ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in ARTIFICIAL INTELLIGENCE

Assessing the Robustness of Optimization-Based Decision Support Systems for Sustainable Urban Planning: The Bologna Green Cells Case Study

CANDIDATE
Armando Renzullo

SUPERVISOR
Prof. Andrea Borghesi

Academic Year 2025/26

Session 1st

Abstract

Optimization-based decision support systems can support sustainable urban planning by making spatial trade-offs explicit, but their recommendations may depend strongly on uncertain input data. This thesis studies that issue in the TALEA Bologna Green Cells case study, where optimization models select candidate locations for modular green interventions over a city grid. The work first reconstructs the baseline data pipeline, preprocessing stages, and implemented MiniZinc models, then evaluates robustness under controlled synthetic perturbation scenarios applied to environmental, demographic, land-use, geometric, and allocation-related inputs. Robustness is assessed by comparing each perturbed solution with the corresponding no-noise baseline. Spatial robustness is measured through Jaccard similarity and centroid-buffer IoU, while quantitative robustness is evaluated through output-ratio metrics for utility, yard allocation, and street allocation. A cross-model comparison further normalizes these metrics to summarize average stability, lower-tail fragility, and degradation trends across increasing perturbation levels. Within the tested perturbation scenarios, many outputs show good average stability, but the results also reveal scenario-specific fragility: some model-source-perturbation combinations produce substantial spatial shifts or unstable quantitative ratios. The analysis is scenario-based and reproducible, but it is not a full Monte Carlo uncertainty quantification. Overall, the thesis shows how robustness evaluation can support more transparent and uncertainty-aware use of optimization-based decision support systems in urban planning.

Contents

Abstract 1

1 Introduction 1

1.1 Motivation 1
1.2 Optimization-Based Decision Support 2
1.3 Case Study and Thesis Scope 2
1.4 Research Objective 3
1.5 Research Questions 3
1.6 Thesis Organization 4

2 Background and Related Work 5

2.1 Sustainable Urban Planning and Climate Adaptation 5
2.2 Urban Green Infrastructure 6
2.3 Green Accessibility and Green Justice 6
2.4 Urban Heat and Environmental Indicators 7
2.5 UHEI, NDVI, and Fractional Vegetation Cover 7
2.6 Spatial Data and Indicators for Urban Planning 8
2.7 Decision Support Systems in Urban Planning 9
2.8 Optimization-Based and Interpretable Decision Support 9
2.9 Spatial Optimization and Location-Allocation Concepts 10

3 The TALEA Project and the Bologna Green Cells Case Study 11

3.1 Project Context 11

3.2 Bologna Green Cells Case Study 12
3.3 Planning Problem Addressed by the Baseline DSS 12
3.4 Intended Outputs 13
3.5 Baseline Workflow 13
3.6 Assumptions and Limitations of the Baseline Project 14

4 Data and Preprocessing 15
4.1 Overview 15
4.2 Spatial Unit of Analysis 19
4.3 Preprocessing Operations 19
4.4 Macro-Scale and Micro-Scale Indicators 20
4.5 Derived Variables 21
4.6 Dataset, Code, and Parameter Mapping 23
4.7 Missing Metadata and Data Limitations 25

5 Optimization Models 26
5.1 Implementation Sources and Modelling Scope 26
5.2 Sets, Parameters, and Decision Variables 26
5.3 Common Objective and Constraints 28
5.4 Standard Model 29
5.5 Difference Model 29
5.6 Inverse UHEI Model 30
5.7 NDVI Model 31
5.8 Model Family Comparison 31
5.9 Differences Between the Reported and Implemented Formulations 32

6 Robustness Evaluation Methodology 35
6.1 Evaluation Objective 35
6.2 Perturbation Design 36
6.3 Pipeline Logic 37

6.4 Comparison Design 38
6.5 Spatial Robustness Metrics 40
6.5.1 Jaccard Similarity 40
6.5.2 Centroid-Buffer IoU 41
6.6 Ratio-Based Robustness Metrics 42
6.7 Normalized Cross-Model Robustness Index 43

