead catenary components is challenging because of the variety of OCS elements and because recent LiDAR-based methods still need to improve local feature extraction, contextual feature integration, and feature enhancement while maintaining low latency and reduced model complexity.

This means that the research problem addressed is more demanding than generic point classification. The central question is not simply whether points can be labeled, but whether railway catenary-related elements can be segmented with enough robustness and structural fidelity to support meaningful downstream inspection [11]. In a safety-relevant domain such

as railway electrification, segmentation errors are not merely numerical inaccuracies. If a dropper is missed, if an overhead line is only partially captured, or if support structures are confused with background clutter, any subsequent anomaly analysis becomes less trustworthy. Semantic segmentation must therefore be treated as a critical upstream stage in a larger monitoring pipeline rather than as an end.

The thesis addresses this challenge from a 3D point-cloud perspective. Its main premise is that reliable anomaly-oriented analysis in railway catenary systems requires a reliable prior separation of the relevant infrastructure components from the rest of the scene. Consequently, the work is organized as an end-to-end pipeline that starts with railway point-cloud acquisition and preprocessing, continues through the training and evaluation of semantic segmentation models, and culminates in a downstream stage focused on anomaly-related analysis over catenary-relevant classes. Rather than treating segmentation and anomaly detection as unrelated topics, the thesis frames them as consecutive stages of the same engineering problem: transforming raw 3D railway data into structured maintenance intelligence.

Chapter 2. TECHNOLOGIES BACKGROUND

This chapter introduces the main technologies and technical concepts required to understand the rest of the thesis. Its purpose is to clarify the data representation, processing strategies, and deep-learning paradigms that appear repeatedly throughout the project. The technical core of the thesis is the semantic segmentation of railway mobile LiDAR point clouds, with special attention to overhead-contact-line-related elements and their later use in anomaly-oriented analysis [2], [7], [11], [14].

2.1 MOBILE LIDAR POINT CLOUDS

A point cloud is a set of 3D samples that describe the geometry of a scene through spatial coordinates, typically expressed as (x, y, z). In mobile railway mapping, these points are acquired by LiDAR sensors mounted on inspection platforms or measurement vehicles, which emit laser pulses and estimate the position of observed surfaces from the returned signal. In addition to coordinates, railway point clouds may also include attributes such as reflected intensity, scanning angle, number of returns, or semantic labels when annotation is available [2], [7], [14].

Point clouds are the native data representation used in this thesis because they preserve the three-dimensional structure of the railway corridor. This makes them especially suitable for railway scenes, where many relevant elements are elongated, thin, and spatially structured. Overhead contact line related elements are often difficult to characterize reliably from images alone, while point clouds retain explicit geometric information that is directly useful for segmentation and later structural analysis [11].

A representative example of this type of data is shown in Figure 7, which illustrates railway point-cloud scenes from the WHU-Railway3D benchmark. This figure expresses the kind of 3D input handled in the thesis [7].

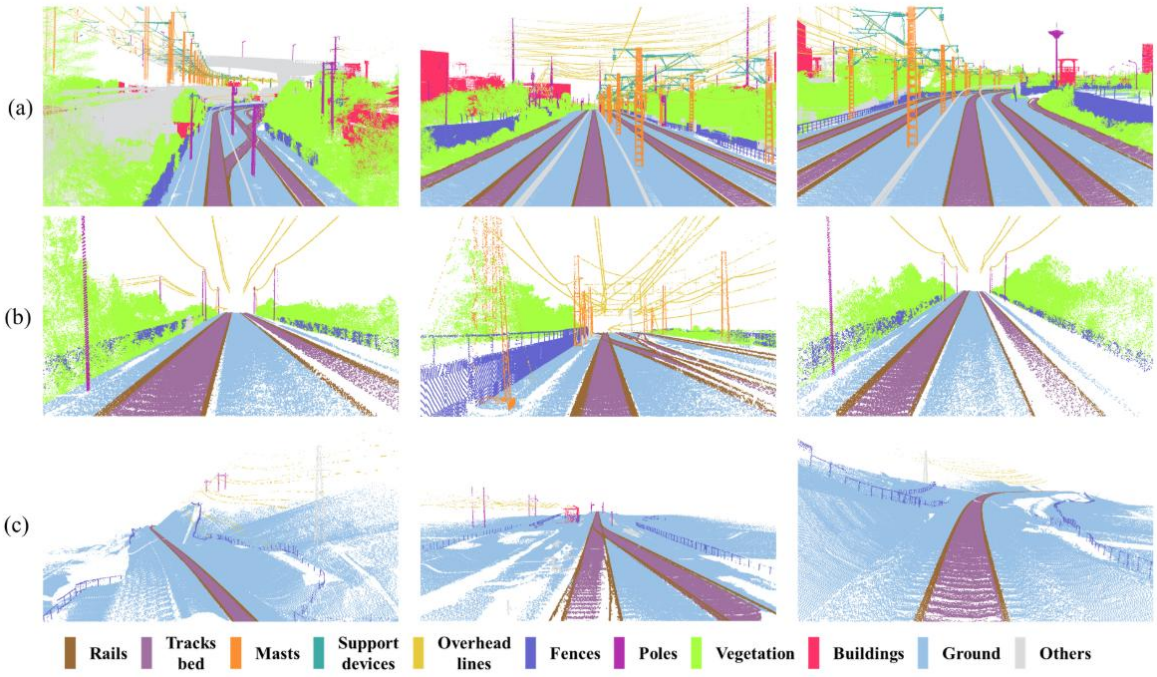

Figure 7: Representative railway point-cloud scenes from the WHU-Railway3D benchmark [7]

2.2 POINT-CLOUD SEMANTIC SEGMENTATION

The central prediction task addressed is semantic segmentation. In semantic segmentation, the objective is to assign a class label to every individual point in the cloud so that the raw railway scene becomes a structured representation in which each point belongs to a semantic category such as ground, rail, mast, vegetation, overhead line, or support device [4], [5], [7], [11].

This differs from global classification, which predicts a single label for the whole input, and from object detection, which typically predicts regions or bounding boxes. Semantic segmentation is especially appropriate for railway point clouds because railway

infrastructure is spatially continuous, strongly interleaved, and often composed of elongated objects that are not well represented by box-based outputs [4], [7].

2.3 *POINT-CLOUD ATTRIBUTES, TILES, AND FILE REPRESENTATION*

Railway point clouds are geometric objects, and furthermore, they are structured data containers. In practice, each point may include coordinates, intensity, and, when available, semantic class information. The WHU-Railway3D [7] benchmark includes 3D coordinates together with additional attributes such as reflected intensity, scanning angle, and number of returns, which enrich the information available to the learning algorithms.

For computational reasons, large railway point clouds are commonly divided into smaller spatial units or **tiles**, which can then be processed independently during training and evaluation. This tiling strategy is particularly important in corridor-scale scenes, where a full acquisition may contain millions or even billions of points. In practical workflows, the data are often stored in formats such as **PLY**, which allow coordinates and per-point properties to be preserved in a consistent way. The relevance of such file structures is mainly practical, because they enable the processing pipeline to read geometry, attributes, and labels in a unified representation.

2.4 *VOXELIZATION AND SPARSE TENSORS*

One of the main computational difficulties of point clouds is that they are irregular and potentially very large. Processing raw points directly at full density can be inefficient in both time and memory. A common strategy to alleviate this problem is **voxelization**, in which the 3D space is divided into a regular grid and the points falling inside each occupied cell are grouped together, so that they can be processed more efficiently [7], [15].

However, after voxelization, most cells in the 3D grid remain empty. A dense volumetric representation would therefore be computationally wasteful. This leads to the concept of **sparse tensors**, in which only occupied coordinates and their associated features are stored and processed. Sparse tensors are the technological basis of sparse-convolution frameworks such as the Minkowski approach [15], which make large-scale 3D semantic segmentation feasible without spending resources on empty space.

2.5 DEEP LEARNING ARCHITECTURES

Point clouds are unordered and irregularly sampled, which means that standard 2D convolutional neural networks cannot be applied directly without adaptation. For that reason, the field has developed specialized neural architectures capable of learning from point sets or sparse 3D structures. In this thesis, the main model families effectively used in the experiments are a per-point MLP baseline, PointNet, PointNet++, RandLA-Net, and sparse-convolution-based Minkowski networks. Together, these models represent a progression from simple baselines to more expressive and computationally specialized architectures [15], [16], [22], [29].

2.5.1 BASELINE PER-POINT MLP

A simple baseline used in this thesis is a **per-point multilayer perceptron (MLP)**, which classifies each point independently from its input attributes. In this type of model, each point is processed as an individual feature vector, typically containing geometric coordinates and optional additional attributes such as intensity. The model does not explicitly model complex spatial neighborhoods or structured local geometry, which makes it less expressive than more advanced point-cloud architectures.

Even so, this baseline remains useful for two reasons. First, it provides a lightweight reference against which more advanced architectures can be compared. Second, it helps quantify how much performance improvement is obtained when local structure, hierarchical context, or sparse spatial processing are incorporated.

2.5.2 POINTNET

PointNet is one of the earliest deep-learning architectures specifically designed for point clouds. Its key contribution is that it processes unordered point sets directly and aggregates information through a symmetric pooling operation, which makes the output invariant to the order of the input points [29]. This was an important milestone in 3D deep learning because it showed that point clouds could be processed natively without converting them into images or dense voxels.

In practical terms, PointNet learns point-wise features and then combines them into a global representation. This makes it effective for basic point-cloud classification and segmentation tasks, but it also introduces a limitation: the model has only a limited ability to capture fine local geometric structure. This is especially relevant in railway scenes, where many classes differ through subtle local arrangements and thin spatial patterns.

2.5.3 POINTNET++

PointNet++ extends PointNet by introducing hierarchical feature learning. Instead of relying mainly on global aggregation, it progressively samples local neighborhoods, groups nearby points, and extracts features at increasing spatial scales [22]. This makes it much better suited than basic PointNet for learning local geometry and multiscale spatial context.

This hierarchical principle is particularly relevant for railway environments, where the model must preserve fine geometric details while still understanding broader structural context. Thin elements, supports, rails, and nearby clutter may share some low-level

properties, so capturing relationships at multiple scales becomes essential. Figure 8 illustrates the general logic of PointNet++, which makes it a useful conceptual reference for point-based segmentation.

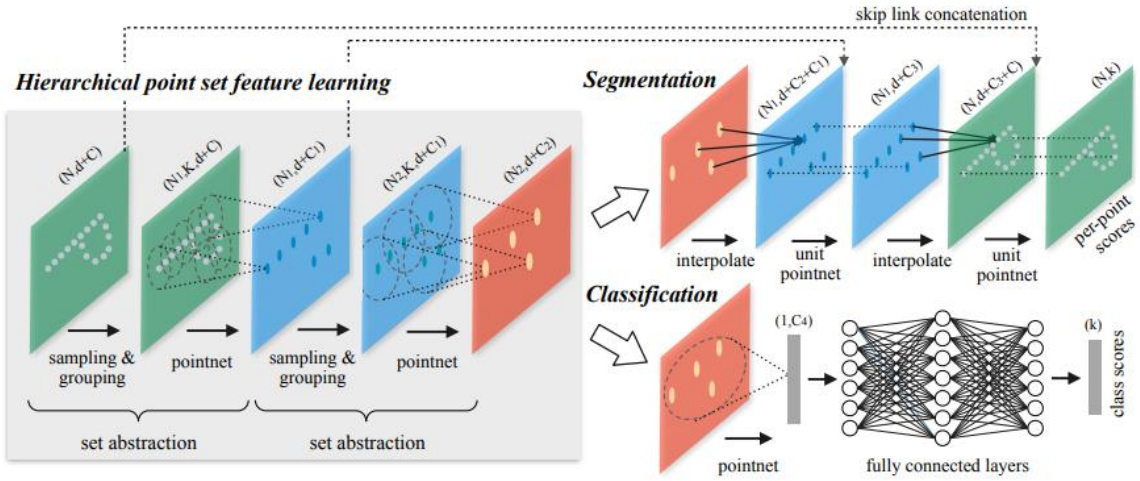

Figure 8: PointNet++ hierarchical feature-learning architecture for classification and segmentation [22]

2.5.4 RANDLA-NET

RandLA-Net was designed for efficient semantic segmentation of large-scale point clouds [16]. Its main contribution is the combination of lightweight random sampling with local feature aggregation, which allows the network to handle very large point sets without the high computational burden of heavier alternatives.

This efficiency is particularly relevant in railway applications because corridor-scale LiDAR scenes can be extremely large. In that context, computational scalability is not merely a secondary implementation issue, but an important part of the technological design. RandLA-Net is therefore relevant to this thesis both as a segmentation model and as an example of how efficiency considerations shape point-cloud deep learning in large outdoor scenes. As shown in Figure 9, the architecture progressively downsamples the input while preserving informative local features through dedicated aggregation modules.

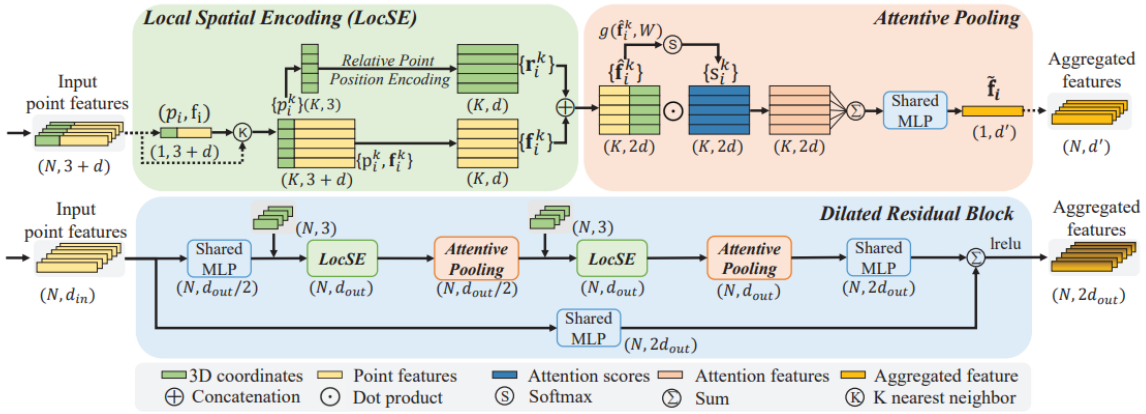

Figure 9. RandLA-Net overview showing progressive downsampling and local feature aggregation for large-scale point-cloud segmentation [16]

2.5.5 MINKOWSKI CNN

Minkowski CNN (Convolutional Neural Networks) operates on sparse tensors instead of dense grids [15]. Their technological importance lies in the fact that they extend the convolutional paradigm to sparse high-dimensional data, which is exactly the situation encountered after voxelizing large LiDAR point clouds.

The Minkowski approach is particularly suitable for large 3D scenes in which voxelization is used as a regularization and downsampling strategy. As shown in Figure 10, this technology is especially relevant because it connects directly with the sparse representation of voxelized railway point clouds and provides a powerful framework for large-scale semantic segmentation.

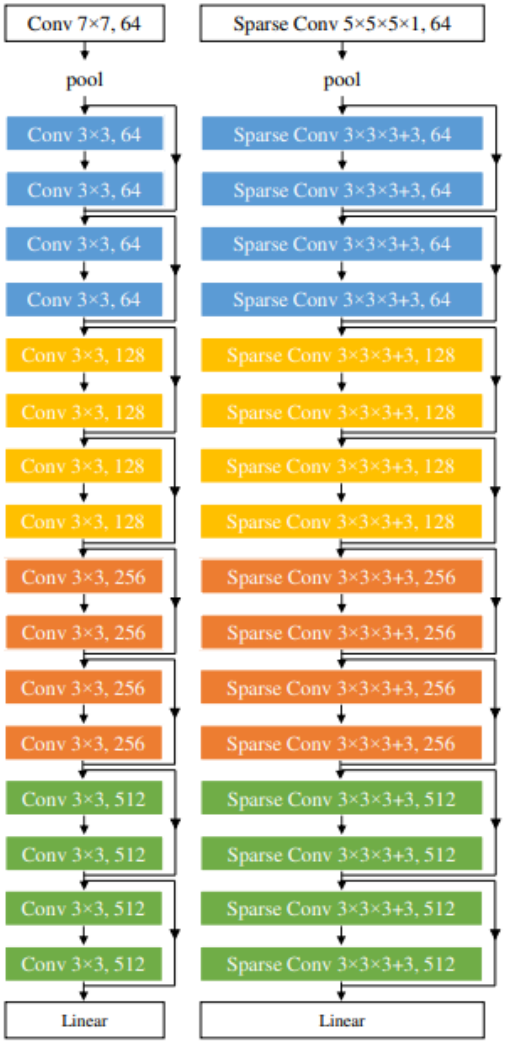

Figure 10: Structural comparison between ResNet18 and MinkowskiNet18, highlighting sparse convolutional processing [15]

2.6 TECHNOLOGICAL BASIS FOR ANOMALY DETECTION

Although the core of this thesis is semantic segmentation, the project is also connected to anomaly detection as a natural downstream step. Once the relevant railway components or catenary-related points have been identified, the next question is whether the observed structure corresponds to normality or to a potentially defective condition. In the literature, this transition from semantic understanding to condition assessment is often approached through reconstruction-based or one-class strategies, especially when defective examples are scarce [1], [21], [26].

Taking together the technologies described above, they form the technical backbone of the thesis. Mobile LiDAR provides the 3D sensing modality, point clouds provide the native data representation, semantic segmentation defines the learning objective, voxelization and sparse tensors make large-scale processing feasible, and the sequence from MLP baseline to PointNet, PointNet++, RandLA-Net, and Minkowski-based models reflects the progression from simple point-wise classification to more expressive and scalable 3D learning paradigms.

Chapter 3. STATE OF THE ART

Railway catenary systems, also referred to as overhead contact line (OCL) systems, are critical assets in electrified railways because they directly affect energy supply continuity, operational reliability, punctuality, and safety. In recent years, their inspection has evolved from periodic, mostly manual procedures to digital, sensor-based, and increasingly intelligent workflows. This transition is part of the broader digitalization of railway maintenance and the growing adoption of AI-enabled predictive maintenance strategies across infrastructure monitoring. In that context, catenary inspection has become an important application domain for computer vision, deep learning, LiDAR processing, and multimodal sensing [1], [8], [9], [10].

Traditionally, catenary inspection relied on human expertise, dedicated measurement campaigns, and specialized vehicles. A representative real-world example is Network Rail's New Measurement Train, which shows how infrastructure managers have progressively incorporated measurement and monitoring technologies into maintenance workflows [6]. However, conventional inspection still presents several limitations: it is expensive, time-consuming, difficult to scale, and strongly dependent on expert interpretation. Moreover, many catenary components are thin, elongated, partially occluded, and embedded in complex outdoor scenes, which makes them especially challenging to inspect under varying illumination, weather, motion, and background conditions [1], [9].

A useful way to understand the technical difficulty of the problem is to look first at the diversity and spatial arrangement of the components that make up a catenary support system. As shown in Figure 11, the inspection target is not a single object but a set of heterogeneous elements, including: 1) Insulator base; 2) Insulator; 3) Brace sleeve; 4) Brace sleeve screw; 5) Rotary double-ear; 6) Binaural sleeve; 7) Isoelectric line; 8) Steady arm base; 9) Bracing wire hook; 10) Double sleeve connector; 11) Messenger wire base; 12) Windproof wire ring. This heterogeneity explains why modern inspection systems increasingly move away from

simple single-object detection and toward multi-component architectures able to cope with strong variation in scale, orientation, and visual appearance [20].

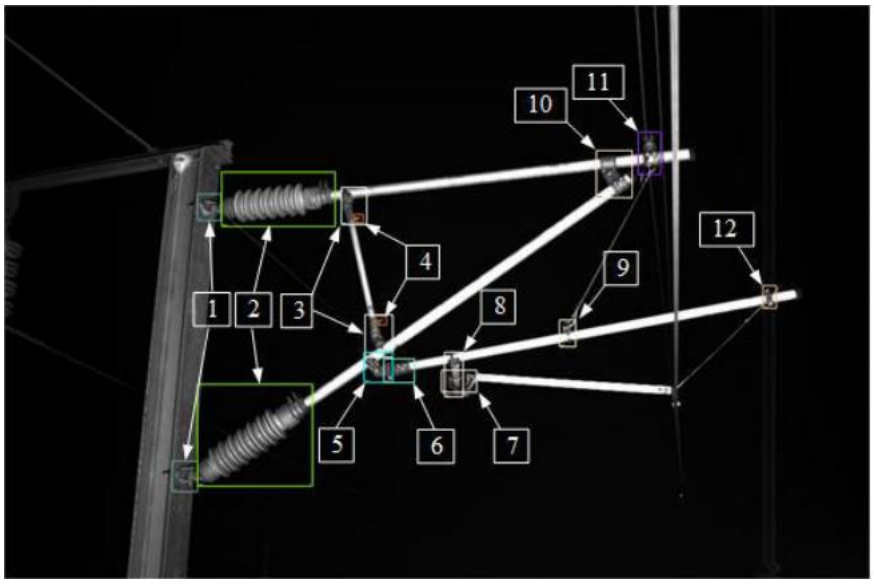

Figure 11: Main catenary support components typically targeted by automatic inspection systems [20]

From a methodological perspective, the state of the art can be divided into two major research directions. The first focuses on **component recognition and semantic understanding**, that is, identifying wires, supports, insulators, fasteners, and related railway elements. The second focuses on **condition assessment and anomaly detection**, where the aim is to determine whether a given component is damaged, loose, missing, broken, or otherwise abnormal. In recent works, both objectives are addressed sequentially: the system first localizes or segments the component and then evaluates its condition using supervised classification, reconstruction error, uncertainty estimation, or geometry-based reasoning [17], [20], [21], [26], [27].

3.1 DATASETS AND BENCHMARKS

Data availability remains one of the main bottlenecks in railway AI research. For many years, most catenary-inspection studies were developed on proprietary datasets collected by individual railway operators, which limited reproducibility and made fair comparison between methods difficult. The emergence of public datasets has improved this situation considerably, but the ecosystem is still fragmented across sensing modalities, tasks, and annotation levels [1], [8], [9].

In the 2D domain, **RailSem19** was a major milestone because it introduced the first public dataset for semantic rail scene understanding, with 8,500 annotated short sequences from the ego perspective of trains and trams [25]. This dataset is highly relevant because it widened railway perception beyond isolated defect studies and established a benchmark for rail-scene understanding under realistic operating conditions. As illustrated in the following Figure 12, RailSem19 captures a wide variety of railway-relevant classes and scene contexts, which makes it especially useful for robust visual learning in the rail domain rather than in generic road-driving scenarios.

Figure 12: Examples of annotated railway scene classes from RailSem19 [25]

For railway anomaly-related tasks, more specialized public datasets have also begun to appear. **RailFOD23**, for example, focuses on foreign-object detection on railroad transmission lines and expands the availability of public data for overhead-line safety

monitoring [28]. In parallel, the field is also moving toward **multimodal railway perception**, as shown by **OSDaR23**, which combines RGB, infrared, LiDAR, radar, and localization sensors in a railway environment [23]. This trend reflects that robust railway monitoring increasingly depends on the combination of complementary sensing streams rather than on a single visual modality.

In the 3D domain, the most relevant benchmark for this thesis is **WHU-Railway3D**, which establishes railway point-cloud semantic segmentation as a benchmark problem. The dataset spans approximately 30 km, contains around 4.6 billion points, and covers urban, rural, and plateau railway environments with 11 semantic classes [7]. This is especially important for catenary analysis because thin overhead structures are difficult to preserve after subsampling and are often heavily imbalanced relative to dominant classes such as ground, vegetation, or large structural elements. The value of WHU-Railway3D lies precisely in exposing that real complexity. As shown in Figure 13, the dataset includes markedly different railway contexts, making it suitable for evaluating how segmentation models behave across scene diversity rather than on a single homogeneous corridor.

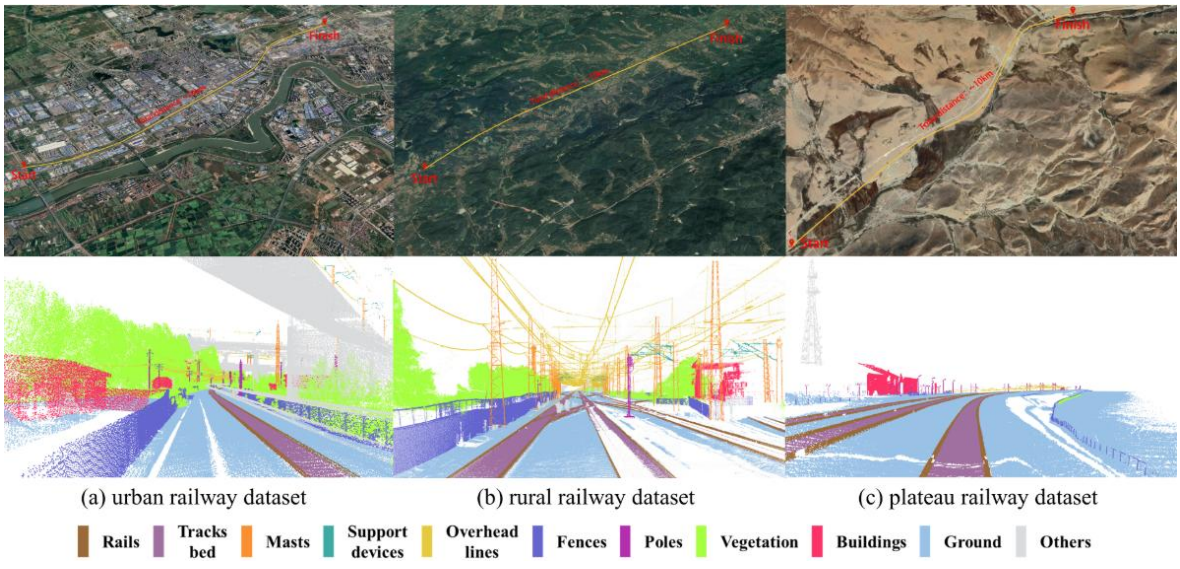

Figure 13: Examples of WHU-Railway3D scenes in urban, rural, and plateau environments [7]

Beyond WHU-Railway3D, the 3D benchmark ecosystem is also expanding with resources such as **SemanticRail3D** and other multi-context railway point-cloud datasets [3], [18]. This trend is especially relevant for current research because it supports reproducibility, cross-scene evaluation, and more realistic testing of models under infrastructure diversity.

3.2 DEEP LEARNING FOR CATENARY COMPONENT DETECTION IN IMAGES

The first wave of deep-learning-based catenary inspection research focused mainly on image-based recognition of specific support components and defect categories. Early work shows that convolutional neural networks can outperform traditional handcrafted approaches in defect detection tasks involving catenary fasteners and related support elements. A representative example is the work of Chen et al. [13], who demonstrated that deep CNNs could effectively identify fastener defects under realistic railway inspection conditions.

Subsequent studies moved from isolated defect classifiers toward more structured multi-stage architectures that explicitly separate localization, feature extraction, and final diagnosis. An influential example is the framework proposed by Kang et al. [17], which combines component localization, Bayesian segmentation, and prior geometric knowledge for contact wire detection.

A major advance was the attempt to detect all catenary support components under a unified architecture. Liu et al. [20] propose the **CSCD-Architecture**, which is specifically designed to manage components at different scales through a staged pipeline involving feature extraction, region proposal, and classification/regression subnetworks. As shown in Figure 14, this architecture is not a simple single-step detector but a hierarchical system that explicitly addresses the coexistence of large and small catenary support components.

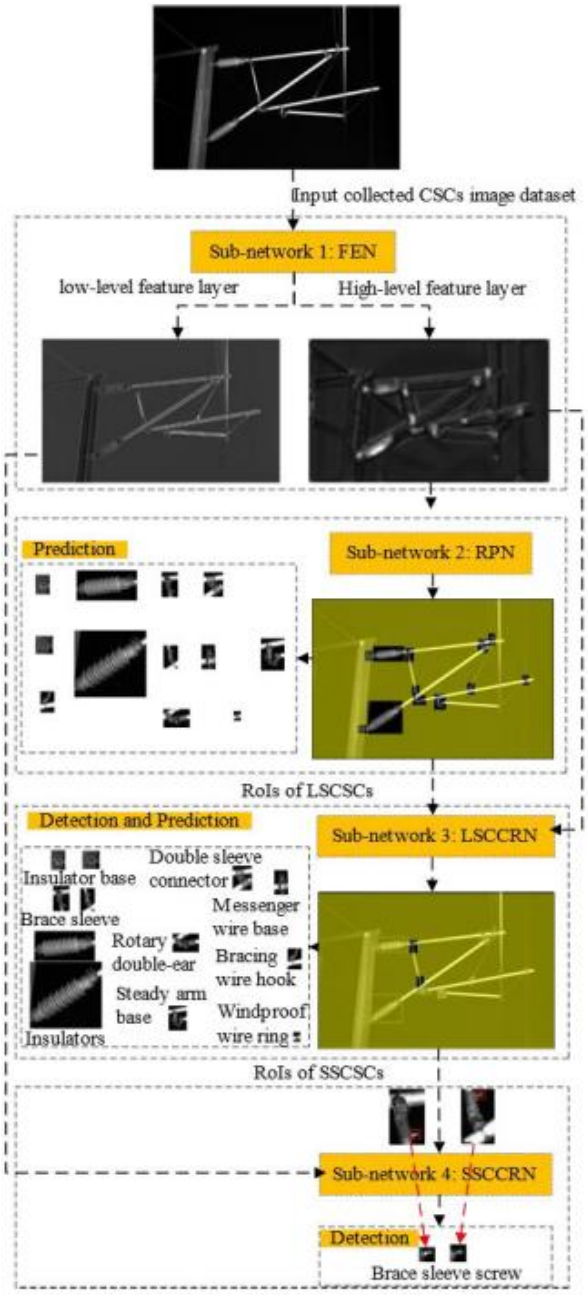

Figure 14: Overview of the CSCD architecture at different scales [20]

The same paper also makes clear why conventional general-purpose detectors are insufficient for railway catenary scenes. In fact, the structural comparison between Faster R-CNN and the modified CSCD architecture shows how the network was redesigned to better preserve relevant feature maps and improve the detection of both large-scale and small-scale catenary support components. For that reason, Figure 15 reinforces the idea that railway inspection often requires task-specific architectural adaptation rather than direct reuse of detectors.

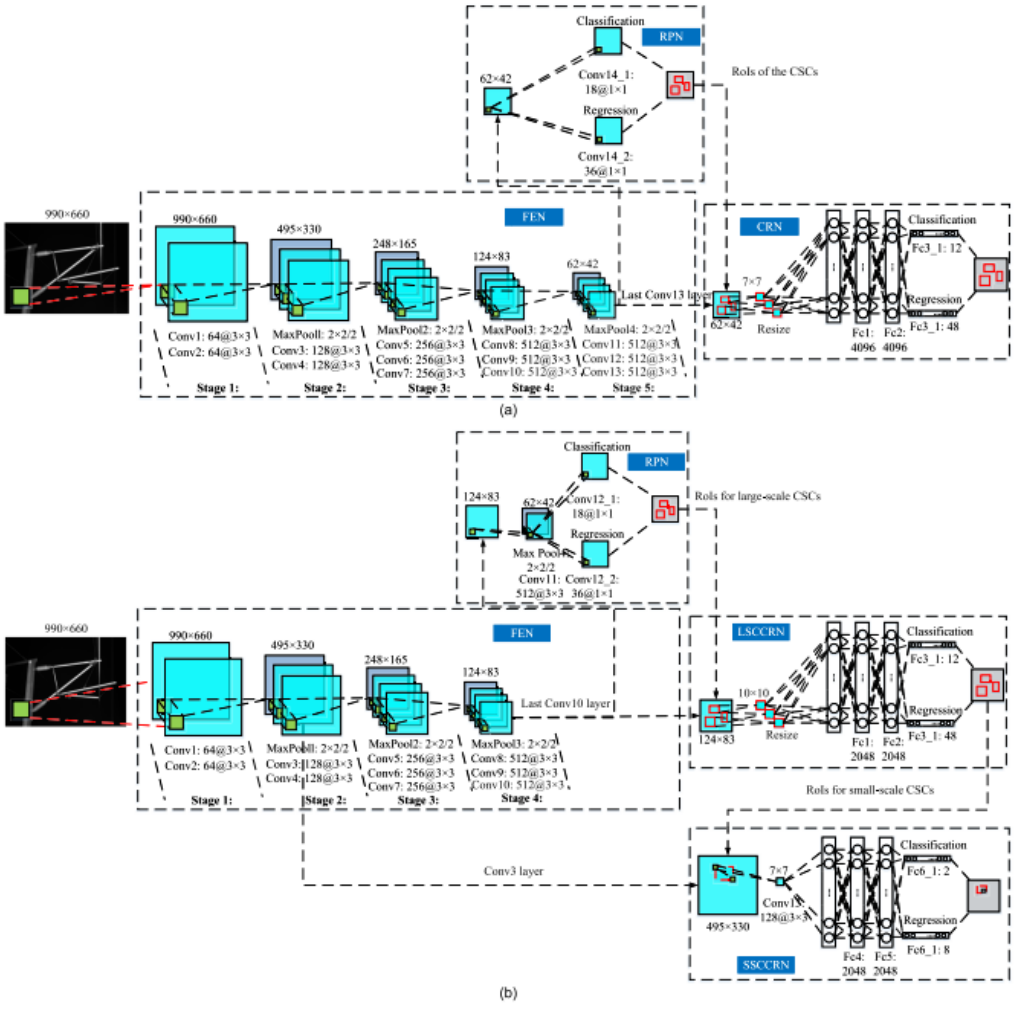

Figure 15: Structural comparison between Faster R-CNN and the CSCD architecture [20]

The literature also includes highly specialized studies devoted to component families and defect modes. Dropper fault detection, for instance, has been addressed through an efficient CNN architecture designed for high-speed inspection conditions [19]. Fastener looseness has been studied through refined localization and reinforcement-learning-assisted diagnosis [27]. Likewise, catenary insulator defect detection has been addressed through tighter oriented localization followed by adversarial reconstruction-based analysis [26]. This direction bridges supervised localization and unsupervised anomaly detection in a single pipeline. As shown in Figure 16, the method first localizes the insulator tightly and then performs defect assessment by reconstruction-based comparison, which is especially suitable when defect samples are scarce.

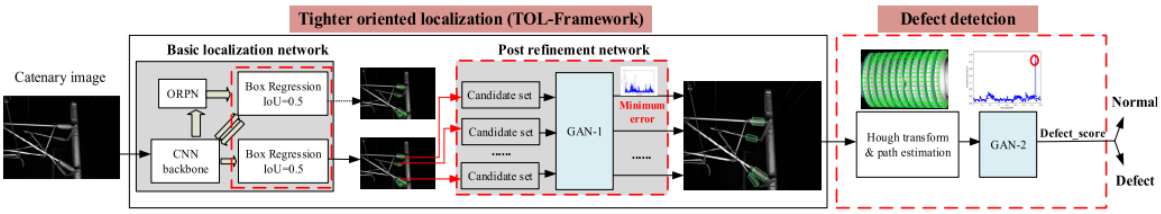

Figure 16: Overview of the insulator defect detection method [26]

Taken together, these studies show that image-based catenary inspection has reached a high degree of specialization and technical sophistication. At the same time, this specialization also reveals one of the field's limitations: performance is often strong when the target component and defect type are narrowly defined, but system-wide generalization remains more difficult.

3.3 3D POINT-CLOUD SEGMENTATION IN RAILWAY ENVIRONMENTS

While image-based methods dominate much of the defect-inspection literature, point-cloud analysis has become increasingly important for railway infrastructure modeling and monitoring. Point clouds provide explicit 3D geometry, which is particularly valuable in railway scenes because many relevant objects are spatially structured, elongated, and thin. In catenary systems, geometry is often as important as appearance, since wires, supports, and overhead arrangements must be understood not only as visual patterns but also as spatial configurations [2], [11], [14].

Early railway point-cloud studies focused on classification and segmentation using geometry-driven or machine-learning-based methods. Chen et al. [2] addressed overhead contact system component recognition using mobile 2D LiDAR, while Chen et al. later explored point-cloud classification for railway overhead contact systems [14]. More recent work by Lamas et al. [5], Grandio et al. [4], and Tu et al. [11] moves the field toward deep-learning-based semantic segmentation of complex railway environments and overhead contact systems specifically.

This railway-specific evolution is closely connected to the broader development of 3D deep learning. **PointNet++** is foundational because it introduces hierarchical feature learning on point sets, progressively aggregating local neighborhoods at increasing contextual scales [22]. This principle is particularly relevant for railway scenes, where local geometric cues need to be preserved without losing broader structural context.

After PointNet++, later architectures improve either geometric expressiveness or computational efficiency. **KPConv** introduced deformable convolutions directly on point clouds [24], while **RandLA-Net** focused on efficient semantic segmentation of very large-scale point clouds through random sampling and local feature aggregation [16]. This is especially relevant for railway infrastructure, where corridor-scale data can easily become prohibitively large.

Sparse-convolution approaches based on the **Minkowski** framework are also highly influential because they provide an efficient way to process voxelized sparse 3D tensors at scale [15]. Together with point-based networks such as PointNet++ and RandLA-Net, these architectures now form the core backbone of modern 3D semantic segmentation research and are directly relevant to railway benchmarks such as WHU-Railway3D and SemanticRail3D.

3.4 ANOMALY-DETECTION STRATEGIES

Although supervised component recognition has progressed rapidly, anomaly detection remains one of the most difficult problems in catenary inspection. Defect samples are usually rare, heterogeneous, and often insufficiently labeled. In real maintenance practice, large volumes of normal data are available, but anomalous cases may be scarce or unique. This has motivated the use of one-class, unsupervised, and reconstruction-based strategies that learn normality rather than relying on large defect datasets [1], [9], [21].

A representative example is the use of **generative adversarial networks** for learning the appearance of normal catenary support components and then flagging deviations as potential anomalies. Lyu et al. [21] proposed such a framework for generic anomaly detection in catenary support components. These methods are attractive because they reduce dependence on extensive annotated fault libraries and offer a realistic path for deployment in maintenance settings where abnormal samples are limited [26].

Another trend is the combination of deep learning with geometric and physical priors. Kang et al. [17] integrated prior geometric knowledge into Bayesian segmentation, and related work has used refined localization to support looseness diagnosis in fasteners [27]. Such hybrid approaches reduce false positives and produce decisions that are easier for infrastructure managers to interpret.

3.5 EVALUATION AND CURRENT TRENDS

The evaluation criteria used in the literature depend on the task. For image-based detection and defect classification, studies usually report precision, recall, F1-score, and mean Average Precision. For semantic segmentation, the most common metrics are class-wise IoU, mean IoU, and overall accuracy. In anomaly-detection settings, researchers also consider ROC curves, AUC, reconstruction-error distributions, and qualitative visual inspection of the results [7], [17], [20], [21], [26].

Several broad trends emerge from studies. First, specialized deep-learning pipelines now clearly outperform earlier handcrafted approaches in targeted catenary-inspection tasks [13], [17], [19]. Second, hybrid systems that combine learning with localization refinement, geometric priors, or reconstruction-based scoring tend to be more robust and more interpretable [17], [26], [27]. Third, in 3D railway segmentation, point-based and sparse-convolution architectures have significantly improved performance, but fine overhead elements remain difficult because they are geometrically thin, minority-class in nature, and sensitive to sampling strategy [7], [11], [15], [16], [22], [24].

Another decisive tendency is the increasing importance of **deployability**. Railway inspection systems are only valuable if they can be integrated into real sensing pipelines, inspection vehicles, or scalable offline workflows. Efficient designs such as RandLA-Net, benchmark-oriented datasets such as WHU-Railway3D, and multimodal resources such as OSDaR23 all reflect that the field is moving toward systems that are not only accurate, but also operationally realistic [7], [16], [23].

3.6 RESEARCH GAP

The studies reviewed above show that railway catenary inspection has advanced substantially in both image-based defect recognition and 3D railway scene understanding. However, important gaps remain. Much of the image-based literature is highly specialized around specific components or defect types, which limits generalization at full-system scale [19], [20], [26]. In the 3D domain, public benchmark culture is improving quickly, but fine overhead elements remain difficult to segment consistently, especially in large and heterogeneous railway environments [3], [7], [11]. Finally, anomaly detection built on top of 3D semantic understanding remains much less mature than supervised component detection or semantic segmentation [1], [7], [21], [26].

This gap is directly relevant to the present thesis. On the one hand, the state of the art already provides strong foundations for railway point-cloud semantic segmentation through architectures such as PointNet++, RandLA-Net, and sparse-convolution frameworks [15], [16], [22], [24]. On the other hand, there is still a clear need for pipelines that connect robust 3D semantic segmentation of catenary-related elements with anomaly-oriented analysis capable of supporting scalable, data-driven railway maintenance. This combination remains both technically challenging and scientifically valuable, which is precisely why it constitutes an appropriate research direction for this thesis [1], [7], [8], [9].

Chapter 4. PROJECT DEFINITION

After reviewing the technological background and the current state of the art, the project must now be defined in precise engineering terms. While the previous chapters established why railway point-cloud understanding is relevant and which technologies make it possible, this chapter explains why the present project is worth developing, what specific goals it pursues, and how those goals are organized into a coherent plan.

4.1 JUSTIFICATION

The justification of the project must be understood at three complementary levels: technical, scientific, and practical. From a technical perspective, the project addresses a complex perception problem that remains difficult despite recent advances in 3D deep learning. From a scientific perspective, it contributes to the evaluation of segmentation strategies on a railway-specific problem in which thin and minority classes are especially challenging. From a practical perspective, it is aligned with the growing need for scalable, data-driven inspection workflows in railway maintenance.

4.1.1 TECHNICAL JUSTIFICATION

Railway overhead infrastructure inspection is a technically demanding problem because the assets of interest are thin, spatially structured, safety-critical, and strongly embedded in large outdoor environments. In mobile LiDAR scenes, OCL related elements such as wires, supports, and associated structures are typically underrepresented with respect to dominant classes such as ground, vegetation, or large surrounding objects. As a result, their accurate segmentation is significantly more difficult than the segmentation of larger and denser railway elements [7], [11].

This difficulty is not merely a matter of visual complexity. It is also a data-processing challenge. Railway point clouds are large, irregularly sampled, and computationally expensive to process, which makes model design and training dependent on adequate preprocessing strategies such as tiling, normalization, voxelization, and sparse representations [7], [15].

In addition, point-cloud semantic segmentation is a necessary enabling step for more advanced downstream tasks. Without reliable semantic separation of railway overhead elements, later condition assessment and anomaly-oriented analysis become poorly constrained and much less interpretable. For that reason, improving segmentation quality is not only an isolated machine-learning objective, but also a prerequisite for future inspection and maintenance intelligence [1], [21], [26].

4.1.2 SCIENTIFIC JUSTIFICATION

The scientific relevance of the project comes from the fact that, although railway point-cloud segmentation has advanced significantly in recent years, important gaps remain in the treatment of fine-grained overhead infrastructure. Public railway benchmarks such as WHU-Railway3D [7] have made it possible to compare deep-learning approaches under reproducible conditions, but the segmentation of thin catenary-related classes remains a challenging problem. Similarly, newer resources such as SemanticRail3D [3] confirm that the research community is moving toward richer 3D railway benchmarks.

From a modeling perspective, the literature offers several architecture families for point-cloud understanding, including PointNet, PointNet++, RandLA-Net, and sparse-convolution approaches [15], [16], [22], [29]. However, their relative suitability for railway overhead segmentation is not obvious a priori, since each family makes different trade-offs between local geometric learning, hierarchical context, efficiency, and scalability. Comparing these paradigms under a consistent workflow is therefore scientifically valuable because it helps

clarify which types of representation and processing are most appropriate for this kind of infrastructure data.

Moreover, the project does not limit itself to reporting segmentation accuracy. Its broader scientific value lies in building a reproducible workflow that links data preparation, model comparison, evaluation criteria, and anomaly-oriented extension within the same technical framework.

4.1.3 PRACTICAL AND INDUSTRIAL JUSTIFICATION

From a practical point of view, the project addresses a real operational need. Railway infrastructure managers require scalable and objective methods to inspect large networks while reducing dependence on purely manual review. A system capable of semantically structuring railway point clouds can help narrow the inspection focus, prioritize expert attention, and support more systematic asset monitoring.

The project is also attractive from an industrial perspective because it is built on public data, open model families, and reproducible processing steps. This lowers the barrier to adoption compared with solutions that depend on proprietary closed pipelines. In addition, the modular nature of the workflow makes it transferable: once established for catenary-related elements, the same general pipeline can be adapted to other railway assets or even to other infrastructure domains requiring large-scale 3D scene understanding.

In that sense, the value proposition of the project is not only “a segmentation model,” but a structured technical asset with three kinds of value, as seen in Figure 17:

1. **scientific value**, because it benchmarks advanced 3D segmentation strategies on railway data;
2. **engineering value**, because it defines a reproducible end-to-end workflow; and

3. **industrial value**, because it provides a foundation for future inspection and anomaly-detection systems in railway maintenance.

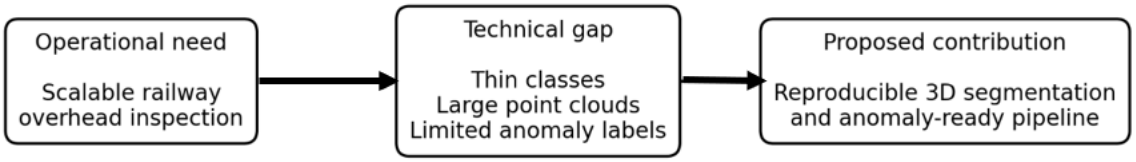

Figure 17: Project justification

4.2 OBJETIVES

To translate the project justification into a concrete development plan, the work is structured around one general objective and a set of specific objectives. These objectives define both the technical scope of the thesis and the criteria by which its contribution can be assessed.

The general objective of this project is to design, implement, and evaluate a reproducible deep-learning-based workflow for the semantic segmentation of railway mobile LiDAR point clouds, with particular emphasis on overhead-contact-line-related elements and with a view toward future anomaly-oriented inspection.

Objective 1. Establish a reproducible data-driven workflow

Develop a complete and transparent pipeline for catenary inspection, from data acquisition to preprocessing and annotation. This includes cleaning, normalization, and labeling of key components (contact wire, messenger wire, droppers, insulators, and supports) using representative datasets such as WHU-Railway3D. The workflow ensures reproducibility and provides a foundation for future benchmarking.

Objective 2. Evaluate and optimize deep learning architecture for segmentation

Analyze and compare state-of-the-art models, such as PointNet, PointNet++, RandLA-Net and Minkowski Engine, for both 3D and image-based segmentation of catenary components. Baseline models using MLPs and CNNs serve as references to assess trade-offs in accuracy, scalability, and computational cost.

Objective 3. Design and implement an anomaly-detection module

Develop a hybrid module capable of identifying deviations from normal catenary conditions through unsupervised or reconstruction-based learning. Autoencoders, one-class classifiers, and GAN-based approaches are explored to detect rare or unseen faults. The system integrates geometric constraints to improve interpretability and minimize false detections.

Objective 4. Build and validate an integrated processing pipeline

Combine 2D imagery and 3D point-cloud data into a unified framework that supports efficient training, inference, and visualization. The pipeline incorporates voxelization, spatially separated training sets, and augmentation strategies for robust generalization. Validation relies on quantitative metrics, mean Intersection-over-Union (mIoU), F1-score, precision, recall, and AUC, complemented by expert visual inspection.

4.3 METHODOLOGY

The methodology of the project is organized as a sequence of interconnected phases designed to ensure technical rigor, reproducibility, and coherence between the project objectives and the implemented system. The project is structured as a complete workflow that begins with railway point-cloud preparation, continues through model development and comparative experimentation, and culminates in a validated semantic-segmentation pipeline that can later support anomaly-oriented inspection.

A relevant aspect of this methodological design is that it combines machine-learning and deep-learning experimentation with an engineering perspective. This means that not only does the project obtain the highest possible segmentation score, but also it considers how the data are prepared, how the different model families are evaluated under comparable conditions, and how the resulting outputs can be integrated into a meaningful inspection-oriented workflow. This is particularly important in railway point-cloud analysis, where data preparation, computational feasibility, and the interpretability of the resulting segmentation are main parts of the problem.

The overall methodological logic of the project is summarized in the following Figure 18, which shows how raw railway point clouds are progressively transformed into structured semantic outputs through preprocessing, model training, comparative evaluation, and final integration into an anomaly-ready workflow.

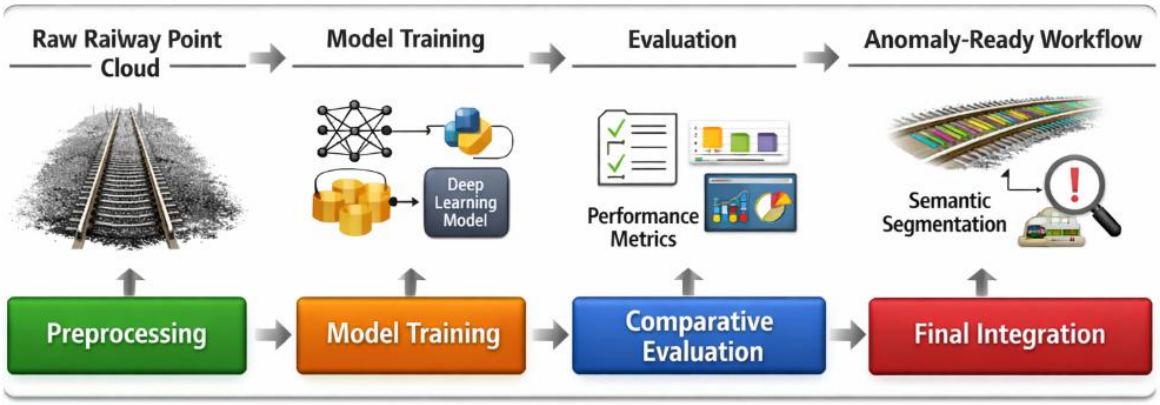

Figure 18: High-level methodological workflow of the project

4.3.1 PHASE 1: DATA ACQUISITION AND PREPROCESSING

The first phase of the methodology addresses data acquisition, inspection, and preprocessing. Since the project is centered on railway semantic segmentation in 3D, the main data source is **WHU-Railway3D** [7], a public large-scale mobile LiDAR benchmark specifically designed for railway point cloud semantic segmentation. The dataset covers approximately **30 km of railway infrastructure** and contains around **4.6 billion annotated points**, distributed across **urban, rural, and plateau railway environments**. It provides point-wise semantic annotations for **11 railway-related classes**, including elements such as rails, track bed, masts, support devices, overhead lines, fences, poles, vegetation, buildings, ground, and other surrounding objects.

In this thesis, the dataset is used as the experimental basis because it provides both the scale and the diversity required to evaluate semantic segmentation models under realistic railway conditions. The selected scenes include different types of railway environments, with variations in infrastructure density, surrounding vegetation, terrain complexity, and the presence of overhead-contact-line-related components. This is especially relevant for the objective of the work, since the target elements associated with the catenary system are usually thin, sparse, and visually complex within the global point cloud.

Once the data is understood, preprocessing becomes a critical methodological step. Railway point clouds are large, irregularly sampled, and heterogeneous, and therefore require a structured preparation process before they can be used for learning. This preparation includes the treatment of per-point attributes, spatial normalization, the partition of large scenes into manageable units, and, where required, voxelization to regularize the spatial representation and reduce computational burden. The purpose of these operations is to preserve relevant geometric information while making the data suitable for training and evaluation.