7 Implementation **45**
7.1 Scope of the Implementation Analysis 45
7.2 Raw-Data Perturbations 46
7.3 Grid-Level Perturbations 49
7.4 Instance Generation and Full-Space Perturbation 52
7.5 Perturbation Formulas and Code Snippets 52
7.5.1 Additive and Multiplicative Numerical Noise 52
7.5.2 Missingness and Median Imputation 53
7.5.3 Land-Use Mislabelling 54
7.5.4 Geometric Polygon Perturbation 55
7.5.5 Full-Space Perturbation 55
7.6 Evaluation Script 56
7.7 Cross-Model Comparison Outputs 58
7.8 Handling of Missing or Inconsistent Cases 59

8 Results **60**
8.1 Evaluated Result Set 60
8.2 Spatial Similarity by Model 61
8.3 Aggregation Scenarios 65
8.4 Ratio-Based Metrics 66
8.5 Cross-Model Quantitative Comparison 68
8.6 Summary of Main Observations 74

9 Discussion **75**

9.1 Interpreting Robustness in This Case Study 75

9.2 Meaning of the Cross-Model Comparison 76

9.3 Spatial Stability of Selected Cells 77

9.4 Quantitative Stability of Output Values 77

9.5 Model-Source-Perturbation Sensitivity 78

9.5.1 Model-Level Patterns 79

9.6 Relationship Between Spatial and Quantitative Metrics 80

9.7 Practical Implications for Decision Support 80

9.8 Methodological Limitations 81

10 Conclusion **83**

Bibliography **85**

List of Figures

3.1 Schema of the baseline TALEA decision-support workflow reconstructed from the project report and the implementation files. 13

4.1 Schema of the data-to-model pipeline for the baseline TALEA DSS, reconstructed from the project report and the implementation. 16

4.2 Schema of the macro-scale and micro-scale indicator structure of the baseline DSS. 20

5.1 Comparison of the implemented model families for the baseline DSS. 32

6.1 Schema of the robustness comparison design. The same model is compared against its own no-noise baseline under each recognized perturbation scenario. 38

8.1 Jaccard similarity for the `std` model. The plot shows high similarity for many scenarios but visible drops for land-use mislabelling, multiplicative external-space perturbation, and `num_areas` perturbation. 62

8.2 Jaccard similarity for the `diff` model. Aggregation scenarios for UHEI and NDVI remain at 1.0 in this metric, while several perturbations with ordered noise levels reduce spatial overlap. 62

8.3 Source-specific Jaccard plot for ext_space perturbations in the diff model. Multiplicative perturbation produces the strongest decrease among the tested ext_space scenarios. 63
8.4 Jaccard similarity for the inverse_uhei model. The largest spatial changes appear for land-use mislabelling and full-space perturbations. 64
8.5 Jaccard similarity for the ndvi model. The strongest drop is associated with Gaussian perturbation of the NDVI source. . . 64
8.6 Source-specific Jaccard plot for NDVI perturbations in the ndvi model. Gaussian NDVI perturbation has a much stronger effect than multiplicative NDVI perturbation within the tested scenarios. 65
8.7 Utility-ratio plot for the ndvi model. The extreme Gaussian NDVI values show a limitation of ratio-based utility comparison when the perturbed mean utility changes sign or becomes very large relative to the baseline. 67
8.8 Mean normalized robustness by model. The plot summarizes all valid level-based metric rows and should be read as an aggregate comparison, not as a replacement for metric-specific evidence. 69
8.9 Mean normalized robustness by model and metric. Darker or lower cells identify metric-specific sensitivity that can be hidden in the overall model average. 70
8.10 Mean normalized robustness by model and noise type. The plot highlights broad method-level patterns but does not show whether a low value is caused by one severe source or several moderate ones. 70
8.11 Mean normalized robustness by model and noise source. Source-level summaries help distinguish broad model sensitivity from sensitivity to specific input components. 71

8.12 Worst 20 normalized robustness series, ordered by the worst value reached across available noise levels. This plot emphasizes lower-tail fragility rather than average model behaviour. . 72
8.13 Series with the strongest negative degradation slopes. Slopes are fitted only for level-based perturbations with at least two valid levels and are reported in robustness percentage points per noise level. 73