Attention is given to the treatment of overhead-contact-line-related elements. These classes are often thin, sparse, and underrepresented with respect to dominant classes such as ground, vegetation, or surrounding infrastructure. Therefore, preprocessing is not treated as a purely technical formality, but as a stage that directly influences the difficulty and fairness of the subsequent learning process. In addition, the data are partitioned into training, validation, and test subsets using a spatially coherent logic to reduce the risk of information leakage and to ensure that the evaluation reflects generalization to unseen scenes rather than memorization of nearby structures.

4.3.2 PHASE 2: MODEL DESIGN AND TRAINING STRATEGY

The second phase focuses on the selection, design, and training of the deep-learning models considered in the project. The methodological logic followed here is progressive: experimentation begins with simple reference models and then advances toward more expressive and specialized architectures for point-cloud understanding. This progression allows the project to understand how much benefit is obtained when increasingly sophisticated forms of spatial modelling are incorporated.

At the baseline level, simple per-point models provide an initial reference point. On top of these baselines, the methodology incorporates more specialized architectures such as PointNet, PointNet++, RandLA-Net, and sparse-convolution-based networks built on the Minkowski paradigm. Each of these model families embodies a different strategy for

learning from point clouds, ranging from direct point-wise processing to hierarchical aggregation, local neighbourhood learning, efficient large-scale sampling, and sparse 3D convolution.

The experimental design also reflects the scale and computational complexity of the work. The original WHU-Railway3D dataset comprises large railway mobile LiDAR scenes distributed across three representative environments: plateau, rural, and urban railway areas. In the processed experimental structure used in this thesis, these environments were organized into three spatial splits per scenario, resulting in nine train/validation configurations: **plateau_splitA, plateau_splitB, plateau_splitC, rural_splitA, rural_splitB, rural_splitC, urban_splitA, urban_splitB, and urban_splitC**.

Across these configurations, the processed benchmark contained **185 training .ply cases** and **86 validation .ply cases**, resulting in **271 accumulated train/validation cases**. These figures refer to the accumulated cases across the different split configurations, since the purpose of the split design is not only to increase the amount of data processed, but also to test the behaviour of the models under different spatial partitions of the railway scenes. Therefore, the experimental campaign is not based on a single fixed division of the data, but on repeated train/validation configurations covering different railway contexts and scene distributions.

Training is further expanded by running the main architectures across different random seeds, particularly seeds 42, 43, and 44. This is done to reduce the dependence of the results on a single initialization and to make the comparison more robust. As a result, the project involves multiple independent training and evaluation runs across several dimensions: model family, railway scenario, spatial split, and random seed. This structure provides a more reliable view of model performance than a single train/validation experiment, especially in a context where the target classes related to railway infrastructure and overhead contact line components are sparse and geometrically complex.

All experiments are carried out under the computational constraints available for the project, using a university CPU-based computing environment rather than GPU acceleration. The

working environment was based on an Intel Xeon E5-2640 v4 CPU at 2.40 GHz, with multiple CPU cores and high system memory, but without dedicated GPU support. This limitation had a direct impact on the methodological design. Complete railway point clouds could not be processed as full scenes in a straightforward way, so the models had to work with manageable spatial units, controlled point sampling, and, where necessary, voxelized or sliding-window representations.

The CPU-only setting also influenced the time and complexity of the experimentation. Training and evaluation could not be treated as immediate interactive processes, but as a sequence of controlled executions, with logs, checkpoints, validation tables, and summary files generated for later comparison. This was especially relevant for the more demanding architectures, such as RandLA-Net and Minkowski-based sparse convolutional networks, whose training and inference procedures required additional control of memory usage, voxel size, batch size, validation frequency, checkpoint selection, and spatial coverage during evaluation.

The training stage is therefore carried out under a controlled comparative logic. Rather than treating each architecture as an isolated experiment, the methodology evaluates the different models under a common conceptual framework, with comparable data preparation, validation monitoring, and result tracking. The same railway scenarios, spatial split logic, and evaluation criteria are used whenever possible, so that differences in performance can be attributed primarily to the modelling strategy rather than to inconsistent experimental conditions.

The models are assessed as part of the same experimental question: which type of deep-learning strategy is most suitable for railway point-cloud semantic segmentation under the constraints of the project? The internal comparison logic used in this phase is summarized in Figure 19, where the different model families are evaluated under a common preprocessing and assessment framework before converging into a unified comparative analysis.

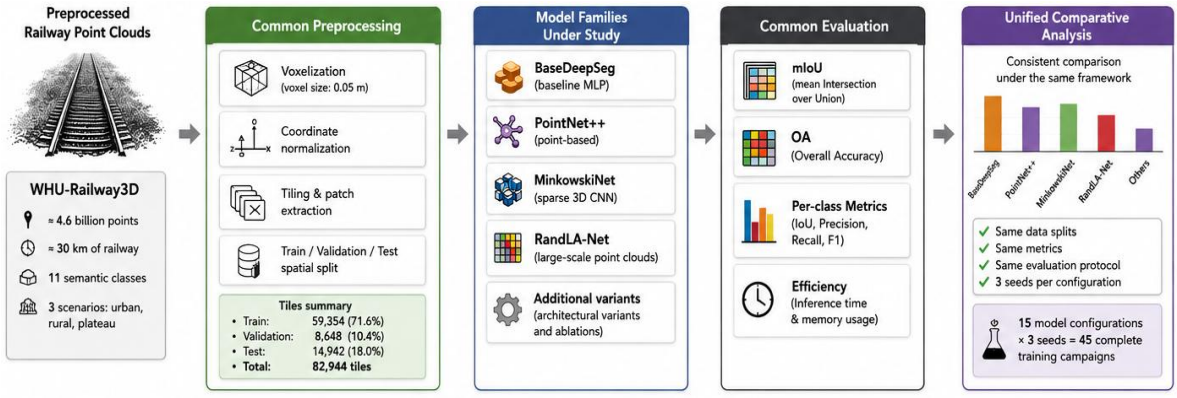

Figure 19: Experimental design used to compare segmentation architectures

4.3.3 PHASE 3: EVALUATION AND COMPARATIVE ANALYSIS

The third phase of the methodology is devoted to the systematic evaluation of the trained models. This phase is essential because the purpose of the project is not simply to train several architectures, but to understand their behavior under a consistent and meaningful comparison framework. The evaluation, therefore, combines quantitative criteria with qualitative inspection of the predicted railway scenes.

At the quantitative level, the analysis relies on standard segmentation metrics such as overall accuracy, class-wise Intersection over Union, and mean Intersection over Union. These measures provide a structured view of model behavior at both global and class-specific levels. However, because the project focuses on railway scenes with thin and minority overhead classes, the interpretation of the results does not rely only on aggregate values.

At the qualitative level, the segmented outputs are visually inspected to determine whether the predicted semantics are coherent with the geometry of the railway scene. This visual inspection is especially important in railway point-cloud analysis because numerically acceptable scores do not always guarantee structurally plausible predictions in thin and spatially complex elements.

Another relevant aspect of this phase is the analysis of trade-offs. The methodology explicitly considers not only segmentation quality, but also robustness, scalability, and practical suitability for the available computational environment. Consequently, the comparison between models is not reduced to identifying the best numerical score, but instead aims to understand the relationship between accuracy, computational burden, and operational realism.

4.3.4 PHASE 4: INTEGRATION, VALIDATION, AND ANOMALY-ORIENTED EXTENSION

The fourth phase integrates the outcomes of the previous stages into a coherent and reusable workflow. After comparative analysis, the most informative preprocessing decisions, model families, and evaluation criteria are consolidated into an integrated pipeline that can serve both as the result of the thesis and as a basis for further work.

Validation in this phase is carried out on held out or previously unseen data to assess how well the selected workflow generalizes beyond the conditions directly observed during training. It is valuable to do this final validation stage, because the value of the project depends not only on performance within the training loop, but also on the consistency of the overall workflow when applied to new data partitions or different railway scenes.

In addition, this phase defines the bridge toward anomaly-oriented inspection. Once the railway scene has been semantically segmented, the resulting point-wise labels can be used to isolate the relevant structures and to constrain the space in which anomalies are sought. This provides a natural foundation for future extensions based on reconstruction-oriented methods, one-class strategies, or geometry-informed consistency checks. In the present thesis, anomaly detection is therefore framed as a downstream extension enabled by semantic segmentation, rather than as an independently developed subsystem.

Taking together these four phases, they provide a complete methodological path from raw railway point clouds to structured semantic outputs and anomaly-ready scene representations. This phased organization ensures that the project can be understood as a coherent engineering process and prepares the ground for the more detailed implementation and development chapters that follow.

4.4 PLANNING AND ECONOMIC ESTIMATION

This section presents the temporal organization of the project and an approximate estimation of the development cost. Its purpose is twofold. First, it shows that the work is structured according to a realistic sequence of activities, with clear dependencies between conceptual definition, experimentation, evaluation, and final documentation. Second, it provides an approximate valuation of the engineering effort involved in the project if it is considered as a small-scale technical development based on research, implementation, and validation tasks.

The planning perspective is important in a project of this kind because the outcome depends not only on the quality of the models, but also on the correct ordering of the activities. Tasks such as literature review, dataset inspection, preprocessing, model implementation, experimentation, evaluation, and documentation are strongly interdependent.

From the economic point of view, the project is based mainly on public datasets and open-source tools, which significantly reduce direct licensing costs. However, this does not mean that the project is cost-free. The most important resource involved is specialized engineering time, followed by computational infrastructure and modest contingency margin.

4.4.1 PLANNING

The temporal organization of the project follows a progressive logic. The first stage is devoted to literature review, technology analysis, and project definition, since these tasks establish the conceptual and methodological foundation of the work. Once the problem is clearly framed, the focus shifts toward dataset inspection, preprocessing design, and preparation of the experimentation environment. These activities are necessary before model development can begin, because they define the structure and quality of the data that will be used in the learning pipeline.

The central stage of the project is dedicated to model implementation and experimentation. In line with the methodological design, the work starts with simpler baselines and progressively advances toward more expressive point-cloud architectures. This progression is important because it allows the project to compare not only final performance, but also the relative improvement associated with increasingly sophisticated learning strategies. After experimentation, the focus moves toward comparative evaluation, integration of the most relevant outcomes, and preparation of the anomaly-oriented extension. Finally, the last stage is devoted to writing, revision, and defense preparation.

The overall timeline of the project is summarized in Figure 20**¡Error! No se encuentra el origen de la referencia.**, which presents the main work packages in Gantt-chart form.

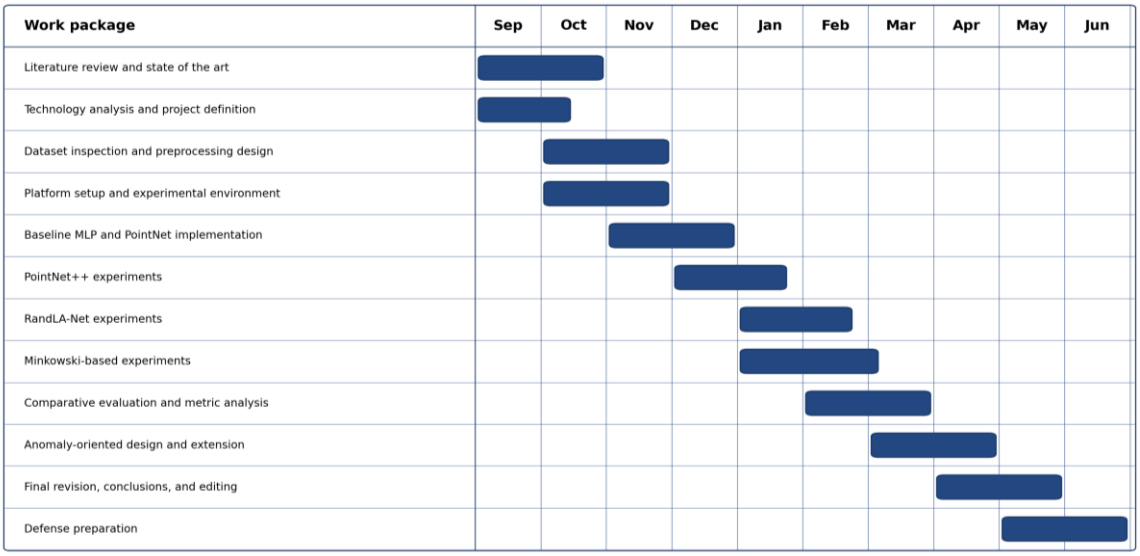

Figure 20: Gantt chart of the project schedule

This planning reflects the actual logic of the project: conceptual work first, then data preparation and setup, followed by experimentation, evaluation, and final integration into the written thesis.

4.4.2 ECONOMIC ESTIMATION

The economic estimation of the project assumes that it is carried out as a specialized engineering and research effort using public datasets and open-source software. Under this assumption, the main cost driver is qualified labor, since most of the project lies in technical analysis, preprocessing, experimentation, evaluation, and documentation. Additional costs are associated with computing resources, storage, and a contingency margin for unforeseen needs.

The estimation presented here is not intended as a commercial quotation, but as a realistic approximation of the development effort required to reproduce a project of similar scope

under professional conditions. For this reason, the cost structure is divided into two categories: personnel costs and technical-resource costs.

Cost item	**Estimated hours / units**	**Unit cost**	**Subtotal**
Literature review and project definition	50 h	25 €/h	1,250 €
Data inspection and preprocessing	80 h	25 €/h	2,000 €
Platform setup and experiment preparation	60 h	25 €/h	1,500 €
Baseline and advanced model implementation	140 h	25 €/h	3,500 €
Training, evaluation, and comparative analysis	90 h	25 €/h	2,250 €
Documentation and thesis writing	80 h	25 €/h	2,000 €
Personnel subtotal			12,500 €
Compute infrastructure equivalent cost	1	900 €	900 €
Storage, backup, and auxiliary resources	1	200 €	200 €
Software licenses	1	0 €	0 €
Technical resources subtotal			1,100 €
Contingency (10%)			1,360 €
Total estimated cost			14,960 €

Table 1: Estimated development cost of the project

As shown in Table 1, the largest share of the estimated cost corresponds to personnel. This is coherent with the nature of the project, since most of the effort lies in highly specialized tasks such as data preparation, deep-learning experimentation, model comparison, and technical reporting. The direct monetary cost of tools is relatively low because the project relies on public benchmarks and open-source model families. However, this reduction in licensing cost does not reduce the complexity of the engineering work itself.

The estimate should also be interpreted considering the academic context of the thesis. In a university setting, access to computational infrastructure, software environments, and research supervision can partially absorb some of these costs. Even so, presenting the equivalent development cost is useful because it highlights the real technical value of the work and makes it explicit that a project of this kind represents a substantial engineering effort rather than a purely theoretical exercise.

Finally, it should be noted that the cost would increase considerably if the project were extended toward an industrial-grade solution. In that case, additional expenses would appear in areas such as large-scale annotation, field validation, integration with operator systems, interface development, deployment maintenance, and long-term monitoring support.

Chapter 5. PLATFORM CONFIGURATION

This chapter presents the computational platform and technical configuration used to develop the project. While the previous chapter defined the methodological approach and the main phases of the work, the purpose of the present chapter is to describe the environment in which the project was executed, the operational constraints that shaped it, and the configuration decisions that made the full workflow feasible.

The proposed system must support the processing of large 3D railway point clouds, the training and evaluation of several deep-learning models, the generation of anomaly-oriented outputs, and the production of visualization-ready results for analysis and interpretation. In addition, the project is developed under an important practical constraint: the available execution environment did not include a dedicated GPU. Therefore, computational efficiency, memory management, experiment persistence, and careful organization of intermediate and final outputs become central aspects of platform design.

This chapter is organized into two main parts. First, the system requirements and operational constraints are analyzed. Second, the design and implementation of the platform are presented, including its logical architecture, hardware and software environment, dataset and directory organization, and execution workflow.

5.1 SYSTEM ANALYSIS

The design of the platform is conditioned by the characteristics of the problem itself. This project focuses on semantic segmentation and anomaly-oriented analysis of railway catenary point clouds. Unlike conventional tabular or image-based tasks, point-cloud processing introduces specific technical demands. Each scene is represented as an irregular and unstructured set of 3D points, which makes both data handling and model execution more complex than in regular-grid domains. In addition, the scenes are relatively large, several train-validation splits had to be managed, and different model families had to be compared, from lightweight baselines to more computationally demanding sparse-convolution approaches.

From a platform perspective, these conditions translated into several practical requirements. First, the environment must provide enough memory to load, preprocess, and batch point-cloud data without excessive simplifications. Second, it must support long-running executions in a stable way, since some training and evaluation tasks cannot be completed as short interactive jobs. Third, the full experimentation process must remain reproducible, which required a clear organization of raw data, derived splits, checkpoints, logs, summaries, anomaly outputs, and visualization artifacts.

Flexibility is another important requirement. The project is not limited to a single workflow but progressively evolved from initial preprocessing and baseline segmentation to more advanced models, cross-scene evaluation, anomaly-oriented post-processing, and visual inspection of results. For that reason, the platform must operate as a reusable technical foundation capable of supporting the complete development cycle.

A major practical constraint is the absence of a dedicated GPU. The available computational resource is an institutional ICAI cluster node equipped with an Intel Xeon E5-2640 v4 processor at 2.40 GHz, 20 physical cores, 40 threads, and 251 GB of RAM, but without GPU acceleration. This condition has a direct impact on the system configuration. In a GPU-based setting, many 3D deep-learning architectures can be trained comparatively quickly; in a

CPU-only environment, the same models require longer runtimes, more conservative batch configurations, and stricter control of memory usage. Consequently, the platform must be designed around realistic execution strategies rather than ideal hardware assumptions.

This limitation does not make the project infeasible, but it changes the optimization criteria. Instead of maximizing throughput, the platform must prioritize stability, continuity of execution, efficient use of RAM, and the possibility of reusing intermediate outputs. In practice, this makes data representation, checkpoint storage, persistent execution, and output traceability just as important as the training scripts themselves.

The software requirements are equally clear. The platform needs a development stack capable of supporting numerical computing, deep learning, and specialized 3D processing. Python is selected as the main language because of its maturity and wide adoption in scientific computing. PyTorch provides the general deep-learning framework, while MinkowskiEngine was incorporated for sparse 3D convolutional experimentation. Linux is the operating system of the execution environment, which is well suited to remote access, command-line experimentation, and long-running processes. The software configuration also must remain modular enough to integrate preprocessing utilities, training scripts, evaluation routines, anomaly-scoring procedures, and result-visualization tools without breaking the overall structure of the project.

Data organization is also a central part of the analysis. The project relies on the WHU-Railway3D [7] dataset, which contains railway point-cloud scenes with semantic labels. However, the raw dataset alone is not sufficient for experimentation. Since the work involves repeated training, validation, transfer analysis, anomaly-oriented post-processing, and the generation of summaries and visual outputs, a derived data structure is needed to store processed subsets, internal splits, prediction artifacts, and result files in an orderly way. Maintaining a strict separation between raw inputs and generated artifacts is necessary to preserve traceability and prevent accidental overwriting.

Overall, the system analysis shows that the platform must satisfy five main conditions: it must be computationally stable, capable of handling memory-intensive point-cloud

workloads, compatible with CPU-only execution, flexible enough to support segmentation and anomaly-related workflows and organized in a way that guaranteed reproducibility.

5.2 DESIGN AND IMPLEMENTATION

Once the system requirements and operational constraints have been identified, the platform is designed and configured as a modular experimentation environment for railway point-cloud analysis. Its purpose is to provide a stable and reproducible framework able to support the complete workflow, from data preparation to model execution, anomaly-oriented post-processing, and result visualization.

5.2.1 LOGICAL ARCHITECTURE OF THE PLATFORM

The platform is structured around five main layers: raw data storage, preprocessing and split generation, model execution, post-processing and anomaly analysis, and output management and visualization. This organization makes it possible to separate the major stages of the workflow while preserving a coherent end-to-end experimentation pipeline.

The first layer corresponds to the raw dataset. This layer stores the original WHU-Railway3D scenes and acts as the immutable source of the project. Keeping this layer isolated preserves traceability and prevents the original files from being mixed with transformed data or experiment outputs.

The second layer is the preprocessing and split-generation stage. Its function is to transform the raw scenes into an organized structure suitable for training, validation, and controlled experimentation. This includes the preparation of derived partitions and split-specific subsets that can later be consumed consistently by different model families. By separating this stage from model execution, the platform avoids unnecessary coupling between data handling tasks and learning logic.

The third layer is the model-execution stage. This is the layer in which the segmentation models are trained, validated, and evaluated. The platform is designed so that different model families can be executed under a common organizational scheme, even when their internal architectures differ substantially. This makes the project evolve from simpler baselines to more computationally demanding models without requiring a redesign of the overall environment.

The fourth layer corresponds to post-processing and anomaly-oriented analysis. Once semantic predictions are generated, the platform supported additional procedures to transform raw outputs into more interpretable information. This includes the generation of per-class predictions, anomaly-related indicators, and structured outputs suitable for later inspection and comparison.

The fifth layer is the output-management and visualization stage. This layer stores the artifacts generated by the experiments, including checkpoints, logs, summary tables, anomaly outputs, and visualization-ready files. Its purpose is not simply to save results, but to make the full experimentation process interpretable and auditable. Since the project involves repeated runs across scenes, splits, seeds, and configurations, a structured output layer identifies which experiments are executed, which artifacts are produced, and which results can be compared or visualized.

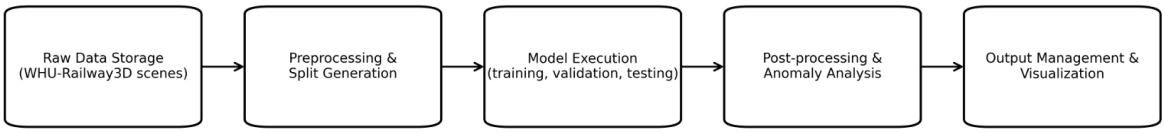

Figure 21: Logical Architecture

The above Figure 21 illustrates this layered design. It shows how the workflow progresses from the raw dataset to preprocessing and split generation, then to model execution, anomaly-oriented post-processing, and finally to the storage and visualization of experimental outputs.

5.2.2 HARDWARE AND SOFTWARE ENVIRONMENT

The logical design described above is implemented on a Linux-based institutional cluster provided by ICAI. The computational node used for the project is equipped with an Intel Xeon E5-2640 v4 processor running at 2.40 GHz, 20 physical cores, 40 threads, and 251 GB of RAM. No dedicated GPU is available.

From a computational point of view, the available RAM is particularly valuable for handling memory-intensive point-cloud workloads. Large railway scenes may contain a high number of points and associated attributes, so sufficient memory capacity was essential for data loading, preprocessing, model input preparation, and evaluation. The multicore CPU environment also supports the execution of several parts of the workflow, especially file handling, data preparation, batch-oriented scripts, and long-running model training and inference.

However, the absence of GPU acceleration imposes a significant restriction. Deep-learning models that normally benefit from GPU-based parallelization are executed in a CPU-only setting, which increases training times and requires more conservative operational decisions.

The software stack is selected accordingly. Linux provides the base environment for remote access and persistent command-line execution. Python is used as the main programming language because it allows the complete workflow to be maintained within a unified environment. PyTorch serves as the central deep-learning framework, providing enough flexibility to support different segmentation approaches under CPU-based conditions. In addition, MinkowskiEngine is incorporated to enable sparse 3D convolutional experimentation, which was relevant for the most advanced model line explored in the project.

The strength of this software configuration lies in its coherence. Instead of distributing the workflow across unrelated tools, preprocessing routines, experiment scripts, evaluation procedures, anomaly analysis, and result-generation stages are integrated into a common technical environment. This reduces fragmentation and helps maintain consistency

throughout the full experimentation process. The following Figure 22 summarizes the main hardware and software resources available in the project, reinforcing the practical context in which the platform was configured.

Figure 22: main hardware and software resources

5.2.3 DATASET AND DIRECTORY ORGANIZATION

A key part of the platform implementation is the organization of the dataset and project directories. Since the work involved repeated experiments, multiple splits, and the generation of many intermediate and final artifacts, a clear directory structure was necessary to keep the workflow understandable and reproducible.

The data organization followed a three-level logic. First, the raw WHU-Railway3D data is preserved as the original source layer. Second, a derived data layer is used to store processed partitions and split-specific structures required by the experiments. Third, a separate output layer is used for generated artifacts such as checkpoints, logs, summaries, anomaly-related files, and visualization outputs. This separation reduces ambiguity and prevents source data from being mixed with experimental products.

The raw-data area contains the original point-cloud scenes and semantic labels. These files act as the reference input of the project and were kept separate from all transformed or generated elements. The derived-data area stores the prepared subsets required for training

and validation, allowing the project to maintain a clean distinction between source files and experiment-ready data.

The output area is designed to support experiment traceability. Since each run can produce multiple artifacts, including saved model states, execution traces, evaluation summaries, anomaly indicators, and visual representations of results, a structured storage logic is necessary. Organizing these files systematically makes it easier to identify candidate runs, revisit previous experiments, and prepare later comparison and interpretation tasks. In the following Figure 23 it is shown the data directory logic.

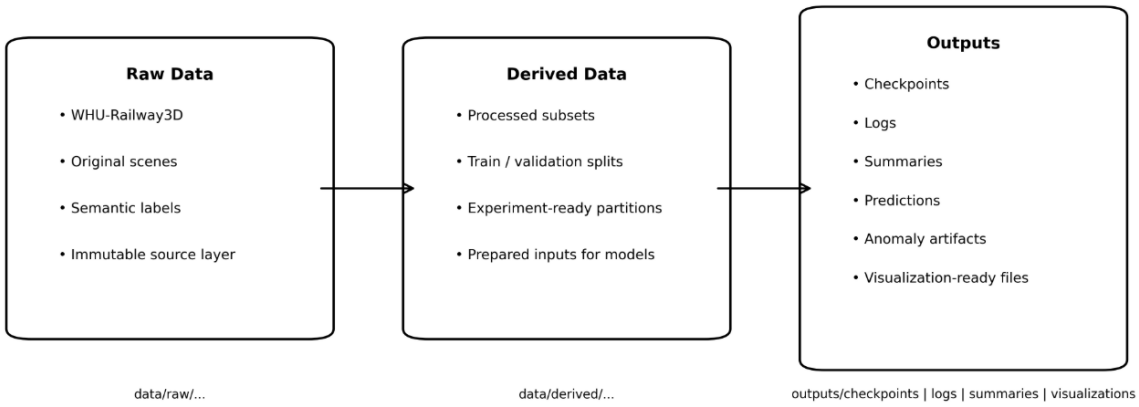

Figure 23: Data directory logic

5.2.4 EXECUTION WORKFLOW, PERSISTENCE, AND VISUALIZATION SUPPORT

The final part of the platform implementation concerns the way in which experiments are executed, maintained over time, and transformed into interpretable outputs. Because the project is developed in a CPU-only environment, some training and evaluation runs require long execution times, making execution persistence a necessary feature rather than a convenience.

The workflow follows a stable sequence. First, the relevant raw point-cloud scenes are accessed from the dataset layer. Second, the corresponding derived partition or split

configuration is selected as experiment input. Third, the chosen training or evaluation script is launched within the configured software environment. Fourth, the generated outputs are stored in the output layer in the form of checkpoints, logs, prediction files, summary tables, anomaly-related outputs, and visualization-ready artifacts.

Persistent execution management is especially important in this setup. Long-running processes cannot depend on a single interactive session, so experiments are managed in a Linux environment that allow runs to continue over extended periods. This is particularly relevant for sparse-convolution experiments supported by MinkowskiEngine, which are among the most computationally expensive tasks in the project.

Checkpoint storage also plays a central role. Saving intermediate model states makes it possible to preserve partial progress, recover valuable runs, and avoid repeating costly executions from the beginning. In parallel, logs and summary files provide a structured record of the experimentation process, enabling later inspection and comparison without relying only on manual notes.

The platform also supports the generation of outputs for anomaly-oriented analysis and result visualization. Once predictions are obtained, additional scripts produce structured summaries and derived files that could be used to inspect segmentation quality, identify anomalous patterns, and visualize representative outputs.

This configuration supports an incremental research process. The project progresses from initial baseline experiments toward more advanced model configurations and later incorporated anomaly-related analysis and result visualization.

Overall, the design and implementation of the platform transform the available institutional resources into a coherent environment for railway point-cloud experimentation. The combination of a Linux-based cluster, a Python-centered software stack, a structured directory organization, and persistent execution management create the technical basis required for the project.

Chapter 6. SYSTEM DEVELOPMENT

This chapter describes the practical development of the system implemented in this project. It focuses on implementation decisions, integration logic, and technical adaptations required to make the proposed approach executable under real project conditions.

The development of the system follows a progressive logic. First, the raw point-cloud data must be inspected and transformed into a consistent and reusable input representation. Second, several semantic segmentation architectures are implemented and adapted to the characteristics of the dataset and to the limitations of the available computational environment. Third, a structured training and evaluation workflow is defined to compare different model families under a common protocol. Finally, the system is extended with an anomaly-oriented analysis layer designed to identify suspicious spatial regions and support their qualitative validation by means of visualization.

A key aspect of this development process is that the system is not conceived as a collection of isolated scripts, but as a reproducible experimental framework. This means that the implementation includes not only the segmentation models themselves, but also the supporting mechanisms required to organize derived datasets, manage experimental splits, store checkpoints and logs, consolidate summary outputs, and generate visualization-ready artifacts. This broader view of system development is especially important because the project was carried out in a CPU-only environment, where stable execution, persistence, and disciplined output management were essential.

6.1 DATA PREPARATION AND PREPROCESSING PIPELINE

The first major implementation block of the system is the data preparation and preprocessing pipeline. In this project, the input is not a single homogeneous point-cloud collection, but a structured railway dataset composed of three main scenes: ***plateau_railway***, ***urban_railway***, and ***rural_railway***. The original raw data follows a directory structure of the form ***data/raw/WHU-Railway3D/<scene>/train*** and ***data/raw/WHU-Railway3D/<scene>/test***, which provides the initial basis for data inspection and later experimentation.

However, the raw dataset organization is not sufficient on its own for the experimental needs of the project. Since the objective is to compare several segmentation models under controlled and reproducible conditions, the raw folders must be transformed into a derived experimental structure. For this reason, the preprocessing workflow is designed not only to inspect and standardize the original ***.ply*** tiles, but also to create a reproducible system of derived subsets that could be reused across models, seeds, and validation runs.

The implemented pipeline therefore combines two complementary goals. On the one hand, it standardizes the point-cloud inputs by verifying geometric coordinates, optional features, and semantic labels. On the other hand, it reorganizes the data into a derived experimentation framework composed of **nine validation splits in total**: ***plateau_splitA***, ***plateau_splitB***, ***plateau_splitC***, ***urban_splitA***, ***urban_splitB***, ***urban_splitC***, ***rural_splitA***, ***rural_splitB***, and ***rural_splitC***. Each of these derived subsets is further divided into ***train/*** and ***val/*** folders, which later enables systematic training and internal validation across architectures.

This preprocessing stage is therefore not limited to data cleaning in the narrow sense. It establishes the full experimental input layer of the project: raw scene inspection, semantic consistency checking, feature preparation, density control, and split generation. This is essential because all later stages of the system, such as training, evaluation, model comparison, and anomaly-oriented analysis, depended directly on the reliability of these prepared inputs.

6.1.1 RAW DATASET INSPECTION AND INPUT STANDARDIZATION

The development process starts with a direct inspection of the raw ***.ply*** files stored under the three main scenes of the dataset. From an implementation perspective, this stage is necessary because the point-cloud files cannot be treated as opaque containers. Before any model can be trained, it is necessary to verify which fields are available in each file, how labels are stored, and whether optional attributes such as intensity are present and usable.

The raw dataset structure of train and test is a useful starting point, but the files still must be inspected individually to validate their internal schema. The implementation checks the availability of the Cartesian coordinates and explores possible variants for the semantic label field, since different files may use slightly different naming conventions for conceptually equivalent information.

A flexible loading strategy is therefore implemented. Instead of assuming a single rigid field name, the parser is designed to detect label-related properties through a set of likely alternatives such as ***class, label, classification, category_id,*** or similar variants. This allows the preprocessing layer to remain robust even when the original file schema is not perfectly uniform.

The raw-data inspection stage also provides a first tangible notion of scale. The dataset is a structured set of railway tiles with varying point counts and scene characteristics. This directly affects the later need for voxelization, subsampling, and block-based inference. In the plateau experimentation workflow, for example, the internal training-side reference set comprises **16 tiles**, which already illustrates the need for a disciplined preprocessing and execution strategy when multiple architectures and repeated runs are involved.

6.1.2 SEMANTIC LABEL PREPARATION AND CLASS HANDLING

After the raw schema is inspected, the next step is to validate and standardize the semantic labels used for training and evaluation.

The label inspection process shows that the effective label configuration is not identical across all scenes. This distinction is important because it affects both model configuration and later evaluation.

This means that the pipeline must verify which labels are present in each experimental context and ensure that the segmentation models are configured consistently with that observed label space. The class-handling stage therefore links the raw annotations to the concrete ***num_classes*** logic used in training.

Another important aspect is class imbalance. In the railway catenary domain, structurally thin elements such as wires or droppers are much less frequent than broader or more visually dominant classes. This imbalance is not merely descriptive; it has direct implications for model learning and for metric interpretation. For this reason, the label-preparation stage is tightly connected to later decisions involving weighted losses, present-class evaluation criteria, and the selection of meaningful comparison metrics.

To preserve consistency across the project, once the semantic label handling is validated, the same conventions are propagated through the derived splits. This ensured that all model families operate on the same semantic definitions within each experimental setup.

6.1.3 FEATURE PREPARATION AND NORMALIZATION

Once the label structure is validated, the preprocessing pipeline focuses on the preparation of point-level features. The central input variables of the project are the three-dimensional coordinates of the points, but the final representation depends on the model family and on the availability of additional attributes such as intensity.

The feature design follows a pragmatic logic. All models rely on geometry as the primary source of information, but some experiments also incorporate ***intensity*** when it is available and beneficial. This is especially relevant in the lighter baselines, where the inclusion of intensity could enrich the point-wise representation without fundamentally changing the architecture. At the same time, raw intensity values are not used blindly; the preprocessing stage incorporates controlled scaling so that extreme values do not dominate the input space.

The coordinate preparation also requires normalization decisions. Since railway tiles may differ in absolute position and spatial extent, it is not appropriate to treat raw coordinates as directly comparable in all cases. The preprocessing workflow therefore includes normalization strategies intended to stabilize the numerical representation while preserving the relative geometry of the scene. In the lighter baselines, robust coordinate normalization based on statistics such as median and interquartile range proved especially useful, since those models depend very directly on the statistical distribution of the input features.

This feature-preparation stage ensures that the coordinates retain their geometric meaning while being numerically stable enough for repeated training and evaluation across different architectures.

6.1.4 VOXELIZATION, SUBSAMPLING, AND BLOCK GENERATION

A major practical challenge in the project is the size of the point-cloud tiles and the computational burden associated with processing them repeatedly in a CPU-only environment. For this reason, the preprocessing pipeline incorporates explicit density-control strategies.

The most important of these strategies is ***voxelization***. In the implemented workflow, voxel sizes such as ***0.15 m*** and, in some cases, ***0.30 m*** are used depending on the model and the experimental objective. This allows the project to reduce point density while preserving the main geometric arrangement of the railway scene. The choice of ***0.15 m*** provides a reasonable compromise between preserving thin structures and keeping the computational load under control.

In addition to voxelization, the project also relies on explicit point caps and local partitioning mechanisms. For large tiles, values such as ***max_points_per_tile = 1,000,000*** are used to prevent memory blow-ups. In block-based models, local processing units are constructed using values such as ***block_points = 1024***, while larger sampling settings such as ***batch_points = 32768*** or ***65536*** are used in training, depending on the architecture.

This density-control strategy is particularly important in the case of models such as PointNet++, RandLA-Net, and Minkowski-based sparse networks. Each of these architectures benefit from a controlled spatial reduction stage, although in different ways. In sparse-convolution settings, voxelization directly defines the sparse tensor structure. In point-based settings, subsampling and block generation limit the effective size of the local processing units while preserving local geometry.

The use of these concrete preprocessing values is one of the clearest examples of how the system was adapted to real computational constraints. Without voxelization, point caps, and block partitioning, the project would not have been operationally viable across repeated runs and model comparisons.

6.1.5 SPLIT GENERATION AND DERIVED DATA ORGANIZATION

The final stage of the preprocessing workflow is the creation of derived train and validation subsets that could be reused systematically across experiments. This is based on repeated experimentation across multiple segmentation architectures and **three fixed random seeds: *42, 43,* and *44***.

The derived-data layer created for experimentation comprises **nine scene-specific split** configurations stored under ***data/derived/: plateau_splitA, plateau_splitB… etc.*** Each split is organized into explicit ***train/*** and ***val/*** folders. In practical terms, the documented split sizes range from compact ***plateau*** subsets, such as the 9/4 train-validation distribution of ***plateau_splitA***, to larger ***rural*** subsets with 32 training tiles and 16 validation tiles per split, and ***urban*** subsets with 21–22 training tiles and 10–11 validation tiles.

As seen in Table 2, the experimental dataset is therefore heterogeneous not only in scene content but also in split scale, which reinforces the need for careful experiment tracking and for evaluation criteria capable of handling non-uniform class and tile distributions.

Scene	Derived split	Training tiles	Validation tiles	Tiles in split	File naming pattern	Typical class setup
Plateau	plateau_splitA	9	4	13	Tibet-*.ply	8 classes (0,1,4,5,6,8,9,10)
Plateau	plateau_splitB	8	1	9	Tibet-*.ply	8 classes (0,1,4,5,6,8,9,10)
Plateau	plateau_splitC	8	1	9	Tibet-*.ply	8 classes (0,1,4,5,6,8,9,10)
Rural	rural_splitA	32	16	48	section*.ply	11 classes (0–10)
Rural	rural_splitB	32	16	48	section*.ply	11 classes (0–10)
Rural	rural_splitC	32	16	48	section*.ply	11 classes (0–10)
Urban	urban_splitA	21	11	32	L*-M01*.ply	11 classes (0–10)
Urban	urban_splitB	21	11	32	L*-M01*.ply	11 classes (0–10)
Urban	urban_splitC	22	10	32	L*-M01*.ply	11 classes (0–10)

Table 2: Derived experimental structure and split sizes

This organization allows the project to disentangle the original dataset structure from experimental logic. The raw folders remain preserved as source data, while the derived folders act as the operational input for the segmentation experiments. This separation

simplifies automation, reduces ambiguity in the training scripts, and makes it easier to relate each run to a specific scene and split configuration.

From a methodological point of view, the derived split design also strengthens the fairness of the experimental comparison. Reusing the same split definitions across different models and seeds ensured that observed performance differences were more likely to reflect architectural behavior rather than accidental variation in the underlying subsets.

At the end of this stage, the dataset has been transformed from a raw collection of railway point-cloud files into a reproducible and explicitly organized experimental resource. This prepares the ground for the implementation and comparison of the segmentation models described in the next section.

6.2 IMPLEMENTATION OF THE SEGMENTATION MODELS

After establishing the data preparation pipeline, the next major development stage consists of implementing the semantic segmentation models used throughout the project. The goal of this stage is not simply to test several architectures independently, but to incorporate them into a shared experimental framework in which they can operate on the same derived datasets, be trained under controlled conditions, and later be compared using a common evaluation logic.

The implementation covers five model families representing different levels of spatial complexity: a lightweight per-point multilayer perceptron baseline, PointNet, PointNet++, RandLA-Net, and a sparse-convolution model based on a Minkowski-style U-Net design. In practical terms, these models are developed through dedicated training scripts, including ***baseline_mlp_xyz.py***, ***pointnet_seg.py***, ***pointnet2_lite_seg.py***, ***randla_lite_seg.py***, and ***minkunet_small_seg.py***. Their outputs are later connected to corresponding evaluation scripts such as ***eval_tile.py***, ***eval_tile_pointnet*.py***, ***eval_tile_randla_lite.py***, and

eval_tile_minkowski.py, which allows the models to be assessed under a common tile-level comparison workflow.

This implementation strategy builds a comparative framework in which different architectural families could be examined in terms of segmentation quality, computational feasibility, and suitability for later anomaly-oriented analysis.

Model	**Scripts**	**Role**
Baseline MLP	baseline_mlp_xyz.py / eval_tile.py	Lightweight baseline and sanity check
PointNet	pointnet_seg.py / eval_tile_pointnet*.py	Point-set benchmark
PointNet++	pointnet2_lite_seg.py / eval_tile_pointnet*.py	Stronger local-geometry benchmark
RandLA-Net	randla_lite_seg.py / eval_tile_randla_lite.py	Main large-scale model and anomaly basis
Minkowski CNN	minkunet_small_seg.py / eval_tile_minkowski.py	Sparse-convolution benchmark

Table 3: Model Overview

Model	Input	Controls
Baseline MLP	Per-point XYZ, optional intensity	Normalization, --use_intensity, point-wise batching
PointNet	Fixed-size point blocks	Block sampling, local subsets, optional global features
PointNet++	Hierarchical neighborhoods and blocks	block_points = 1024, block inference, optional intensity
RandLA-Net	Large point sets	batch_points = 32768/65536, voxel = 0.15, class weighting, seeds 42/43/44
Minkowski CNN	Sparse voxelized tensors	voxel = 0.15/0.30, max_points_per_tile = 1000000, sliding-window inference

Table 4: Input and controls

As shown in Table 3 and Table 4, the project does not rely on a single modeling strategy. Instead, the implemented architecture covers several representational regimes, from simple point-wise classification to sparse three-dimensional convolution. This diversity is useful not only for performance comparison, but also for understanding how different input representations and computational profiles behave in the context of railway catenary point clouds.

6.2.1 IMPLEMENTATION LOGIC ACROSS MODEL FAMILIES

Although the implemented models differ substantially in architectural design, they share several practical implementation principles. First, all of them are connected to the same derived data organization, which means that training and validation are carried out over the split-specific subsets stored under ***data/derived/.*** Second, all models are integrated into a training–evaluation cycle in which the output of the training script is not considered complete until it could be evaluated at tile level through the corresponding evaluation routine. Third, the experiments are repeated across fixed random seeds, 42, 43, and 44 in the main repeated workflows, to avoid drawing conclusions from a single favorable execution.

Another shared principle is the need to adapt the model input representation to the size of the point clouds and to the CPU-only computational environment. The raw tiles are too large to be processed identically by all the architectures. As a result, each model family requires a different implementation strategy: direct per-point processing for the baseline, block-based set processing for PointNet and PointNet++, efficient local aggregation for RandLA-Net, and sparse voxelized tensors for the Minkowski-based network. The purpose of the implementation stage is to ensure that all of them could operate on the same dataset under comparable experimental conditions.

The scripts themselves also reflect this modular design. Separate training and inference files are maintained for the main model families, which makes it easier to isolate architecture-specific logic while preserving the overall reproducibility of the system. This script-level separation is especially helpful once the project begins to scale into multiple runs, multiple seeds, and scene-specific evaluations.

6.2.2 BASELINE PER-POINT MLP

The simplest implemented model is a lightweight per-point multilayer perceptron, used as a baseline reference. This model is developed through the script ***baseline_mlp_xyz.py***, which operated directly on point-level feature vectors. The main input consists of normalized XYZ coordinates, and some runs also incorporate intensity information through the corresponding feature flag when the attribute was available and considered useful.

From an implementation perspective, the baseline model plays an important role in the early stages of the project. Because its data flow is comparatively simple, it is well suited for validating the consistency of the preprocessing pipeline, the label encoding logic, the loss computation, and the basic metric routines. It helps verify that the segmentation workflow is operational before moving toward more demanding architectures.

The baseline model also makes explicit one of the central questions of the project: how far point-cloud semantic segmentation can be pushed by a simple point-wise feature learner before richer spatial reasoning becomes necessary. Since this architecture does not model neighborhoods or hierarchical structure explicitly, it provides a useful lower-bound reference. Its implementation therefore has methodological value even if it is not expected to become the final practical choice for the anomaly-oriented stage.

At the same time, the limitations of this model are clear. Railway catenary scenes contain thin elongated structures and local geometric relationships that cannot be captured reliably through isolated point features alone. For that reason, the baseline is used mainly as a control model and reference benchmark rather than as a deployment-oriented candidate.

6.2.3 POINTNET

The next implemented architecture is PointNet, which introduces a more suitable representation for unordered point sets than the baseline per-point MLP. The training workflow is built around the script ***pointnet_seg.py***, while evaluation is handled through pointnet-specific tile-level inference scripts. In contrast to the baseline model, PointNet requires the input to be organized as fixed-size local point sets, which led to a block-based implementation strategy.

This block-based design is an important practical decision. Full railway tiles vary substantially in size and cannot be processed efficiently as single monolithic inputs. By dividing the point clouds into manageable local point subsets, the implementation makes PointNet compatible with the available hardware and with the broader experimental workflow. These blocks serve as effective training and inference units, allowing the model to aggregate information across a local set of points rather than treating each point independently.

From an implementation viewpoint, PointNet is useful because it represents a first step toward a genuinely point-cloud-specific segmentation pipeline. It preserves the permutation-invariant logic characteristic of point-set learning while remaining easier to manage than more elaborate hierarchical models. At the same time, the architecture still relies heavily on global aggregation within each block, which limit its ability to model fine local structure in detail.

Railway catenary elements often differ from one another not simply through absolute feature values, but through their relative spatial arrangement. PointNet improves over the baseline by moving from isolated points to local point sets, but it still does not provide the strongest local-geometry representation among the models implemented in the project.

6.2.4 POINTNET++

To capture local geometry more effectively, the project then incorporates a PointNet++-style model through the script ***pointnet2_lite_seg.py***. This architecture introduces hierarchical grouping and neighborhood-based abstraction, which makes it more suitable than PointNet for representing local structures at multiple spatial scales. The corresponding evaluation stage is handled through dedicated pointnet-style tile inference scripts.

The implementation of PointNet++ requires a more structured local-processing workflow than the previous models. The model workes with local blocks and neighborhood abstractions rather than with a single flat point set. For block-level operations, values such as ***block_points = 1024*** are used as part of the local inference and sampling logic. This allows the system to process large tiles through a collection of manageable units while preserving local spatial detail more explicitly than PointNet.

PointNet++ is one of the first models in the project capable of expressing not only set-level information but also neighborhood-scale relations. From a comparative perspective, it serves as a stronger point-based benchmark and helped clarify how much local abstraction mattered for the segmentation task.

However, the richer representation came with increased implementation complexity. The data pipeline, batching logic, and evaluation routines had to support hierarchical local processing, which made the model more demanding than the baseline MLP and PointNet. This trade-off between expressiveness and operational burden later became one of the key themes of comparative analysis.

6.2.5 RANDLA-NET

RandLA-Net is one of the most important models implemented in the project and, in practical terms, one of the most useful ones. It is developed through the script ***randla_lite_seg.py*** and evaluated through ***eval_tile_randla_lite.py***. Among the implemented model families, RandLA-Net offers one of the clearest balances between scalability and local-geometry modeling, which makes it especially attractive in the context of large railway point clouds processed in a CPU-only environment.

The implementation of RandLA-Net is tightly linked to controlled input-size management and repeated experimentation. Typical runs used values such as ***batch_points = 32768 or 65536***, often combined with ***voxel = 0.15*** and optional flags such as ***--use_intensity***, ***--class_weights***, and different sampler settings including random and balanced. This is important because it shows that RandLA-Net is not treated as a black-box imported model, but as a configurable architecture integrated into the specific constraints of the project.

Another important aspect of its implementation is the repeated execution across the fixed random seeds 42, 43, and 44, and across the scene-specific derived splits. This gives RandLA-Net a particularly strong presence in the experimental workflow, because it can be trained and compared systematically without becoming prohibitively expensive.

This balance is precisely why RandLA-Net later becomes the natural backbone for the anomaly-oriented extension. A downstream anomaly-analysis stage requires more than just a promising segmentation score; it also needs an operationally stable and repeatable model family. RandLA-Net satisfies these conditions better than several of the alternatives, which explains why the final anomaly workflow was built primarily on its strongest-performing split-and-seed configurations.

6.2.6 MINKOWSKI CNN

The most advanced model family implemented in the project is a sparse-convolution architecture based on a Minkowski-style U-Net, developed through ***minkunet_small_seg.py*** and evaluated with ***eval_tile_minkowski.py***. This model represents the most computationally demanding end of the implementation spectrum and requires a substantially different input representation from the point-based architectures.

The core of the Minkowski implementation is the use of sparse voxelized tensors. For this reason, voxelization is not simply a preprocessing convenience in this case, but an intrinsic part of the model representation. Typical configurations included ***voxel = 0.15*** and, in some experiments, ***0.30***, as well as point caps such as ***max_points_per_tile = 1000000*** to control the size of the evaluated scenes. Inference over large tiles require a dedicated strategy, and the implemented evaluation therefore uses a sliding-window voxel procedure with settings such as ***window_size_vox = 256*** and ***stride = 256*** to maintain broad coverage across the scene.

This model demonstrates the feasibility of incorporating sparse three-dimensional convolution into the project despite the absence of a GPU. At the same time, its implementation also makes the computational limits of the environment very visible. Training and inference are significantly more demanding than in the case of the point-based models, which means that the Minkowski workflow must rely heavily on disciplined checkpointing, persistence, and evaluation control.

6.2.7 INTEGRATION OF THE MODEL FAMILY INTO A COMMON EXPERIMENTATION FRAMEWORK

A major strength of the development process is that the implemented models are not left as isolated architecture-specific experiments. Instead, they are integrated into a common experimentation framework in which the same derived split organization, the same general seed logic, and the same tile-level comparison workflow are reused across model families.

This integration means that each model has its own training and evaluation scripts, but all of them are connected to the same broader workflow: prepared splits from ***data/derived/,*** persistent checkpoints and logs, tile-level evaluation outputs, and structured summary files for later comparison.

Furthermore, the integration reduces implementation fragmentation. Because the project uses separate but coordinated scripts for the main architecture, improvements in dataset handling, output organization, or evaluation logic could benefit more than one model family. As the experimentation progresses, this becomes increasingly important, especially when repeated runs must be interpreted jointly rather than as unrelated isolated results.

6.3 TRAINING STRATEGY AND EXPERIMENT MANAGEMENT

Once the segmentation models are integrated into the common development framework, the next step consists of defining a training strategy and an experiment-management procedure that make the comparison between architectures meaningful and reproducible. In a project that includes several model families, multiple derived splits, and repeated executions with different seeds, the experimental framework is not a secondary support layer but a central part of the system itself.

For this reason, the training and experiment-management stage is organized around four principles: **comparability, feasibility, persistence, and traceability**. **Comparability**

means that the implemented models are trained under conditions that remain sufficiently aligned to support fair evaluation, even if each architecture requires its own input format and operational adjustments. **Feasibility** refers to the need to adapt all executions to the CPU-only environment available for the project. **Persistence** means that long-running experiments must preserve checkpoints and logs in a disciplined way. **Traceability** ensures that every relevant result can be linked back to a specific model, scene, split, and random seed.

The experimental workflow consists of maintaining a controlled cycle in which prepared data subsets, training scripts, stored checkpoints, evaluation routines, and summary files remain systematically connected.

Parameter	Typical value(s) used in the project	Main usage context	Purpose in the workflow
Random seeds	42, 43, 44	Repeated training runs	To assess stability across executions and avoid conclusions based on a single favorable run
Voxel size	0.15 m, 0.30 m	Preprocessing / model input / evaluation	To control point density and balance geometric detail against computational feasibility
Maximum points per tile	1,000,000	Large-tile preprocessing and evaluation	To prevent memory overload and keep tile-level processing feasible
Batch points	32,768, 65,536	Training point-based models	To define manageable effective sample sizes during training
Block points	1,024	Block-based inference and local processing	To generate local units suitable for neighborhood-based models
Sampling strategy	random, balanced	Training data preparation	To control how points are drawn during training and reduce bias toward dominant classes
Intensity usage	optional / enabled when available	Feature preparation and model input	To enrich the geometric representation with an

			additional point-level attribute
Class weighting	enabled in selected runs	Loss definition during training	To mitigate class imbalance, especially for thin catenary-related classes
Present-class support threshold	50 points	Evaluation (mIoU_present)	To avoid unstable IoU interpretation for classes with extremely low support in a tile
Sliding-window voxel inference	window_size_vox = 256, stride = 256	Minkowski-based evaluation	To maintain broad tile coverage during sparse-convolution inference
Execution mode	CPU-only environment	Entire experimental workflow	To adapt the pipeline to the available hardware and enforce computationally feasible settings

Table 5: Recurring training and evaluation configuration values

As shown in Table 5, the experimental workflow relies on a compact but consistent set of configuration values that recur across the different stages of the project. Some of these parameters are mainly associated with computational feasibility, such as voxel size, point caps, or batch-related limits. Others are directly linked to the reliability of comparison, such as fixed seeds, class-weight handling, and presence-aware evaluation thresholds. Together, these settings define a controlled operational space in which the different model families can be trained and compared under reproducible conditions.

6.3.1 TRAINING SETTINGS AND CONFIGURATION LOGIC

Each model family requires its own operational configuration, but the project follows a common logic when defining the main training settings. The aim is to establish stable and interpretable configurations that can be reused across comparable experiments.

The most relevant recurring settings include the number of processed points per batch or local block, voxel size where applicable, the use or omission of intensity, sampling strategy, class weighting, and the fixed random seeds used for repeated runs. In the point-based families, values such as ***batch_points = 32768 or 65536*** are used to keep training feasible while still exposing the models to sufficiently rich local geometry. In local block processing, values such as ***block_points = 1024*** define manageable spatial units for training or inference. In sparse-convolution settings, voxel sizes such as ***0.15 m and 0.30 m*** control the granularity of the sparse representation and its associated computational cost.

The configuration logic remains deliberately conservative. The project does not attempt to maximize the performance of every architecture through a separate and highly specialized optimization loop. Instead, it seeks to define settings that are technically defensible, computationally realistic, and reusable across repeated runs. This is important because the goal of the project is not merely to obtain an isolated score, but to compare model families under a coherent framework and then carry forward the most convincing candidates into the anomaly-oriented stage.

Another key part of this logic is the explicit use of repeated seeds. The main repeated workflows rely on the fixed seeds ***42, 43, and 44***, which make it possible to distinguish stable tendencies from one-off favorable executions.

6.3.2 EXECUTION UNDER CPU-ONLY CONSTRAINTS

A defining condition of the project is the absence of a dedicated GPU. This condition does not operate as a minor inconvenience but as a structural constraint that shapes the full experimentation strategy. Large-scale point-cloud segmentation is computationally demanding by nature, and the lack of GPU acceleration means that model execution must be planned with particular care.

The most immediate consequence of this constraint is the need to control memory usage and execution time at all stages of the workflow. Large railway tiles, complex architectures, and repeated training runs can quickly become impractical if the system does not incorporate explicit density-control and batching strategies. For this reason, the project relies on mechanisms such as voxelization, point caps, local block generation, controlled batch sizes, and architecture-specific inference procedures.

The CPU-only condition also affects how training sessions are managed over time. Some runs complete within moderate execution windows, but others, especially those associated with sparse-convolution architectures, require substantially longer training and evaluation cycles. As a result, the project organizes execution in a persistence-oriented way. Long runs are launched in stable command-line sessions, and the workflow depends on systematic checkpointing and output storage rather than on short interactive experimentation.

This constraint also influences the comparative scope of the project. Some model families are easier to repeat systematically across several seeds and splits, while others impose a heavier cost per run. This does not invalidate the comparison, but it means that feasibility itself becomes part of the technical interpretation of the models.

6.3.3 CHECKPOINTS, LOGS, AND REPRODUCIBILITY

To support long-running executions and repeated experimentation, the project incorporates an explicit persistence layer based on checkpoints, logs, and structured summary outputs. These elements transform model training from a transient computation into a traceable experimental process. Without this persistence layer, the project could still run models, but it could not compare them reliably or recover from interruptions in a disciplined way.

Checkpoints play a central role in this design. During training, relevant model states are stored so that the workflow can preserve the progress of a run and later recover the most useful configuration without repeating the full execution from scratch. Checkpointing therefore supports both robustness and efficiency.

Logging serves a complementary purpose. While checkpoints preserve model states, logs preserve process information. They make it possible to track how a run evolves, whether training remains stable, and whether validation behavior appears promising or problematic. This visibility is important not only for debugging, but also for interpretation. A final metric is more meaningful when it can be understood as the result of a monitored training process rather than as an isolated opaque number.

The organization of persistent outputs also follows a structured logic. Model checkpoints are stored under directories such as ***outputs/checkpoints/,*** while comparative summaries and aggregated evaluation results are stored under ***outputs/summary/.*** This separation of output types simplifies the later analysis and reduces the risk of confusion once the number of runs increases.

A further contribution to reproducibility comes from naming conventions. Checkpoint and summary filenames encode relevant experimental dimensions such as model family, scene, split, and seed. In a multi-run project, poorly organized filenames can make later interpretation unreliable, whereas disciplined naming allows each result to be traced back to its exact experimental context. For this reason, reproducibility in the present project depends

not only on code and data organization, but also on persistent and interpretable output management.

6.3.4 MULTI-RUN EXPERIMENTATION AND MODEL SELECTION

Because the project compares several segmentation architectures and repeats key runs across multiple seeds and scene-specific splits, the experimentation strategy must operate at a multi-run level from the outset. The purpose is to construct a framework in which those executions can later be interpreted jointly.

The derived split structure and the architecture-specific scripts make this multi-run logic possible. Each experiment is associated with a clearly defined scene, split, seed, and model family, and its outputs are stored in a way that allows later comparison. Once several runs are completed, the project consolidates the relevant information into summary files that support comparative reading across models and configurations.

This consolidated view is particularly useful because it makes it possible to distinguish stable tendencies from accidental strong performances. A model that performs well only in one favorable run does not provide the same level of confidence as a model that behaves consistently across several related settings. For that reason, the project interprets repeated experimentation not as redundancy, but as a methodological safeguard against overinterpreting isolated results.

This multi-run logic also plays a decisive role in the transition toward the anomaly-oriented extension. The project does not select downstream anomaly candidates arbitrarily. Instead, it identifies strong-performing segmentation configurations from within the broader comparative workflow and uses those as the basis for further analysis. This is especially relevant for RandLA-Net, whose balance between segmentation quality, scalability, and repeatability makes it suitable for repeated execution and later anomaly-oriented use.

6.4 EVALUATION WORKFLOW

Once the segmentation models are trained under the common experimental framework, the next stage consists of evaluating their outputs in a consistent and comparable way. In this project, evaluation is not limited to computing a final score after training. It also includes the generation of tile-level predictions, the control of spatial coverage, the computation of robust multiclass metrics, and the consolidation of comparable outputs across model families, splits, and seeds.

Some models work directly on point-level features, others rely on block-based local processing, and the sparse-convolution approach uses voxelized sparse tensors. As a result, the evaluation workflow must translate these different internal representations into a common output space in which the predictions can be analyzed under shared criteria.

For this reason, the project treats evaluation as a dedicated component of the system rather than as a simple post-processing step. The evaluation scripts are designed to read trained checkpoints, apply them to complete tiles or to controlled local partitions, reconstruct comparable prediction outputs, and store summary files that later support the comparative analysis.

Component	Implementation	Main purpose
Tile-level inference	eval_tile.py, eval_tile_pointnet*.py, eval_tile_randla_lite.py, eval_tile_minkowski.py	To generate comparable predictions over full tiles or controlled local partitions
Main global metric	Overall Accuracy (OA) / acc	To quantify the overall proportion of correctly classified elements
Main balanced metric	mIoU_present with present_min_support = 50	To compare segmentation quality across classes while reducing instability from extremely rare classes
Coverage control	coverage, drop_frac	To verify that the evaluated prediction covers the intended tile extent and is not based on a small favorable subset
Sparse-convolution inference control	window_size_vox = 256, stride = 256	To maintain broad spatial coverage in Minkowski-based evaluation
Output scale indicators	raw_points, vox_points	To record the size of the original and voxelized representations during evaluation
Summary outputs	Files in outputs/summary/	To consolidate results
Model-selection artifacts	best_plateau_ckpts_present50.txt, best_urban_ckpts_present50.txt	To identify the strongest-performing configurations for later stages

Table 6: Main evaluation elements and comparison criteria

As shown in Table 6, the evaluation workflow combines metric computation with operational controls such as coverage and structured output generation. This is important because, in large point-cloud segmentation, a model is not only useful when it produces good scores, but when those scores are associated with sufficiently complete and interpretable predictions over the evaluated tiles.

6.4.1 TILE-LEVEL INFERENCE

The first requirement of the evaluation workflow is the generation of tile-level predictions from models that are trained on different input representations. This is a non-trivial step because the internal unit of processing is not identical across architectures. The baseline model operates directly on point-level features, PointNet and PointNet++ rely on local point subsets or blocks, RandLA-Net combines efficient aggregation over large point sets, and the Minkowski-based model works on sparse voxelized inputs. The evaluation procedure must therefore adapt to each family while still producing outputs that can later be compared in a common framework.

To address this, the project uses dedicated evaluation scripts for each model family. The baseline workflow relies on ***eval_tile.py***, point-based hierarchical models use the corresponding ***eval_tile_pointnet*.py*** routines, RandLA-Net uses ***eval_tile_randla_lite.py***, and the sparse-convolution workflow uses ***eval_tile_minkowski.py***. This script-level separation is necessary because inference logic differs across architectures, but the outputs are later consolidated into the same comparative space.

A central concern at this stage is spatial completeness. In large railway tiles, evaluation cannot rely on a naive one-pass procedure if that produces only partial predictions or becomes computationally unstable. The project therefore treats tile-level inference as a controlled reconstruction process. In point-based families, this means combining local predictions in a consistent way over the tile. In the sparse-convolution workflow, this means using a sliding-window voxel strategy with values such as ***window_size_vox = 256*** and

stride = 256, so that prediction coverage remains broad and the results do not depend on a small partial fragment of the scene.

6.4.2 METRIC COMPUTATION AND EVALUATION CRITERIA

Once tile-level predictions are generated, the next step consists of computing metrics that make the model families comparable under realistic conditions. The project does not rely on a single global score because railway point-cloud segmentation involves multiclass outputs, heterogeneous tile composition, and strong class imbalance. Under these conditions, a single metric can easily give an incomplete or even misleading picture of model behavior.

Overall, accuracy remains part of the evaluation because it provides a general measure of how many elements are classified correctly. In the stored evaluation outputs, this usually appears as acc. However, the project does not interpret accuracy in isolation. In a multiclass setting with dominant and minority classes, a model may obtain a relatively strong accuracy value while still performing poorly on smaller but structurally important categories.

For this reason, the evaluation workflow gives particular importance to mean Intersection over Union in its presence-aware form, namely ***mIoU_present***. In practice, the project uses a support threshold such as ***present_min_support = 50***, which means that classes represented by fewer than 50 points in a tile do not distort the per-tile present-class average. This choice prevents the comparative analysis from being dominated by unstable ***IoU*** values associated with extremely rare class occurrences. As a result, ***mIoU_present*** becomes one of the most informative metrics in the project for assessing balanced segmentation quality.

In addition to OA and ***mIoU_present***, the evaluation workflow also records operational indicators related to prediction completeness and scale. Coverage-related values such as ***coverage*** and ***drop_frac*** make it possible to verify whether a model prediction meaningfully covers the target tile. Scale indicators such as ***raw_points*** and ***vox_points*** help contextualize the computational dimension of the evaluated sample. This is especially useful in the sparse-

convolution setting, where voxelization changes the effective representation size and must therefore be considered when interpreting evaluation behavior.

6.4.3 COMPARATIVE EVALUATION PROCEDURE

The final layer of the evaluation workflow consists of consolidating model-specific outputs into a comparative structure that supports later analysis and selection. This is necessary because the project is not based on one model and one run, but on a set of architectures evaluated across several scene-specific splits and repeated seeds.

To support this, the project writes consolidated evaluation outputs to structured summary files under ***outputs/summary/.*** These files store the relevant metrics for different runs, tiles, and configurations, making it possible to compare model families under shared criteria. Representative examples of this summary layer include files such as ***plateau_val_present50_ABC_ALLMODELS.csv***, as well as model-selection artifacts such as ***best_plateau_ckpts_present50.txt***, ***best_urban_ckpts_present50.txt and best_rural_ckpts_present50.txt***. Their purpose is not to interpret the results yet, but to organize them so that later interpretation becomes methodologically grounded.

The comparative evaluation procedure therefore closes the loop between model implementation, training management, and downstream analysis. It transforms raw checkpoint outputs into a coherent set of comparable evidence.

6.5 ANOMALY-ORIENTED EXTENSION OF THE SYSTEM

Once the segmentation framework is established and the evaluation workflow is defined, the system incorporates an anomaly-oriented extension designed to move from semantic prediction toward inspection-oriented interpretation. The purpose of this extension is to build upon it to identify suspicious spatial regions, rank them according to their anomaly evidence, and generate interpretable visual outputs for qualitative validation.

Semantic segmentation provides a structured description of the point cloud, but maintenance-oriented analysis requires an additional layer capable of highlighting where potentially problematic behavior concentrates. For this reason, the developed system transforms model-derived outputs into region-level anomaly indicators that can support technical inspection and later discussion in the results chapter.

In practical terms, the anomaly-oriented workflow is applied on top of the strongest-performing **RandLA-Net configurations**, rather than exhaustively across every trained model family. This decision follows the comparative logic established in the previous sections. RandLA-Net offers a convincing balance between segmentation quality, scalability, and repeatability, which makes it the most suitable backbone for the downstream anomaly stage. As a result, the anomaly-oriented analysis remains methodologically connected to the broader segmentation workflow while avoiding unnecessary dispersion across weaker or less operationally convenient configurations.

The anomaly outputs are organized into dedicated folders under ***outputs/anomaly/,*** including summary-level artifacts and model-specific region visualizations. This means that the anomaly stage is treated as an integrated layer of the overall system. It produces ranked tables, region descriptors, and overlay figures that remain structurally linked to the corresponding segmentation runs.

Output	Implementation in the project	Main purpose
Global anomaly summaries	outputs/anomaly/summary_global/	To consolidate anomaly information across processed runs
Ranked tile summaries	top_tiles.csv	To identify the tiles with the strongest anomaly evidence
Global ranking figures	top_tiles_priority.png	To provide a quick visual summary of tile-level anomaly priority
Region extraction script	extract_severe_regions_randla.py	To detect and delimit suspicious spatial regions from anomaly signals
Overlay script	make_region_overlay_figures_randla.py	To create full-context and zoomed visualizations of selected regions
Model-specific anomaly folders	outputs/anomaly/randla/<scene_split_seed>/	To store region-level figures and intermediate outputs for each selected run
Region interval	x_start, x_end, length_m	To localize each suspicious region along the analyzed tile
Severity composition	n_severe_bins, n_warning_bins, dominant_severity	To characterize the internal intensity profile of the region
Anomaly scoring	score_max, dominant_anomaly_type, region_priority	To rank regions according to their maximum evidence and overall priority

Table 7: Main anomaly-analysis outputs and region descriptors

As shown in Table 7, the anomaly-oriented extension produces both numerical descriptors and visual artifacts. This shows that system aims to represent it in a way that can later be interpreted from an inspection perspective.

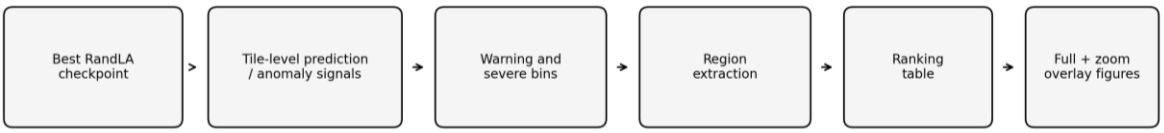

Figure 24: Anomaly-oriented extension of the segmentation workflow

As illustrated in Figure 24, the anomaly-oriented extension operates as a downstream layer of the segmentation framework, transforming model-derived local evidence into ranked regions and interpretable visual outputs.

6.5.1 COMPUTATION OF ANOMALY INDICATORS

The first technical step in the anomaly-oriented extension consists of computing indicators that summarize suspicious behavior at a local level. Instead of treating the segmentation output as a flat and homogeneous prediction map, the system derives numerical signals that reflect the concentration and intensity of anomalous evidence over localized spatial units.

This design is important because isolated point-level irregularities are not always informative on their own. In large point clouds, local ambiguity, prediction noise, or small structural variations may produce scattered signals that do not necessarily correspond to meaningful inspection targets. The anomaly-oriented pipeline therefore focuses on summarized local evidence rather than on isolated point anomalies.

The implementation retains several region-relevant descriptors from the beginning. These include variables such as ***score_max***, which captures the maximum anomaly evidence associated with a candidate region, as well as the number of bins assigned to stronger or

weaker alert categories, represented by ***n_severe_bins*** and ***n_warning_bins***. The system also assigns descriptors such as ***dominant_severity*** and ***dominant_anomaly_type***, which help characterize the prevailing nature of the suspicious behavior observed in each region.

This indicator logic allows the anomaly workflow to distinguish between regions with different internal profiles. A region dominated by severe evidence is not interpreted in the same way as one composed mainly of warning-level signals, even if both appear in the ranking. In other words, the system preserves a richer representation than a simple binary anomaly flag. This is useful both for prioritization and for later visual interpretation.

6.5.2 SPATIAL GROUPING AND PRIORITIZATION OF SUSPICIOUS REGIONS

Once local anomaly indicators are computed, the next step consists of grouping them spatially into coherent candidate regions. This is one of the most important parts of the anomaly-oriented extension, because inspection workflows do not operate naturally on isolated points or disconnected local signals. They operate on segments, areas, or localized zones of interest.

For this reason, the developed system transforms local anomaly evidence into region-level entities. Each candidate region is associated with an interval, represented by variables such as ***x_start***, ***x_end***, and ***length_m***, which make it possible to delimit the suspicious segment spatially.

Once grouped, the regions are prioritized according to their anomaly evidence. The ranking logic incorporates both intensity and spatial coherence. A region with a strong ***score_max***, a substantial number of severe bins, and a consistent local extent is treated as more relevant than a small-fragmented zone with weak or unstable evidence. The resulting priority is summarized through descriptors such as ***region_priority***, which support the construction of ranked summary tables.

This ranking stage is operationally important because it reduces a potentially large amount of low-level anomaly information into a manageable set of candidate regions. Instead of forcing the analyst to examine raw local signals directly, the system presents a prioritized structure that highlights the most relevant suspicious areas first.

The outputs of this stage are stored in summary-level artifacts such as ***global_anomaly_summary.csv***, ***top_tiles.csv***, and other ranking-oriented files under ***outputs/anomaly/summary_global/***. In this way, region prioritization becomes not only a conceptual step but also a persistent and reproducible output of the developed system.

6.5.3 VISUAL VALIDATION THROUGH OVERLAY FIGURES

The final stage of the anomaly-oriented extension consists of visual validation through point-cloud overlay figures. This stage is necessary because anomaly ranking alone is not sufficient to support a convincing technical interpretation. Numerical priority is useful, but the suspicious regions also need to be examined in their geometric context to verify whether the detected patterns are spatially coherent and visually plausible.

To support this validation, the system generates dedicated figures through the script ***make_region_overlay_figures_randla.py***. These figures are stored in model-specific folders under ***outputs/anomaly/randla/<scene_split_seed>/*** and are designed to show the selected regions in a format that combines context and detail. The visual layout includes a broader panel that situates the suspicious region within the full tile or local scene, together with a zoomed panel that makes the concentration of anomaly evidence easier to inspect.

This two-level representation links ranking with spatial interpretation. A suspicious region is easier to understand when the reader can see both where it lies in the overall point cloud and how the highlighted evidence behaves locally. For this reason, the overlay figures are not decorative additions but a core component of the anomaly-oriented pipeline.

The qualitative role of these figures is also methodologically relevant. Since the anomaly stage does not rely on exhaustive annotated ground truth for every possible fault scenario, visual validation provides an essential complementary mechanism for assessing whether the ranked regions correspond to coherent spatial patterns rather than arbitrary numerical artifacts. This does not replace quantitative analysis, but it strengthens the interpretability of the anomaly outputs.

At the dissertation level, these figures also serve a second purpose: they generate presentation-ready artifacts that support the discussion developed in Chapter 7. Chapter 7. In the context of the system-development chapter, however, their main role is to show that the pipeline is able not only to compute anomaly descriptors but also to render them in a form suitable for human inspection.

6.6 INTEGRATED WORKFLOW OF THE DEVELOPED SYSTEM

The system developed in this project operates as an integrated workflow that connects data preparation, semantic segmentation, controlled experimentation, evaluation, and anomaly-oriented interpretation within a single reproducible framework. Rather than functioning as a collection of isolated scripts, the different implementation blocks remain structurally linked, so that the output of each stage becomes the input of the next one.

The workflow starts with the inspection and standardization of the raw railway point clouds and continues with the generation of derived train and validation subsets organized under a consistent experimental structure. Once these inputs are prepared, the system feeds them into several segmentation model families, each one implemented through its own training and inference routines but integrated into the same comparative framework. The resulting checkpoints, logs, and summary outputs are then processed through a common evaluation layer that produces tile-level predictions and comparable performance indicators.

On top of this segmentation framework, the system incorporates an anomaly-oriented extension that uses the strongest-performing configurations as the basis for downstream analysis. This extension computes local anomaly indicators, groups suspicious evidence into spatial regions, ranks those regions according to their severity and priority, and generates overlay figures that support qualitative inspection. In this way, the workflow moves from raw point-cloud data to interpretable outputs that are suitable not only for segmentation benchmarking but also for inspection-oriented analysis.

A relevant characteristic of this integrated design is that it preserves methodological continuity across the whole project. The anomaly-oriented stage does not operate independently from the segmentation workflow, but directly depends on the same data organization, the same trained models, and the same evaluation logic. Similarly, model comparison is not treated as a disconnected analytical task, but as part of a broader system in which data preparation, execution control, output persistence, and downstream interpretation remain coordinated.

From a technical point of view, this integrated workflow is one of the main contributions of the project. The work does not consist only of implementing individual architectures or generating visual outputs in isolation. Instead, it delivers a structured pipeline in which preprocessing, model execution, evaluation, and anomaly analysis support each other within a common operational framework. This is important in a CPU-only environment, where reproducibility, efficiency, and disciplined experiment management are essential for feasibility.

The implementation described in this chapter provides the operational basis for the experimental analysis presented in the following chapter, where the behavior of the different models and the anomaly-oriented outputs are examined in detail.

Chapter 7. RESULTS ANALYSIS

This chapter presents the experimental results obtained with the framework developed in the previous chapter. The analysis is organized in two main stages. First, the semantic segmentation performance of the implemented model families is examined across the considered railway scenes and experimental configurations. Second, the anomaly-oriented extension built on top of the selected backbone is analyzed through ranked suspicious regions and qualitative visual validation.

7.1 SEGMENTATION RESULTS ACROSS MODEL FAMILIES

The first part of the analysis focuses on the semantic segmentation results obtained across the implemented model families, beginning with the simplest per-point baseline and progressing toward more expressive point-based and sparse-convolution architectures. These results provide the basis for understanding how different architectural choices behave on railway point-cloud scenes and, later, for selecting the most suitable backbone for anomaly-oriented analysis.

The comparison places particular emphasis on the mean Intersection over Union computed under a present-class support threshold, denoted in the workflow as ***mIoU_present50***. In practice, this metric averages ***IoU*** values only over classes that reach a minimum support level in the evaluated tile, which makes it especially informative in the present context, where railway point clouds contain minority classes and thin overhead elements whose representation varies from tile to tile.

Before presenting the final validation comparison, Table 8 summarizes the training workload and convergence behaviour of the main model families. This table is included to make explicit the scale of the experimental campaign, since the comparison was not based on

isolated executions but on repeated runs across railway scenarios, spatial splits, and random seeds. The table reports the number of main runs, the train/validation cases loaded during training, the number of epochs, the average evolution of the training loss, and the best validation mIoU monitored during training.

Model family	**Scene**	**Runs**	**Train / Val per run**	**Epochs**	**Mean train loss start → final**	**Best val. mIoU**
PN2-lite focal	Plateau	9	7–8 / 1	30	0.387 → 0.152	0.274 ± 0.036
PN2-lite focal	Rural	9	9 / 1	30	0.957 → 0.351	0.205 ± 0.019
PN2-lite focal	Urban	9	9 / 1	30	1.027 → 0.476	0.159 ± 0.072
PNPP-lite focal	Plateau	9	7–8 / 1	30	0.443 → 0.125	0.291 ± 0.029
PNPP-lite focal	Rural	9	9 / 1	30	1.010 → 0.253	0.301 ± 0.022
PNPP-lite focal	Urban	9	9 / 1	30	1.060 → 0.426	0.156 ± 0.069
RandLA-lite focal	Plateau	9	8–9 / 1–4	30	7.149 → 0.104	0.632 ± 0.033
RandLA-lite focal	Rural	9	32 / 16	30	1.752 → 0.155	0.574 ± 0.060

RandLA-lite focal	Urban	9	21–22 / 10–11	30	0.908 → 0.171	0.583 ± 0.040
Minkowski CNN	Plateau	9	8–9 / 1–4	35	0.929 → 0.033	0.856 ± 0.050
Minkowski CNN	Rural	9	32 / 16	35	1.011 → 0.093	0.800 ± 0.015
Minkowski CNN	Urban	9	21–22 / 10–11	35	1.273 → 0.248	0.676 ± 0.032

Table 8: Training workload and convergence summary across model families

The training summary shows that the experimental campaign involved a substantial number of controlled executions. Each main model family was evaluated across three railway scenarios, three spatial splits, and three random seeds. The loss evolution confirms that the trainable architectures converged during training, especially RandLA-lite and Minkowski CNN. However, the monitored validation scores during training should not be interpreted as the final comparable result, since they correspond to the internal validation process of each training run. For this reason, the final comparison is reported separately in Table 9 using the common mIoU_present50 validation protocol.

The following Table 9 summarizes the mean validation performance obtained across scenes and model families using ***mIoU_present50*** as the main balanced segmentation metric.

Scene	Baseline MLP mean ± std	PN2-lite focal mean ± std	PNPP-lite focal mean ± std	RandLA-lite focal mean ± std	Minkowski CNN
Plateau	0.0015 ± 0.0017	0.199 ± 0.037	0.242 ± 0.046	**0.464 ± 0.062**	0.297 ± 0.057
Rural	0.0024 ± 0.0013	0.179 ± 0.030	0.268 ± 0.018	**0.647 ± 0.053**	0.425 ± 0.054
Urban	0.0004 ± 0.0002	0.107 ± 0.013	0.130 ± 0.010	**0.577 ± 0.042**	0.175 ± 0.043

Table 9: Mean validation performance by scene and model family (mIoU_present50)

As shown in Table 9, the comparative structure of the results is already clear at scene level. The baseline MLP remains far below the other architectures in all three scenes, which confirms that the task cannot be addressed satisfactorily through simple per-point feature learning alone. The two lightweight point-based models improve substantially over that lower bound, with ***pnpp_lite_focal*** consistently outperforming ***pn2_lite_focal***. The ***Minkowski CNN*** also provides a clear improvement over the MLP baseline and obtains competitive results in the plateau and urban scenes, particularly in ***rural***, where it reaches a mean ***mIoU_present50*** of 0.425 ± 0.054. Overall, the best consolidated performance corresponds to ***randla_lite_focal***, which remains clearly ahead of the other evaluated architectures in plateau, rural, and urban environments.

The scene also indicates that the benchmark is heterogeneous. Rural produces the strongest values for the best-performing families, while plateau and urban remain more demanding, particularly for the simpler models. This scene dependence is important because it shows that the thesis does not evaluate the models on a single homogeneous railway context, but on structurally distinct environments with different segmentation difficulty levels.

7.1.1 BASELINE PER-POINT MLP

The simplest reference model in the experimental comparison is the baseline per-point MLP. Its role in the results analysis is not to compete directly with the strongest architectures, but to establish a lower-bound segmentation reference against which the gains of more expressive models can be interpreted.

Although its numerical performance remains extremely limited, the baseline is still methodologically useful because it makes the difficulty of the task explicit and provides a lower-bound reference for the later architectural gains. In this sense, its value lies less in its absolute segmentation quality than in its ability to anchor the benchmarking progression of the chapter.

The consolidated validation results confirm that this baseline remains extremely limited for the present task. Its mean ***mIoU_present50*** reaches only **0.0015** in plateau, **0.0024** in rural, and **0.0004** in urban. These values are several orders of magnitude below those obtained by the stronger point-based configurations and by RandLA-Net. The same pattern appears across splits. In plateau, the split means range from approximately **0.0009** to **0.0018**. In rural, they range from **0.0006** to **0.0035**. In urban, they remain between approximately **0.0004** and **0.0005**.

This behavior shows that the railway point-cloud segmentation problem cannot be solved meaningfully through isolated point-wise feature learning alone. The overhead elements of interest are thin, spatially structured, and strongly dependent on local geometric relations. A model that processes points mainly as independent feature vectors does not capture that structure with sufficient quality.

Even so, the baseline remains methodologically relevant. It confirms that the task is not artificially easy and that the gains observed in later architectures are not marginal refinements but substantial improvements over a very weak point-wise reference.

7.1.2 LIGHTWEIGHT POINT-BASED MODELS

Beyond the per-point MLP baseline, the project evaluates two lightweight point-based configurations, ***pn2_lite_focal*** and ***pnpp_lite_focal***, which provide a stronger representation of local geometry while remaining substantially lighter than RandLA-Net or sparse-convolution alternatives.

Among them, ***pnpp_lite_focal*** consistently outperforms ***pn2_lite_focal*** across the three scenes. In plateau, the mean ***mIoU_present50*** rises from **0.199** to **0.242**. In rural, it increases from **0.179** to **0.268**. In urban, it improves from **0.107** to **0.130**. This scene-wise consistency suggests that the richer local abstraction of the PNPP-style configuration provides a measurable advantage over the simpler PN2-lite alternative.

The split-level results reinforce this interpretation. In plateau, ***pnpp_lite_focal*** reaches its best value in **splitC seed43** with **0.330**, while ***pn2_lite_focal*** reaches its best plateau value in **splitC seed42** with **0.279**. In rural, the best PNPP result is **0.290** in **splitC seed44**, compared with a best PN2 value of **0.227** in **splitC seed42**. In urban, the strongest PNPP result reaches **0.150** in **splitA seed43**, while the best PN2 value reaches **0.121** in **splitA seed42**.

These values show that the lightweight point-based models capture a relevant portion of the railway geometry, but not enough to deliver strong segmentation quality on thin overhead structures and heterogeneous scene context. Their contribution is therefore mainly comparative. They demonstrate that local abstraction matters, since both improve dramatically over the baseline MLP, but they also show that lightweight point-based processing alone remains insufficient to match the stronger family-level performance of RandLA-Net.

7.1.3 RANDLA-NET

RandLA-Net provides the best consolidated segmentation performance among the currently available validation results and emerges as the most competitive family across the three scenes. Its mean ***mIoU_present50*** reaches **0.464 ± 0.062** in plateau, **0.647 ± 0.053** in rural, and **0.577 ± 0.042** in urban, placing it clearly above the baseline MLP and the two lighter point-based alternatives in every case.

This advantage is visible not only in the aggregated scene means, but also in the split-level behavior. In plateau, RandLA-Net reaches split means of **0.423** in splitA, **0.510** in splitB, and **0.458** in splitC. In rural, the split means rise to **0.639**, **0.667**, and **0.636** across splitA, splitB, and splitC, respectively. In urban, the corresponding values are **0.603**, **0.605**, and **0.523**. This consistency across split means shows that the superiority of the family does not depend on a single accidental validation partition.

The seed-level values also reinforce the strength of the family. In plateau, the best RandLA result is **0.525** in **splitB seed42**, and the three runs in that split remain close at **0.525**, **0.505**, and **0.500**. In rural, the strongest value reaches **0.706** in **splitC seed43**, while several other runs also remain very high, including **0.687** in **splitB seed42** and **0.684** in **splitB seed43**. In urban, the best result reaches **0.616** in **splitA seed43**, with additional strong values such as **0.614** in **splitB seed44** and **0.604** in **splitB seed43**.

Taken together, these results indicate that RandLA-Net provides not only the highest available segmentation quality, but also a convincing degree of repeatability across related experimental settings. This balance between performance and operational consistency becomes decisive in the later selection of the backbone used for anomaly-oriented analysis.

7.1.4 MINKOWSKI CNN

The sparse-convolution benchmark based on the Minkowski framework is incorporated into experimental analysis as the most advanced three-dimensional convolutional family considered in this thesis. Its inclusion is relevant because it complements the point-based models with a sparse-tensor representation, which is theoretically well suited to large-scale railway point clouds and to the modelling of local three-dimensional spatial context.

The Minkowski CNN achieves a mean ***mIoU_present50*** of **0.297 ± 0.057** in plateau, **0.175 ± 0.043** in rural, and **0.425 ± 0.054** in urban. These results place the model clearly above the baseline MLP in all three scenes, confirming that sparse three-dimensional convolutional processing provides a substantial improvement over simple per-point feature learning. However, its behavior is more heterogeneous than that observed for RandLA-Net.

In plateau, Minkowski CNN reaches a competitive mean value of **0.297 ± 0.057**. This result is higher than those obtained by ***pn2_lite_focal*** and ***pnpp_lite_focal***, but remains below ***RandLA-Net***. At split level, the model obtains mean values of **0.267** in **splitA**, **0.285** in **splitB**, and **0.338** in **splitC**. The best individual result in this scene is obtained in **splitC seed44**, with a ***mIoU_present50*** of **0.413**. Other relatively strong values include **0.329** in **splitA seed43** and **0.324** in **splitC seed42**.

In rural, the performance of Minkowski CNN is more limited. The overall mean decreases to **0.175 ± 0.043**, placing it close to ***pn2_lite_focal*** but below ***pnpp_lite_focal*** and clearly below ***RandLA-Net***. The split-level means are **0.174** in **splitA**, **0.188** in **splitB**, and **0.162** in **splitC**. The best rural value is **0.225** in **splitA seed43**, followed by **0.212** in **splitB seed43** and **0.203** in **splitB seed44**. This suggests that, under the current configuration, the additional representational capacity of sparse convolutions does not translate into a consistent advantage in the rural environment.

In urban, Minkowski CNN shows its strongest relative behavior. The model reaches **0.425 ± 0.054**, clearly outperforming both lightweight point-based alternatives and reducing the gap with ***RandLA-Net***. The corresponding split means are **0.434** in **splitA**, **0.475** in **splitB**,

and **0.367** in **splitC**. The best individual result is obtained in **splitB seed43**, with a ***mIoU_present50*** of **0.503**. Additional strong values are observed in **splitA seed43**, with **0.472**, and **splitB seed44**, with **0.470**. This indicates that the sparse voxelized representation is particularly effective in the urban scene, where the spatial structure appears to be better exploited by the model.

From a methodological perspective, the Minkowski benchmark validates a more structurally expressive 3D family within the same evaluation framework. The sliding-window evaluation strategy ensures full validation coverage, with mean ***coverage*** equal to **1.0** and ***mean drop*** fraction equal to **0.0** across the evaluated runs. Therefore, the reported results are not affected by the low-coverage artefacts observed in earlier evaluation attempts.

7.2 COMPARATIVE DISCUSSION OF SEGMENTATION PERFORMANCE

The comparative pattern emerging from the validation results follows a progressive logic. The per-point MLP baseline provides the simplest reference level, the lightweight point-based models improve substantially upon that reference through richer local abstraction, and ***randla_lite_focal*** clearly delivers the strongest consolidated performance among the currently available point-based results. The Minkowski CNN extends this comparison with a sparse-convolutional 3D representation, achieving competitive results especially in the urban scene, although it remains below ***randla_lite_focal*** across the three evaluated environments.

One of the clearest findings is the magnitude of the gap between the baseline MLP and the rest of the benchmark. The MLP mean ***mIoU_present50*** remains close to zero in the three scenes, which confirms that independent point-wise processing is insufficient for the present railway segmentation problem. This makes the later improvements not only statistically

relevant but conceptually meaningful, since they reflect the importance of local geometric structure in catenary-related scene understanding.

A second important finding is the consistent advantage of ***pnpp_lite_focal*** over ***pn2_lite_focal***. This suggests that more expressive local abstraction mechanisms provide measurable gains even within lightweight families. However, the gap between these models and RandLA-Net remains much larger than the gap between the two lightweight variants themselves. In plateau, RandLA exceeds PNPP by approximately **0.222** in mean ***mIoU_present50***. In rural, the difference rises to approximately **0.379**. In urban, it remains around **0.447**. These are not marginal improvements, but large performance margins.

The different scenes also mean that the benchmark is heterogeneous. Rural appears to be the most favorable scene for the stronger families, while urban remains the most challenging for the simpler ones and plateau for the most difficult ones.

At the same time, the RandLA results are not interpreted only through their best isolated values. Their split means and several seed-level runs remain consistently strong, particularly in ***plateau splitB***, ***rural splitB-C***, and ***urban splitA-B***. This matters because the anomaly-oriented extension requires a backbone that is not only accurate, but also stable enough to support downstream analysis without depending on a single favorable run.

The Minkowski CNN adds an additional perspective to this comparison because it introduces a sparse-convolutional representation rather than a purely point-based one. Its results confirm the value of structured 3D processing, since it clearly improves over the MLP baseline and performs competitively in plateau and especially in urban, where it reaches ***0.425 ± 0.054***. However, its behavior is less consistent across scenes than RandLA-Net. In rural, its performance drops to ***0.175 ± 0.043***, remaining below both PNPP-lite and RandLA-Net. Therefore, Minkowski CNN strengthens the benchmark by validating a more expressive 3D family, but it does not alter the overall model ranking.

For these reasons, the comparative discussion points clearly toward RandLA-Net as the strongest currently available family for the thesis workflow. The baseline MLP and the

lightweight point-based models remain useful references and demonstrate the importance of local representation, but they do not provide the same segmentation quality or the same confidence for downstream anomaly-oriented use. The Minkowski CNN is also a valuable reference because it demonstrates the potential of sparse-convolutional 3D processing, but its heterogeneous behavior across scenes makes it less suitable than RandLA-Net as the final backbone for the anomaly-oriented stage.

7.3 SELECTION OF THE MOST SUITABLE BACKBONE FOR ANOMALY-ORIENTED ANALYSIS

The anomaly-oriented extension of the system requires a segmentation backbone that is not only accurate, but also sufficiently stable and operationally manageable. For this reason, the model used in the anomaly stage is not selected solely based on a single best run. Instead, the selection follows the broader comparative logic established in the previous sections, combining segmentation quality, repeated performance across seeds, and practical suitability for downstream analysis.

Within the currently consolidated validation results, ***randla_lite_focal*** satisfies these conditions. It provides the strongest mean performance in the three scenes, maintains competitive split-level behavior, and yields several high-quality checkpoints that can be selected in a principled way. This makes it the most appropriate backbone for the anomaly-oriented stage of the thesis.

The following Table 10 summarizes the strongest RandLA-Net validation checkpoints selected as reference configurations for downstream anomaly-oriented analysis in each scene.

Scene	Model	Split	Seed	Mean mIoU_present50	Checkpoint
Plateau	randla_lite_focal	splitB	42	0.525	outputs/checkpoints/randla_lite_plateau_splitB_focal_seed42.pt
Rural	randla_lite_focal	splitC	43	0.706	outputs/checkpoints/randla_lite_rural_splitC_focal_seed43.pt
Urban	randla_lite_focal	splitA	43	0.616	outputs/checkpoints/randla_lite_urban_splitA_focal_seed43.pt

Table 10: Selected best validation RandLA checkpoints used as reference backbones

As shown in Table 10, the selected RandLA checkpoints correspond to the strongest validation configurations identified in each scene: **plateau splitB seed42**, **rural splitC seed43**, and **urban splitA seed43**. These selections are fully consistent with the broader comparative pattern observed in the previous section and provide a technically justified basis for anomaly-oriented processing.

This selection strategy preserves methodological continuity. The anomaly stage does not operate independently from the segmentation benchmark but is built directly on the strongest-performing configurations identified through the validation workflow. At the same time, the choice of RandLA-Net is not merely opportunistic. It emerges from a consistent family-level advantage across scenes rather than from a single isolated checkpoint.

7.4 END-TO-END CASE STUDY OF THE PROPOSED WORKFLOW

To complement the global quantitative analysis, this section presents an end-to-end case study that illustrates how the proposed workflow operates on a specific railway point-cloud scene, from the processed input data to the final anomaly-oriented result. The selected case corresponds to the urban scene ***L10-1-M01-002.ply***, evaluated under the **urban_splitA_seed43** configuration. This case is particularly representative because it belongs to an urban environment, where the railway corridor contains a dense and complex spatial context, including surrounding structures, vegetation, and thin overhead-contact-line-related elements.

The selected scene is part of the validation subset of the urban split A configuration. This means that the point cloud is not used to fit the model parameters during training, but to assess the behaviour of the trained model on unseen data within the same urban scenario. The selected backbone for this case is the RandLA-lite focal model trained under the **urban_splitA_seed43** run, whose corresponding checkpoint is ***outputs/checkpoints/randla_lite_urban_splitA_focal_seed43.***pt. This run achieves a validation ***mIoU_present50*** of **0.616** in the urban validation benchmark, which makes it one of the strongest urban configurations and therefore a suitable candidate for the anomaly-oriented stage.

The original scene is a large-scale mobile LiDAR point cloud. In the preprocessing stage, the input ***.ply*** file is standardized and prepared according to the common data pipeline used throughout the project. This includes semantic label handling, feature normalization, spatial organization, and the preparation of the scene for tile-level inference. The scale of the selected case is relevant: the scene contains more than 18 million raw points, which illustrates why direct full-scene processing is not practical under the computational constraints of the project. For sparse-processing experiments, the same scene is reduced to approximately 3.88 million voxelized points, showing the importance of spatial reduction and structured representation in the overall workflow.

After preprocessing, the trained RandLA-lite model is applied to the selected validation scene. The model produces a semantic segmentation output in which each point is assigned to one of the railway-related semantic classes. This segmentation output is the basis for the subsequent anomaly-oriented analysis. Instead of applying anomaly detection directly to the raw point cloud, the system first uses the semantic prediction to focus the analysis on relevant railway infrastructure components and their spatial behaviour. This is important because the target elements associated with catenary and overhead-contact-line structures are sparse, thin, and geometrically sensitive.

Once the semantic segmentation is obtained, the anomaly-oriented module computes local indicators over the predicted railway components. In this case, the selected suspicious region is identified as a gap-type anomaly candidate. A gap indicates a local discontinuity or absence of points in a structure that is expected to show spatial continuity. The detected candidate is located in the interval **$x \in [-122.1, -119.1]$** m, with a length of 3.0 m. The region is assigned a severe severity level, with a maximum anomaly score of 5.0 and a region priority score of 15.0. These values make the region relevant for visual inspection and qualitative interpretation. The following Table 11 summarizes the selected case study.

Field	Value
Scene	L10-1-M01-002.ply
Railway environment	Urban
Run	urban_splitA_seed43
Backbone	RandLA-lite focal
Checkpoint	outputs/checkpoints/randla_lite_urban_splitA_focal_seed43.pt
Run validation metric	mIoU_present50 = 0.616
Region ID	03
Anomaly type	Gap
Severity	Severe
Spatial interval	$x \in [-122.1, -119.1]$ m
Region length	3.0 m
Maximum anomaly score	5.0
Region priority	15.0

Table 11: Summary of the selected end-to-end case study

The result is shown in Figure 25. The left panel provides the full context of the railway scene, where the candidate region is interpreted within the surrounding point-cloud structure. The right panel zooms into the suspicious interval and highlights the spatial area where the gap indicator is activated. This two-level visualization avoids interpreting the anomaly as an

isolated numerical output. Instead, the result is placed back into its geometric context, allowing the user to inspect whether the candidate region corresponds to a meaningful discontinuity in the railway infrastructure.

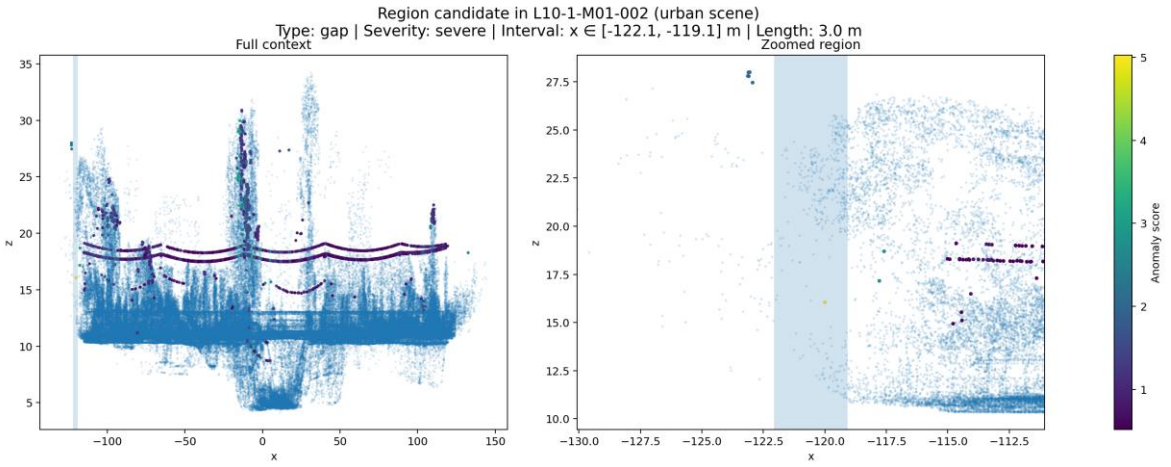

Figure 25: End-to-end anomaly-oriented case study for the urban scene L10-1-M01-002.ply

The case study shows how the different stages of the proposed system are connected in practice. The workflow starts from a large-scale railway point cloud, applies preprocessing and scene organization, uses a trained semantic segmentation backbone, computes anomaly-oriented indicators over the predicted railway components, ranks suspicious regions according to severity and priority, and finally generates an interpretable visual output. Therefore, the example demonstrates that the developed system is not limited to reporting aggregate segmentation metrics but can produce spatially localized and visually inspectable anomaly candidates.

It is important to emphasize that the detected region is interpreted as a **prioritized suspicious candidate**, not as a confirmed infrastructure defect. The dataset does not provide explicit ground truth annotations for real catenary failures or maintenance defects. For this reason, the anomaly-oriented output is used as a decision-support layer that highlights regions requiring further expert inspection. Within this interpretation, the selected case illustrates the practical value of combining semantic segmentation with geometric anomaly indicators: the

segmentation model structures the railway scene, while the anomaly module identifies local regions whose spatial behaviour deviates from the expected continuity of overhead-related components.

7.5 ANOMALY-ORIENTED RESULTS AND REGION PRIORITIZATION

Once the segmentation backbone is selected, the next step consists of analyzing the anomaly-oriented outputs produced by the downstream pipeline. This stage is applied to the strongest RandLA-Net configurations selected in the previous section, namely ***plateau_splitB_seed42, rural_splitC_seed43,*** and ***urban_splitA_seed43***. The objective is not merely to flag isolated anomalous points, but to identify suspicious tiles, extract localized candidate regions, and rank them according to their severity and anomaly evidence.

At tile level, the ranking already reveals a non-uniform distribution of anomaly evidence across scenes. The highest priority tile is ***urban_splitA_seed43 / L6-1-M01-001***, with a priority score of **110.0**, followed very closely by ***plateau_splitB_seed42 / Tibet-12*** with **107.5**. The next most prominent cases are again urban, including ***L6-1-M01-002*** with **98.0**, **L14-1-M01-003** with **90.0**, and ***L17-1-M01-003*** with **89.0**. By contrast, the rural scene contributes a broader set of moderate-priority tiles, led by ***section048*** with **69.0**, followed by several additional sections in the **20–34** range.

This distribution shows that the anomaly-oriented stage captures different behaviors depending on the scene. The **urban** scene dominates the upper part of the tile ranking, which suggests that it concentrates several high-priority suspicious segments. The **rural** scene contributes a larger number of moderate candidates, which indicates a broader but less extreme pattern of anomaly evidence. The **plateau** scene appears mainly through a single outstanding tile, **Tibet-12**, whose anomaly behavior is especially intense but also numerically singular.

The plateau case deserves careful reading. In ***Tibet-12***, the summary shows only **6 bins of target presence**, but **78 anomalous bins**, an anomaly ratio of **13.0**, and an extremely high score_max of **523.49**. This makes it the most extreme anomaly signal in the dataset, but it also indicates that the ratio-based interpretation must be handled cautiously because the target-supported segment is very short. For this reason, ***Tibet-12*** is best interpreted as a highly concentrated outlier rather than as evidence that the plateau scene is globally more anomalous than the others.

The most operationally useful representation, however, is not the tile ranking alone but the **region-level prioritization**. After spatial grouping, the pipeline extracts compact candidate regions and ranks them according to ***score_max***, severity composition, and regional priority. The resulting top-region list is especially informative because it converts broad tile-level anomaly evidence into short, localized intervals that can be inspected visually.

The following Table 12 and Table 13**¡Error! No se encuentra el origen de la referencia.** summarizes the top-ranked anomaly regions extracted from the selected RandLA-Net runs and used later for qualitative validation.

Region ID	Run	Tile	Interval (m)	Length (m)
01	plateau_splitB_seed42	Tibet-12	-139.5 to -137.5	2.0
02	plateau_splitB_seed42	Tibet-12	-214.5 to -211.5	3.0
03	urban_splitA_seed43	L10-1-M01-002	-122.1 to -119.1	3.0
04	urban_splitA_seed43	L15-1-M01-002	75.4 to 78.4	3.0
05	rural_splitC_seed43	section025	-52.2 to -50.2	2.0
06	rural_splitC_seed43	section061	59.5 to 61.5	2.0
07	rural_splitC_seed43	section046	69.3 to 71.3	2.0
08	rural_splitC_seed43	section033	58.9 to 60.9	2.0
09	rural_splitC_seed43	section029	-60.8 to -57.8	3.0
10	urban_splitA_seed43	L14-1-M01-003	-113.0 to -111.0	2.0
11	urban_splitA_seed43	L10-1-M01-001	101.6 to 103.6	2.0
12	rural_splitC_seed43	section015	-46.4 to -44.4	2.0

Table 12: Identification and location of anomalous regions

Region ID	Anomaly type	Severity	Score max	Region priority
01	height_jump	severe	523.5	268.7
02	gap	severe	260.4	142.7
03	gap	severe	5.0	15.0
04	height_jump	severe	8.1	14.5
05	uncertainty	severe	5.3	9.6
06	height_jump	severe	3.7	8.8
07	height_jump	severe	3.7	8.8
08	height_jump	severe	3.6	8.8
09	uncertainty	severe	2.2	8.6
10	height_jump	severe	3.2	8.6
11	height_jump	severe	3.1	8.5
12	uncertainty	severe	2.3	8.2

Table 13: Anomaly characterization and priority

Table 12 and Table 13 reveals several important patterns. First, the two strongest regions by a wide margin belong to the **plateau** scene and both come from the same tile, ***Tibet-12***. This means that a single tile can contain more than one high-priority candidate region and that the pipeline can distinguish them spatially. Second, the **urban** scene contributes to several compact severe regions with priorities between approximately **8.5** and **15.0**, indicating that the upper part of the anomaly ranking is not driven only by one isolated extreme outlier.

Third, the **rural** scene contributes a coherent group of regions with slightly lower but still consistent priorities, most of them in the **8–10** range.

Another relevant finding is that the extracted top regions are not homogeneous in anomaly type. The ranking includes ***gap***, ***height_jump***, and ***uncertainty*** as dominant anomaly modes. This suggests that the downstream pipeline is not merely responding to a single heuristic trigger, but to several forms of anomalous local behavior. At the same time, all top-ranked candidate regions are labeled as ***severe*** at dominant severity level, which is consistent with the fact that they have already passed through a prioritization step designed to keep only the most relevant localized anomalies.

Overall, the quantitative anomaly results indicate that the developed pipeline produces a structured and interpretable hierarchy of suspicious evidence. The output is not a diffuse cloud of isolated anomalies, but a ranked set of localized candidate regions that can be examined in a technically meaningful way.

7.6 QUALITATIVE VALIDATION THROUGH OVERLAY FIGURES

The numerical ranking of anomaly regions becomes much more informative when it is examined together with the corresponding overlay figures. For this reason, the qualitative validation stage focuses on the twelve top-ranked candidate regions listed in Table 12 and Table 13**¡Error! No se encuentra el origen de la referencia.** and visualized through the full-context and zoomed panels generated by the pipeline. These figures make it possible to verify whether the anomaly evidence identified numerically is also spatially coherent when superimposed on the point-cloud geometry.

The decision to include the full set of twelve regions is justified because these cases are not arbitrary visual examples, but the top-ranked outputs of the region-prioritization stage. Together, they cover the three selected scenes and represent the main anomaly types detected by the system.

7.6.1 PLATEAU REGIONS

The two plateau cases, corresponding to ***Region 01*** and ***Region 02***, are extracted from the same tile, ***Tibet-12***, and form the most extreme part of the anomaly ranking. Their overlay figures show highly localized suspicious intervals, both spatially compact and strongly delimited in the zoomed panel. This is consistent with their very high ***score_max*** and regional priority values.

In ***Region 01***, the dominant anomaly type is ***height_jump***, and the figure shows a narrow interval in which the anomaly evidence is sharply concentrated rather than broadly diffused. As seen in Figure 26, the contrast between the full panel and the zoomed panel is particularly useful here: the full view shows that the candidate occupies only a very limited part of the overall tile, while the zoom view confirms that the anomaly signal is spatially focused. In methodological terms, this is an important result because it suggests that the ranking is not driven by widespread noise but by a compact localized pattern.

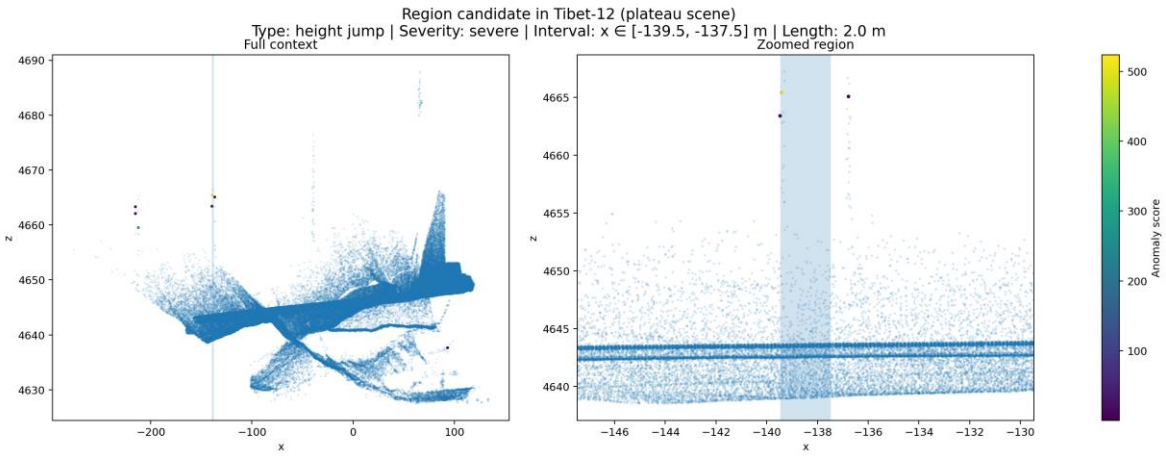

Figure 26: Top-ranked plateau anomaly region in Tibet-12

In ***Region 02***, the dominant type shifts to ***gap***, and the candidate extends over a slightly longer interval. The corresponding figure again supports the idea of a sharply bounded severe region. Taken together, the two plateau cases show that the anomaly pipeline can

identify more than one distinct high-priority segment within the same tile and separate different anomaly modes spatially. They also reveal that the most extreme anomaly evidence in the dataset is associated with short and highly concentrated plateau intervals rather than with diffuse scene-wide instability.

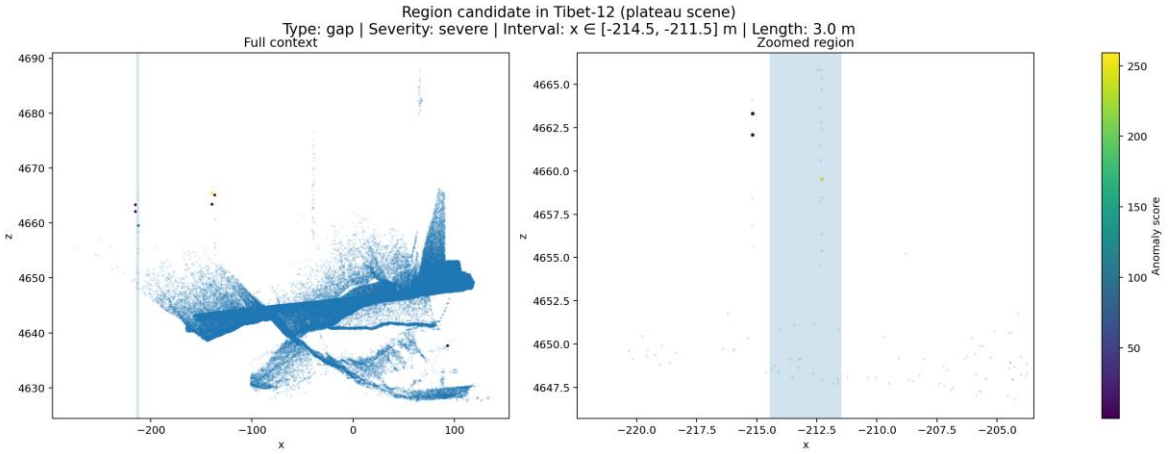

Figure 27: Second top-ranked plateau anomaly region in Tibet-12

The above Figure 27 confirms that the second plateau candidate is also spatially compact and clearly delimited, which supports the interpretation that ***Tibet-12*** contains more than one independently localized high-priority anomaly segment.

7.6.2 URBAN REGIONS

The urban group includes ***Region 03, Region 04, Region 10, and Region 11***, all of them extracted from the selected ***urban_splitA_seed43*** run. These figures are particularly important because the urban scene is structurally denser and visually more cluttered than the rural scene, which makes qualitative anomaly localization more demanding.

The first urban case, ***Region 03***, corresponds to a ***gap*** anomaly in ***L10-1-M01-002***. As seen in Figure 28, it shows a short severe interval of approximately 3 m, localized in a constrained portion of the scene rather than spread across the full point cloud. This supports the idea that

the pipeline can isolate a compact discontinuity-like event even in a visually complex urban environment.

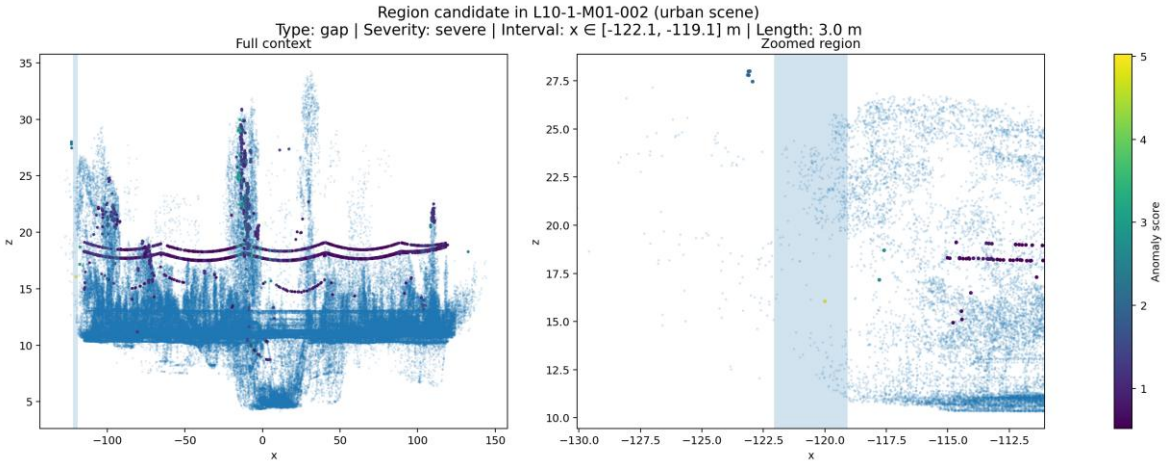

Figure 28: Urban gap-type anomaly in L10-1-M01-002

The second urban case, ***Region 04***, corresponds to a ***height_jump*** anomaly in ***L15-1-M01-002***. As seen in Figure 29, in contrast with the previous example, the suspicious interval is centered in a positive-coordinate portion of the profile, but it preserves the same localized structure: a short severe segment that stands out clearly in the zoomed panel.

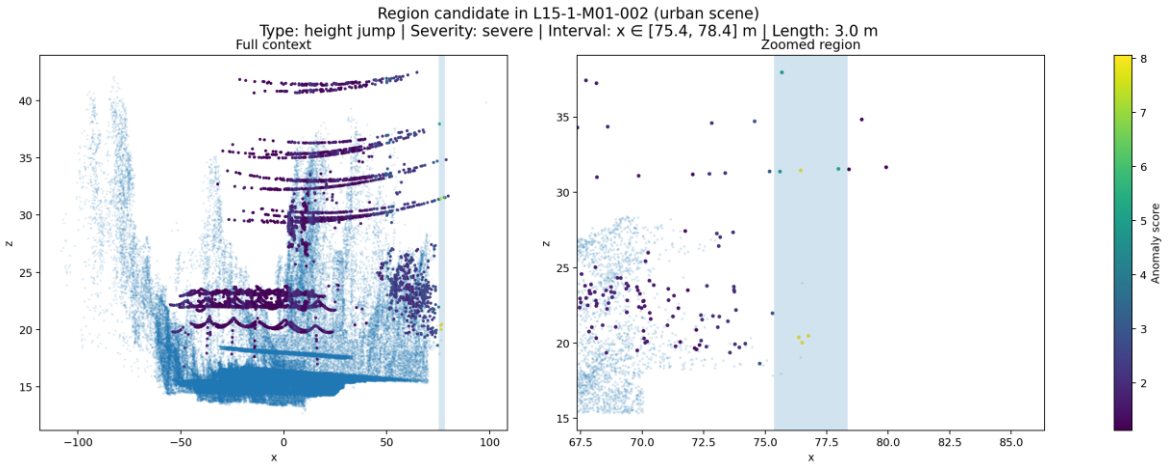

Figure 29: Urban height-jump anomaly in L15-1-M01-002

The remaining urban candidates, ***Region 10*** and ***Region 11***, are also dominated by ***height_jump***. Although they belong to different tiles, they share a similar qualitative pattern: the severe evidence remains concentrated in a short interval and aligns with a limited structural portion of the overhead geometry. The figures Figure 30 and Figure 31 and do not show a random dispersion of high anomaly scores, but rather localized segments in which the suspicious evidence clusters around a small region of the profile. This repeatability across several independent urban tiles strengthens the plausibility of the detected pattern.

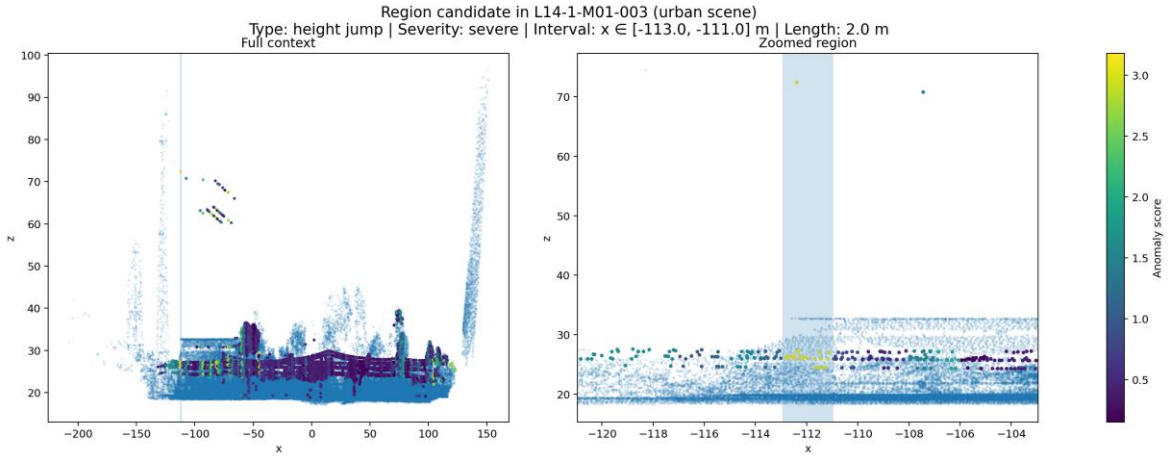

Figure 30: Urban height-jump anomaly in L14-1-M01-003

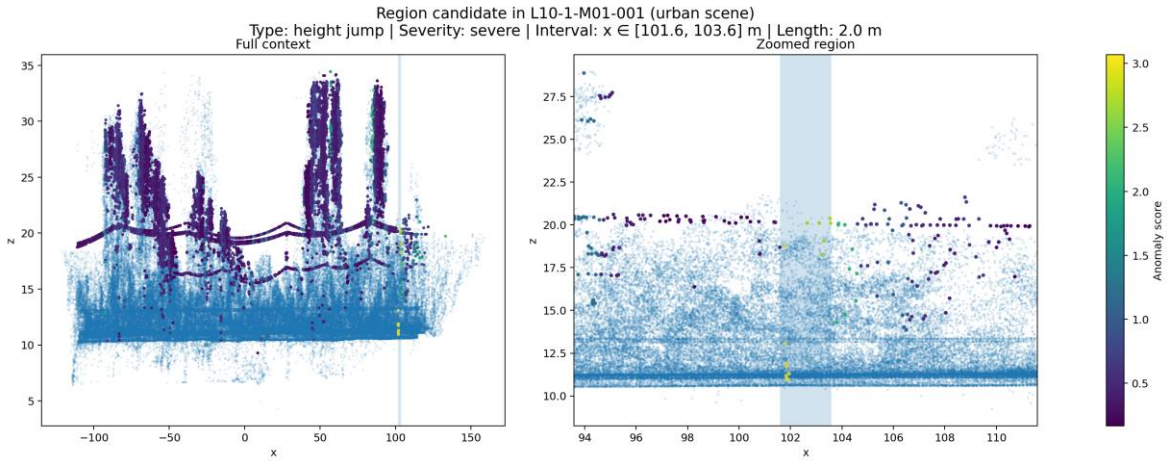

Figure 31: Urban height-jump anomaly in L10-1-M01-001

A particularly valuable aspect of the urban visual group is that it demonstrates that strong anomaly ranking is not reserved only for sparse or simple scenes. The method continues to produce compact severe candidates even under the denser structural context of urban railway environments.

7.6.3 RURAL REGIONS

The rural group comprises ***Region 05, Region 06, Region 07, Region 08, Region 09,*** and ***Region 12***, all of them extracted from ***rural_splitC_seed43***. Their regional priorities are lower than those of the strongest plateau and urban cases, but they form a particularly valuable cluster because they show repeated severe candidates across several different rural sections.

The rural figures reveal two recurring dominant anomaly modes. The first is ***height_jump***, represented by ***Region 06, Region 07,*** and ***Region 08***. These cases display short intervals in which the anomaly evidence concentrates around a compact part of the profile and does not spread arbitrarily through the whole tile. The second dominant mode is uncertainty, represented by ***Region 05, Region 09,*** and ***Region 12***. Although numerically weaker than the strongest plateau outliers, these regions remain severe and visually coherent in their localized zoomed views.

The first rural case, as seen in Figure 32, ***Region 05***, is extracted from ***section025*** and is dominated by ***uncertainty***. Its figure shows a compact interval that remains visually localized and does not fragment into multiple disjoint suspicious sections.

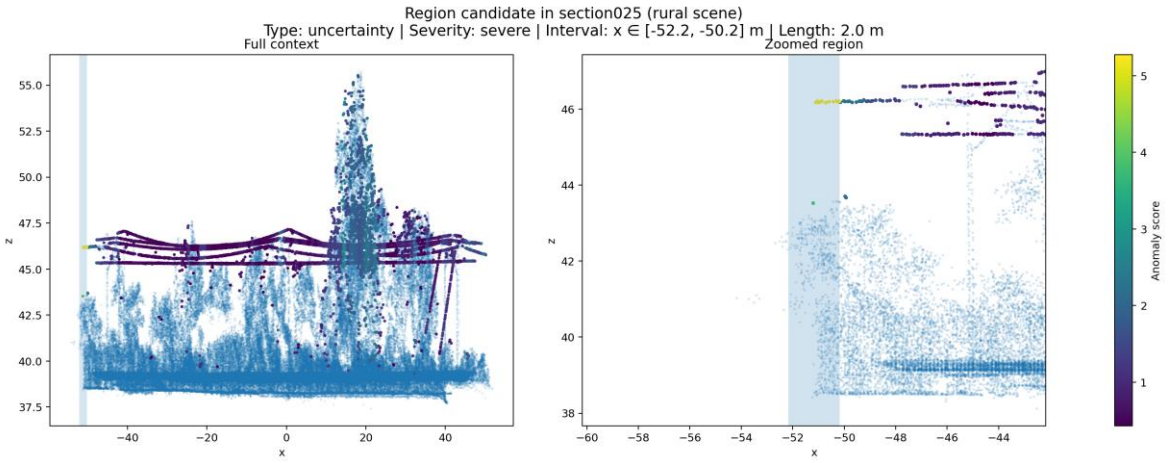

Figure 32: Rural uncertainty-type anomaly in section025

The next three rural cases correspond to ***height_jump*** anomalies in ***section061, section046,*** and ***section033***. Their visual similarity is useful because it reveals a repeated pattern rather than an isolated event. In all three cases, the highlighted interval is short, spatially coherent, and geometrically concentrated.

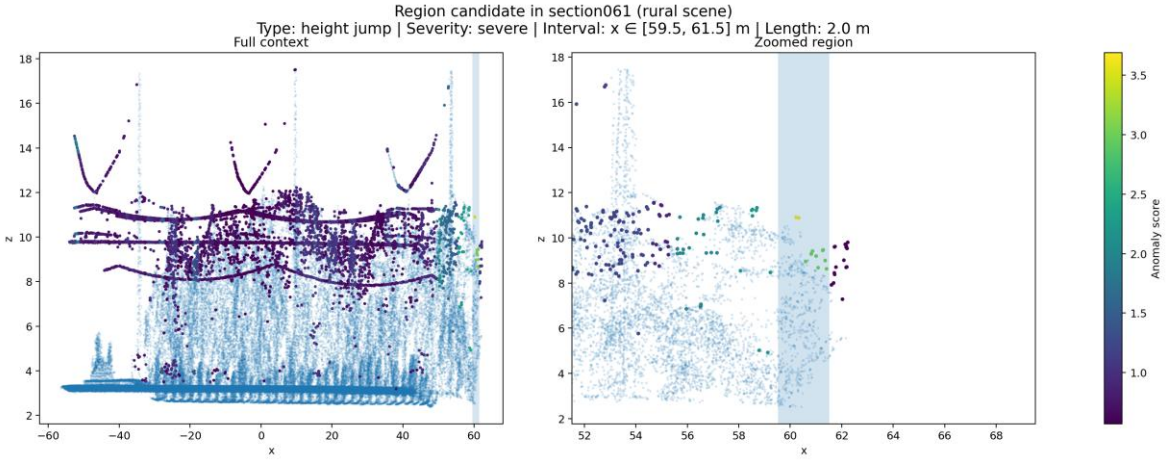

Figure 33: Rural height-jump anomaly in section061

The above Figure 33 is the visualization of ***Region 06*** in ***rural_splitC_seed43 / section061***, corresponding to the interval 59.5 m to 61.5 m. The region is classified as severe, and ***height_jump*** is the dominant anomaly type.

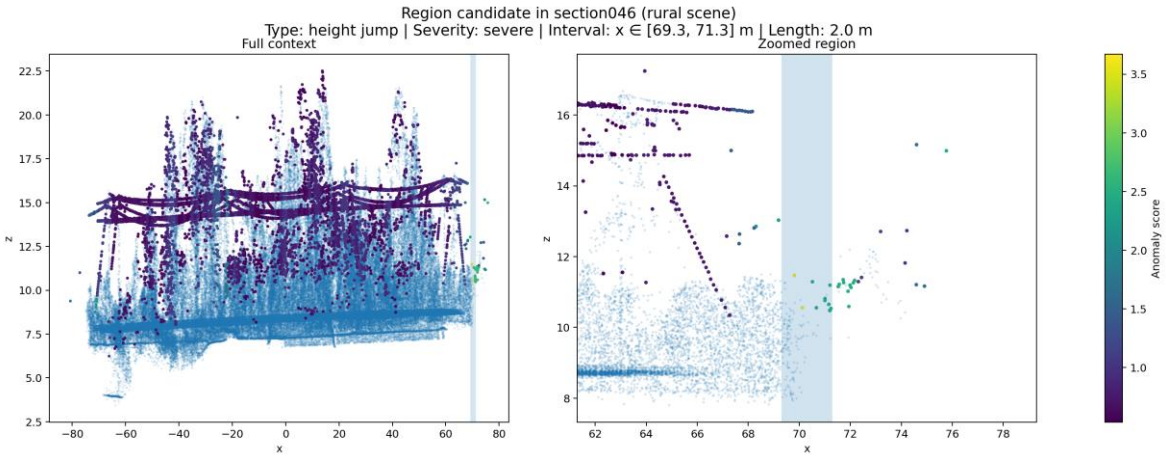

Figure 34: Rural height-jump anomaly in section046

Tha above Figure 34 is the visualization of ***Region 07*** in ***rural_splitC_seed43 / section046***, corresponding to the interval from 69.3 m to 71.3 m. The candidate is classified as severe, with ***height_jump*** as the dominant anomaly type.

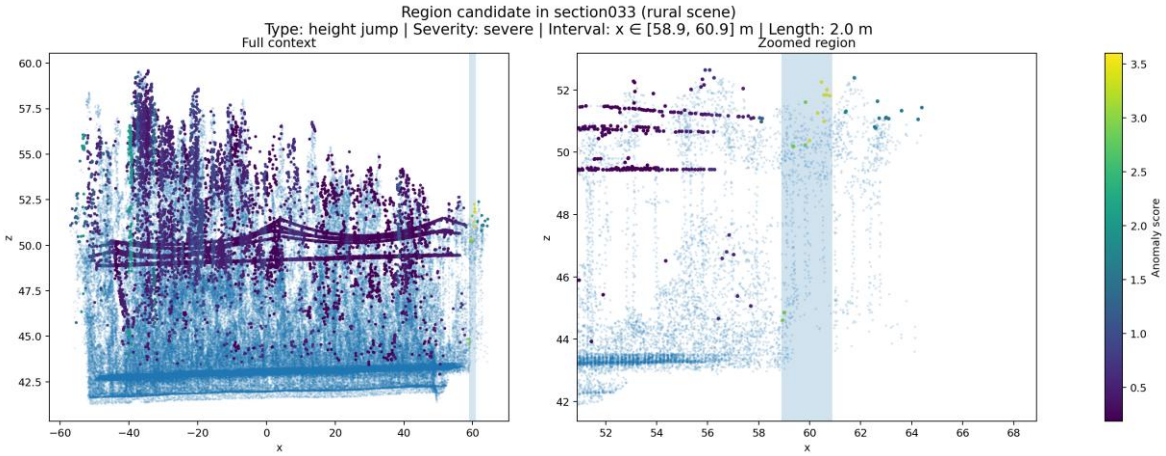

Figure 35: Rural height-jump anomaly in section033

The above Figure 35 is the visualization of ***Region 08*** in ***rural_splitC_seed43 / section033***, corresponding to the interval 58.9 m to 60.9 m. The region is again classified as severe and dominated by height_jump. Its compact structure reinforces the visual consistency of the rural height-jump group.

The remaining rural candidates correspond to uncertainty anomalies in ***section029*** and ***section015***. These figures Figure 36 and Figure 37 show that the rural anomaly set is not restricted to a single dominant mode and that the pipeline remains sensitive to different forms of suspicious local behavior.

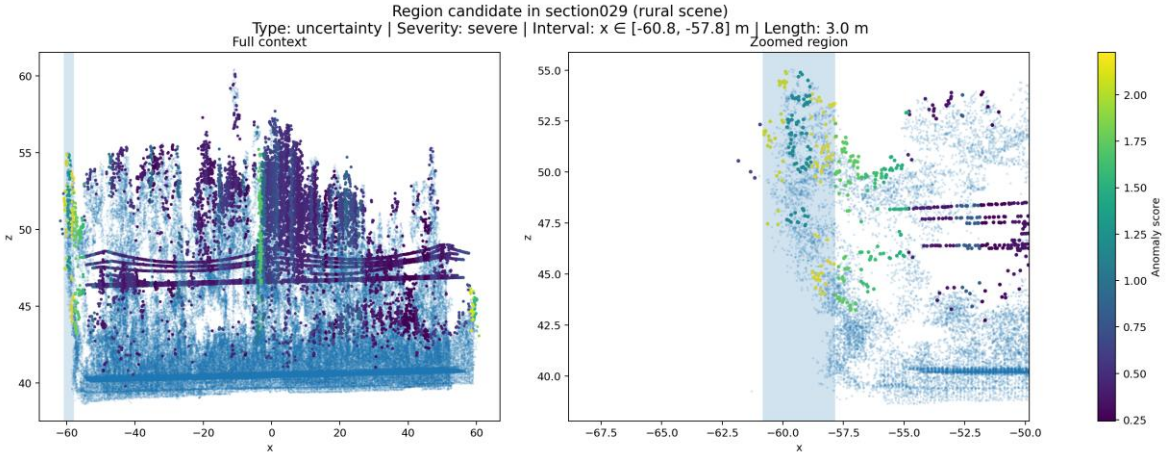

Figure 36: Rural uncertainty-type anomaly in section029

The above Figure 36 is the visualization of ***Region 09*** in ***rural_splitC_seed43 / section029***, corresponding to the interval −60.8 m to −57.8 m. The region is classified as severe, with uncertainty as the dominant anomaly type.

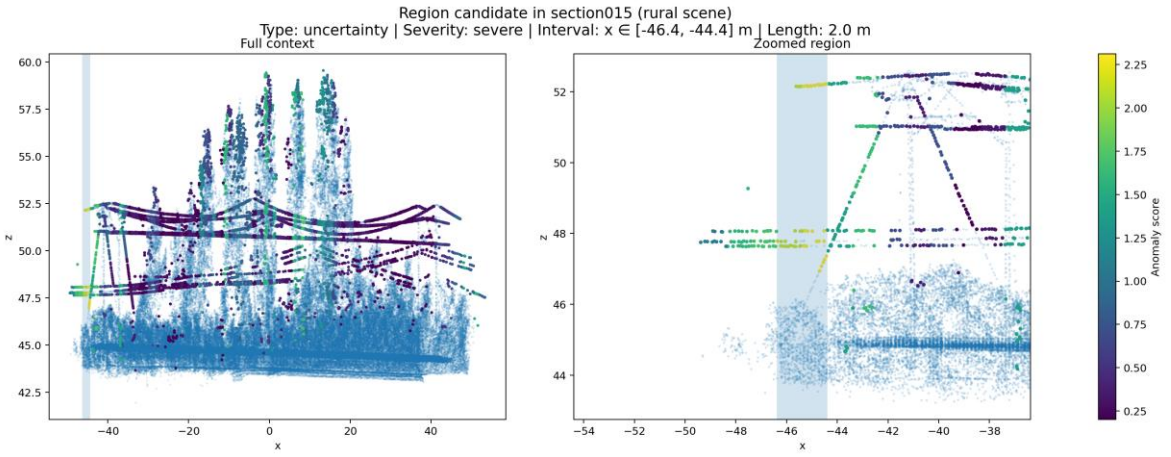

Figure 37: Rural uncertainty-type anomaly in section015

The above Figure 37 is the visualization of ***Region 12*** in ***rural_splitC_seed43 / section015***, corresponding to the interval −46.4 m to −44.4 m. The region is classified as severe, with uncertainty as the dominant anomaly type.

This repetition across several rural sections shows that the anomaly pipeline does not only produce a few isolated candidate regions. Instead, it identifies a family of similarly structured suspicious intervals across multiple independent tiles. From a validation perspective, this is valuable evidence of consistency. Even if the rural regions do not dominate the top of the ranking numerically, their recurrence suggests that the anomaly-oriented stage captures a stable type of local abnormal behavior rather than a single accidental artifact.

7.6.4 OVERALL QUALITATIVE INTERPRETATION

Across the twelve candidate regions, the qualitative validation shows a consistent pattern: the anomaly evidence tends to appear in short, localized intervals, typically between 2 and 3 meters, and the highlighted regions remain compact in the zoomed views. This means that the region extraction stage does not simply select broad noisy areas, but identifies bounded suspicious segments that can realistically be inspected by a human analyst.

A second important observation is that the figures support the presence of more than one anomaly mode in the data. The strongest candidates are not all the same type. Instead, the visual set includes ***severe gap, height_jump,*** and ***uncertainty*** cases. This increases the value of the anomaly-oriented extension because it shows that the downstream analysis is not reduced to a single failure pattern.

Qualitative interpretation also benefits from the scene diversity of the selected figures. Plateau contributes the most extreme outliers, urban contributes several strong compact candidates in cluttered environments, and rural contributes a broader cluster of repeated severe regions. This distribution makes visual validation more robust than a scene-specific analysis based on only one environment.

7.7 DISCUSSION OF RESULTS AND LIMITATIONS

The experimental results presented in this chapter support several conclusions regarding both the segmentation framework and the anomaly-oriented extension. First, the comparative benchmark shows clear progression across model families. The per-point MLP baseline establishes a very weak lower bound, the lightweight point-based models improve substantially over that reference, and RandLA-Net provides the best consolidated performance among the currently available point-based results. This pattern confirms that the segmentation of railway catenary-related point clouds depends strongly on the ability of the model to capture local geometric context rather than on isolated point-wise descriptors

alone. The Minkowski CNN reinforces this trend, although its scene-dependent performance remains less consistent than RandLA-Net.

A second important finding is that the superiority of RandLA-Net is not restricted to a single favorable run. Its advantage appears at scene level, split level, and in several seed-specific configurations.

The anomaly-oriented results also support the practical value of the downstream analysis. The pipeline does not merely highlight isolated anomalous points, but produces ranked suspicious tiles and, more importantly, compact region-level candidates characterized by interval, dominant anomaly type, severity, maximum score, and regional priority. This makes the outputs considerably more interpretable from an inspection perspective. Instead of returning diffuse evidence with unclear operational meaning, the system proposes localized segments that can be reviewed in a targeted way.

Qualitative validation through overlay figures reinforces this interpretation. Across plateau, urban, and rural scenes, the selected regions remain short, spatially concentrated, and visually coherent in the zoomed panels. This suggests that the ranked anomaly regions are not dominated by broad noisy patterns, but by localized structures that remain plausible as inspection candidates. The presence of several dominant anomaly modes, including ***gap,*** ***height_jump,*** and ***uncertainty***, also indicates that the downstream analysis is not restricted to a single type of suspicious behavior.

At the same time, the results must be interpreted with appropriate caution. The most important limitation is that the anomaly-oriented stage does not rely on exhaustive ground-truth labels of real defects. For this reason, the extracted regions are best understood as **prioritized suspicious candidates** rather than as definitively verified faults. Consequently, the anomaly module operates primarily as a ranking and inspection-support mechanism, not as a final fault-classification system.

A second limitation concerns the interpretation of very extreme outliers. The plateau tile ***Tibet-12***, for example, contains the two most highly ranked regions by a wide margin. These

cases are highly informative, but they also show that score-based indicators can become numerically amplified in short-support contexts. Such outliers remain valuable for prioritization, yet they require careful reading and should not be interpreted in isolation as evidence of scene-wide behavior.

A third limitation is that the anomaly-oriented stage currently depends on the selected RandLA-Net backbone rather than on a broader multi-backbone consensus. This choice is justified by the available validation evidence, but it also means that the anomaly ranking inherits part of the behavior of that specific segmentation family.

Overall, the results indicate that the developed framework succeeds in linking railway point-cloud semantic segmentation with anomaly-oriented prioritization in a technically coherent and visually interpretable way. Even with the limitations described above, the chapter provides strong evidence that the proposed workflow is suitable as a basis for data-driven railway catenary inspection support.

Chapter 8. CONCLUSIONS AND FUTURE WORK

This chapter closes the thesis by summarizing the main conclusions derived from the developed framework and the experimental results. It also outlines the most relevant future research directions that can extend the present work toward more robust, scalable, and operationally meaningful railway catenary inspection systems.

8.1 CONCLUSIONS

This thesis develops a reproducible framework for semantic segmentation and anomaly-oriented analysis of railway catenary point clouds. The work starts from the premise that railway catenary systems are critical assets whose inspection remains technically demanding due to the geometric thinness of overhead structures, the large scale of railway corridors, and the difficulty of identifying suspicious patterns in heterogeneous outdoor scenes. In response to this problem, the thesis proposes a workflow that connects data preparation, model benchmarking, evaluation, and downstream anomaly-oriented interpretation within a single experimental structure.

From the point of view of the thesis objectives, the work achieves its main purpose. A complete processing pipeline is defined and implemented, beginning with the preparation of the WHU-Railway3D-derived splits and continuing through the training and evaluation of several segmentation model families. This pipeline remains organized around reproducibility, output traceability, and comparability across scenes, splits, and seeds.

The comparative benchmark also provides a clear technical conclusion. The per-point MLP baseline confirms that point-wise feature learning alone is insufficient for the target task. The lightweight point-based models improve substantially over that lower bound, which shows the value of richer local abstraction. The Minkowski CNN further extends the comparison by introducing a sparse-convolutional 3D representation, achieving clear improvements over the MLP baseline and competitive results in plateau and especially urban

scenes. However, its performance remains more heterogeneous across environments and does not surpass RandLA-Net. Overall, the strongest consolidated results correspond to RandLA-Net, which offers the best balance between segmentation quality, repeatability, and operational feasibility among the currently available results.

A second major contribution lies in the anomaly-oriented stage itself. Rather than stopping at semantic segmentation scores, the framework extends the selected backbone toward the detection and ranking of suspicious regions. This extension transforms local anomaly evidence into region-level outputs described through interpretable descriptors such as interval, severity, dominant anomaly type, and priority. The resulting ranked regions and overlay figures provide a much more inspection-oriented representation than semantic segmentation alone.

The visual analysis further supports the usefulness of this extension. The selected candidate regions remain spatially compact and coherent across the three considered scenes, which suggests that the pipeline can identify localized suspicious segments rather than only diffuse numerical irregularities.

Another relevant conclusion concerns the feasibility of the work under constrained computational conditions. The thesis is developed in a CPU-only environment, which influences preprocessing choices, experiment management, and model execution strategy. Instead of treating this as a purely negative limitation, the work incorporates it into the design of the framework. This is particularly relevant for the Minkowski benchmark, whose sparse-convolutional evaluation required specific adaptations, including a sliding-window strategy to ensure full validation coverage and avoid low-coverage artefacts. The resulting pipeline therefore shows that meaningful experimentation in railway point-cloud analysis remains possible even without dedicated GPU resources, provided that the workflow is carefully structured and computational decisions remain disciplined.

Taken together, the thesis contributes to the field in three main ways. First, it establishes a comparative benchmark for railway catenary point-cloud segmentation under multiple scenes and configurations, including both point-based and sparse-convolutional 3D model

families. Second, it identifies RandLA-Net as the most suitable currently consolidated backbone for the developed workflow, while positioning Minkowski CNN as a valuable but less consistent benchmark under the current experimental conditions. Third, it demonstrates that segmentation outputs can be extended toward anomaly-oriented prioritization through a reproducible and visually interpretable downstream pipeline. These contributions make the work technically relevant and provide a solid basis for future extensions in intelligent railway infrastructure monitoring.

8.2 FUTURE WORK

Although the thesis achieves its main objectives, several directions remain open for future development. One relevant line of future work is the further optimization of the sparse-convolution benchmark based on the Minkowski framework. In this thesis, Minkowski CNN has already been integrated into the comparative analysis and provides a valuable reference for assessing sparse voxelized processing against point-based architectures. However, future work can explore a broader range of sparse-convolution configurations, including alternative voxel sizes, deeper architectures, extended hyperparameter tuning, and GPU-based training. This would help determine whether the heterogeneous behavior observed across scenes is mainly due to architectural limitations, voxelization choices, data characteristics, or computational constraints.

A second natural line of future work is the expansion of the anomaly-oriented stage toward broader robustness analysis. At present, the anomaly workflow is built on the strongest RandLA-Net configurations, which is methodologically justified by the available results. However, a future version of the work can evaluate whether the region ranking remains stable across several strong backbones or across a larger number of selected checkpoints. Such an extension would strengthen the reliability of the anomaly prioritization stage.

A third important direction is the incorporation of more explicit ground truth for anomaly validation. The present thesis treats anomaly outputs as prioritized suspicious candidates, which is an appropriate interpretation given the available data. Future work can improve this by introducing manually validated anomaly labels, expert inspection reports, or known defect references. This would make it possible to move from plausibility-based validation toward a more formal evaluation of anomaly-detection precision and recall.

Another valuable extension concerns multimodal integration. Railway monitoring increasingly combines point clouds with RGB imagery, thermal sensing, radar, and localization data. A future system can therefore enrich the present framework by combining 3D semantic understanding with additional sensor modalities, which may improve both segmentation robustness and anomaly interpretability in complex scenes.

From a methodological perspective, future work can also explore more advanced anomaly strategies. These may include reconstruction-based normality modeling, uncertainty-aware ranking, temporal consistency across repeated inspections, or hybrid geometric rules that explicitly exploit engineering knowledge about catenary structure. Such developments would move the pipeline closer to a fully inspection-aware intelligent support system.

Finally, the deployment perspective remains a significant avenue for future work. The present thesis develops a research-oriented framework, but its outputs are already aligned with the logic of maintenance support. A future version can therefore focus on integration into operational workflows, including semi-automatic review tools, technician-facing visualization interfaces, and scalable offline analysis pipelines for large railway networks.

In summary, the thesis establishes a solid technical foundation. Its main value lies in demonstrating that railway catenary point-cloud segmentation and anomaly-oriented prioritization can be connected within a coherent workflow. Future work can now build upon that basis by strengthening the benchmarking layer, validating anomaly outputs more formally, expanding the sensing modalities, and bringing the framework closer to real inspection practice.

Chapter 9. BIBLIOGRAPHY

[1] Bris-Peñalver, F. J., Verdecia-Peña, R., & Alonso, J. I. (2026). *A survey of AI-enabled predictive maintenance for railway infrastructure: Models, data sources, and research challenges*. *Sensors*, *26*(3), 906. https://doi.org/10.3390/s26030906

[2] Chen, L., Xu, C., Lin, S., Li, S., & Tu, X. (2020). *A deep learning-based method for overhead contact system component recognition using mobile 2D LiDAR*. *Sensors*, *20*(8), 2224. https://doi.org/10.3390/s20082224

[3] Ghasemlou, A., Soilán, M., & Riveiro, B. (2026). *SemanticRail3D: A mobile LiDAR benchmark for semantic and instance segmentation of railway corridors*. *Scientific Data*, *13*, 82. https://doi.org/10.1038/s41597-025-06392-9

[4] Grandio, J., Riveiro, B., Soilán, M., & Arias, P. (2022). *Point cloud semantic segmentation of complex railway environments using deep learning*. *Automation in Construction*, *141*, 104425. https://doi.org/10.1016/j.autcon.2022.104425

[5] Lamas, D., Soilán, M., Grandío, J., & Riveiro, B. (2021). *Automatic point cloud semantic segmentation of complex railway environments*. *Remote Sensing*, *13*(12), 2332. https://doi.org/10.3390/rs13122332

[6] Network Rail. (n.d.). *New Measurement Train (NMT)*. https://www.networkrail.co.uk/our-work/looking-after-the-railway/our-fleet-machines-and-vehicles/new-measurement-train-nmt/

[7] Qiu, B., Zhou, Y., Dai, L., Wang, B., Li, J., Dong, Z., Wen, C., Ma, Z., & Yang, B. (2024). *WHU-Railway3D: A diverse dataset and benchmark for railway point cloud semantic segmentation*. *IEEE Transactions on Intelligent Transportation Systems*, *25*(12), 20900–20916. https://doi.org/10.1109/TITS.2024.3469546

[8] Rodríguez-Hernández, M., Crespo-Márquez, A., Sánchez-Herguedas, A., & González-Prida, V. (2025). *Digitalization as an enabler in railway maintenance: A review from "The International Union of Railways Asset Management Framework" perspective*. *Infrastructures*, *10*(4), 96. https://doi.org/10.3390/infrastructures10040096

[9] Sainz-Aja, J. A., Pombo, J., Brant, J., Antunes, P., Rebelo, J. M., Santos, J., & Ferreño, D. (2025). *Real-time rail electrification systems monitoring: A review of technologies*. *Sensors*, *25*(21), 6625. https://doi.org/10.3390/s25216625

[10] Scordamaglia, D. (2019, February). *Digitalisation in railway transport: A lever to improve rail competitiveness*. European Parliamentary Research Service. https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI%282019%29635528

[11] Tu, X., Zhang, C., Liu, S., Xu, C., & Li, R. (2023). *Point cloud segmentation of overhead contact systems with deep learning in high-speed rails*. *Journal of Network and Computer Applications*, *216*, 103671. https://doi.org/10.1016/j.jnca.2023.103671

[12] Lupi, C., Felli, F., Ciro, E., Paris, C., & Vendittozzi, C. (2021). Railway overhead contact wire monitoring system by means of FBG sensors. Fracture and Structural Integrity, 15(57), 246–258. https://doi.org/10.3221/IGF-ESIS.57.18

[13] Chen, J., Liu, Z., Wang, H., Núñez, A., & Han, Z. (2018). *Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network*. *IEEE Transactions on Instrumentation and Measurement*, *67*(2), 257–269. https://doi.org/10.1109/TIM.2017.2775345

[14] Chen, X., Chen, Z., Liu, G., Chen, K., Wang, L., Xiang, W., & Zhang, R. (2021). *Railway overhead contact system point cloud classification*. *Sensors*, *21*(15), 4961. https://doi.org/10.3390/s21154961

[15] Choy, C., Gwak, J., & Savarese, S. (2019). *4D spatio-temporal ConvNets: Minkowski convolutional neural networks*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3070–3079). https://doi.org/10.1109/CVPR.2019.00319

[16] Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2020). *RandLA-Net: Efficient semantic segmentation of large-scale point clouds*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11108–11117).

[17] Kang, G., Gao, S., Yu, L., Zhang, D., Wei, X., & Zhan, D. (2019). *Contact wire support defect detection using deep Bayesian segmentation neural networks and prior geometric knowledge*. *IEEE Access*, *7*, 173366–173376. https://doi.org/10.1109/ACCESS.2019.2955753

[18] Kharroubi, A., Ballouch, Z., Hajji, R., Yarroudh, A., & Billen, R. (2024). *Multi-context point cloud dataset and machine learning for railway semantic segmentation*. *Infrastructures*, *9*(4), 71. https://doi.org/10.3390/infrastructures9040071

[19] Liu, S., Yu, L., & Zhang, D. (2019). *An efficient method for high-speed railway dropper fault detection based on depthwise separable convolution*. *IEEE Access, 7*, 135678–135688. https://doi.org/10.1109/ACCESS.2019.2942079

[20] Liu, W., Liu, Z., Núñez, A., & Han, Z. (2020). *Unified deep learning architecture for the detection of all catenary support components*. *IEEE Access, 8*, 17049–17059. https://doi.org/10.1109/ACCESS.2020.2967831

[21] Lyu, Y., Han, Z., Zhong, J., Li, C., & Liu, Z. (2020). *A generic anomaly detection of catenary support components based on generative adversarial networks*. *IEEE Transactions on Instrumentation and Measurement, 69*(5), 2439–2448. https://doi.org/10.1109/TIM.2019.2954757

[22] Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). *PointNet++: Deep hierarchical feature learning on point sets in a metric space*. In *Advances in Neural Information Processing Systems* (Vol. 30).

[23] Tagiew, R., Köppel, M., Schwalbe, K., Denzler, P., Neumaier, P., Klockau, T., Boekhoff, M., Klasek, P., & Tilly, R. (2023). *OSDaR23: Open sensor data for rail 2023*. In *2023 8th International Conference on Robotics and Automation Engineering (ICRAE)* (pp. 270–276). https://doi.org/10.1109/ICRAE59816.2023.10458449

[24] Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). *KPConv: Flexible and deformable convolution for point clouds*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6411–6420).

[25] Zendel, O., Murschitz, M., Zeilinger, M., Steininger, D., Abbasi, S., & Beleznai, C. (2019). *RailSem19: A dataset for semantic rail scene understanding*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 32–40). https://doi.org/10.1109/CVPRW.2019.00161

[26] Zhong, J., Liu, Z., Yang, C., Wang, H., Gao, S., & Núñez, A. (2022). *Adversarial reconstruction based on tighter oriented localization for catenary insulator defect detection in high-speed railways*. *IEEE Transactions on Intelligent Transportation Systems, 23*(2), 1109–1120. https://doi.org/10.1109/TITS.2020.3020287

[27] Zhong, J., Liu, Z., Wang, H., Liu, W., Yang, C., Han, Z., & Núñez, A. (2021). *A looseness detection method for railway catenary fasteners based on reinforcement learning refined localization*. *IEEE Transactions on Instrumentation and Measurement, 70*, 3518913. https://doi.org/10.1109/TIM.2021.3086913

[28] Chen, Z., Yang, J., Feng, Z., & Zhu, H. (2024). *RailFOD23: A dataset for foreign object detection on railroad transmission lines*. *Scientific Data, 11*, 72. https://doi.org/10.1038/s41597-024-02918-9

[29] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). *PointNet: Deep learning on point sets for 3D classification and segmentation*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 652–660). https://doi.org/10.1109/CVPR.2017.16

[30] United Nations. (2015). Transforming our world: The 2030 Agenda for Sustainable Development (A/RES/70/1). https://sdgs.un.org/2030agenda

APPENDIX A. CONTRIBUTION OF THE THESIS TO THE SUSTAINABLE DEVELOPMENT GOALS (SDGS)

This thesis contributes to the Sustainable Development Goals (SDGs) [30] by addressing the digitalization and intelligent monitoring of railway infrastructure through point-cloud semantic segmentation and anomaly-oriented analysis. Although the work is primarily technical, its results are aligned with broader sustainability objectives related to infrastructure reliability, transport efficiency, safer maintenance practices, and more resource-efficient asset management.

The thesis focuses on railway catenary systems, which are essential components of electrified rail transport. From this perspective, the proposed framework contributes indirectly to more sustainable mobility by supporting the inspection and maintenance of critical infrastructure through data-driven methods. The use of semantic segmentation and anomaly-oriented analysis has the potential to improve early detection, inspection prioritization, and operational understanding of railway assets, which are all relevant for more resilient and efficient transport systems.

SDG 9. Industry, Innovation and Infrastructure

This is the SDG most directly connected with the thesis. The work develops a technical framework for the intelligent monitoring of railway infrastructure using advanced methods in 3D point-cloud processing, deep learning, and anomaly-oriented analysis. The proposed system contributes to innovation in infrastructure management by exploring how modern AI-based methods can support the inspection of overhead railway systems in a scalable and reproducible way.

The thesis also contributes to the strengthening of infrastructure-related digital capabilities. It does not only evaluate isolated models, but designs a complete workflow including data preparation, controlled experimentation, evaluation, and anomaly-oriented interpretation. In this sense, the contribution is not limited to algorithmic accuracy, but also includes the development of a structured technological pipeline that supports more advanced infrastructure monitoring practices.

SDG 11. Sustainable Cities and Communities

Rail transport plays a central role in sustainable mobility systems, especially in urban and metropolitan environments. By improving the capacity to monitor and interpret railway catenary infrastructure, this thesis contributes indirectly to more reliable railway operation and, therefore, to more robust public and interurban transport systems.

Although the thesis does not address transport policy directly, it supports one of its technological foundations: infrastructure reliability. Better inspection and maintenance support can reduce service disruptions, improve operational continuity, and strengthen confidence in electrified rail transport as a sustainable mobility solution. For this reason, the thesis is aligned with the broader objectives of sustainable and resilient transport systems within cities and regions.

SDG 12. Responsible Consumption and Production

The proposed framework also relates to more responsible asset management. Infrastructure maintenance becomes more efficient when inspection efforts can be prioritized according to data-driven evidence instead of relying only on rigid periodic routines or manual review. In this sense, the anomaly-oriented extension developed in the thesis supports a more focused

use of maintenance resources by helping identify suspicious regions that deserve attention first.

This approach may contribute to reducing unnecessary interventions, improving inspection efficiency, and extending the useful life of infrastructure components through earlier and more informed maintenance decisions. Therefore, even though the thesis is not centered on industrial production processes, it supports more responsible use of technical and operational resources in infrastructure management.

SDG 13. Climate Action

The contribution to climate action is indirect but relevant. Electrified railway systems are a key part of lower-emission transport strategies, and their reliability depends on the proper operation of catenary infrastructure. By supporting more intelligent inspection and maintenance of these systems, the thesis contributes to the technological ecosystem required for sustainable rail transport.

In this context, the work does not reduce emissions directly, but it strengthens the digital tools that can help maintain efficient and reliable electrified transport networks. From a long-term perspective, this type of technological support is compatible with broader climate goals, since resilient rail infrastructure is an important component of sustainable mobility transitions.

Secondary contribution: SDG 8. Decent Work and Economic Growth

A secondary connection can also be established with SDG 8, particularly in relation to safer and more efficient maintenance practices. Railway catenary inspection often involves technically demanding environments and highly specialized manual review. Digital support

tools such as the one developed in this thesis can help reduce the burden of purely manual inspection and improve the quality of technical decision-making.

In addition, infrastructure reliability has an economic dimension. Fewer disruptions, better maintenance prioritization, and more efficient inspection workflows support more stable railway operation, which is beneficial for productivity and for the economic value of transport systems.

Among all SDGs, the clearest contribution of this thesis is to SDG 9 (Industry, Innovation and Infrastructure), followed by indirect but meaningful contributions to SDG 11 (Sustainable Cities and Communities), SDG 12 (Responsible Consumption and Production), and SDG 13 (Climate Action). The thesis does not claim direct social or environmental intervention by itself, but it does provide a technological contribution that is aligned with more resilient, efficient, and sustainable railway infrastructure management.

In the following Table 14 the SDG contributions are summarized.

SDG	Title	Relation to the thesis	Type of contribution
SDG 9	Industry, Innovation and Infrastructure	Development of an AI-based framework for railway catenary monitoring, semantic segmentation, and anomaly-oriented analysis	Direct
SDG 11	Sustainable Cities and Communities	Support for more reliable rail transport infrastructure and more resilient mobility systems	Indirect
SDG 12	Responsible Consumption and Production	More efficient and prioritized infrastructure inspection and maintenance workflows	Indirect
SDG 13	Climate Action	Technological support for the reliability of electrified rail transport systems	Indirect
SDG 8	Decent Work and Economic Growth	Support for safer, more efficient, and more informed maintenance practices	Secondary / indirect

Table 14: Contribution to the Sustainable Development Goals