List of Tables

4.1 Raw spatial and tabular datasets used by the baseline TALEA system: file, format, and CRS metadata. 16

4.2 Raw dataset source status and baseline role in the TALEA decision-support pipeline. 17

4.3 Processed spatial layers used before MiniZinc instance generation. 21

4.4 Derived-variable inventory for the baseline model inputs and outputs. 22

4.5 Mapping between datasets, preprocessing functions, and model parameters. 24

4.6 Missing metadata and baseline data limitations. 25

5.1 Notation for the implemented optimization models. 26

5.2 Comparison of the four implemented optimization models. . . 32

5.3 Differences between the reported and implemented formulations. 33

6.1 Interpretation of the robustness metrics used in the baseline-versus-perturbed comparison. 39

7.1 Raw-data perturbation functions used to generate robustness scenarios. 46

7.2 Grid-level perturbation functions used to generate robustness scenarios. 49

8.1 Summary of spatial robustness metrics in the experimental result set. IoU buffer refers to the centroid-matching metric implemented with a 300 m default tolerance. 61
8.2 Range of the `mean_noise_over_baseline` ratio in the three ratio-based metric files for each model. 66
8.3 Normalized robustness summary by model for level-based perturbation scenarios. The table combines spatial scores and ratio metrics, so it should be interpreted together with the metric-specific results. 68

Chapter 1

Introduction

1.1 Motivation

European cities are increasingly required to plan urban transformations that respond to climate risks and adaptation needs, while recognizing that climate impacts and adaptive capacity are unevenly distributed across urban communities [7, 8]. The original TALEA report by Di Buò and Reale [6] frames Bologna within this context, emphasizing Urban Heat Island (UHI) and Urban Heat Wave (UHW) phenomena as planning challenges. This framing is consistent with literature showing that UHI can increase heat-related health risk in urban areas and that heat waves have been associated with higher mortality in European cities [12, 4].

Urban green infrastructure is one family of interventions used to address these challenges. In the TALEA project, the relevant intervention unit is the TALEA Green Cell (TGC), described in the original report as a modular and adaptable green element intended to reconnect fragmented green spaces, regenerate underused areas, and create local climate refuges. This thesis adopts the terminology and baseline problem definition of the original TALEA project, while keeping the contribution of this thesis separate from that original system.

1.2 Optimization-Based Decision Support

Urban planning decisions often depend on heterogeneous spatial data: land use, existing vegetation, roads, buildings, demographic indicators, and environmental indices. Planning support systems and GIS-based decision-analysis literature describe how spatial data, models, and display or evaluation tools can support planning tasks without replacing professional judgement [10, 13]. The original TALEA decision support system (DSS) uses these data to construct a spatial optimization problem over a grid representation of Bologna. The DSS therefore produces candidate locations for new Green Cells according to explicit utility functions and parameter choices.

The baseline system is particularly relevant for this thesis because its recommendations are generated by optimization models. Mathematical-programming models are built from explicit decision variables, objectives, and constraints [15]. In the inspected implementation, this makes the system suitable for technical analysis: its decisions can be traced to input data, preprocessing choices, and mathematical formulations.

1.3 Case Study and Thesis Scope

The case study considered in this thesis is the Bologna Green Cells planning problem developed within the TALEA project. The original project formulates the placement of TGCs as a Constraint Optimization Problem (COP), implemented in MiniZinc and solved using the HiGHS Mixed Integer Programming (MIP) solver. The planning surface is discretized into grid cells, and each cell is enriched with macro-scale and micro-scale indicators before the optimization models are solved.

The contribution of this thesis is not to redesign the TALEA DSS, redefine the Green Cell concept, or replace the baseline optimization models. Instead, the thesis first reconstructs the academic and technical foundation of the baseline system and then assesses how the optimization-based recommendations change when selected input data are affected by controlled uncertainty.

The robustness assessment is grounded in the available Python implementation and result files. It perturbs selected data sources, re-executes the optimization models under those perturbed scenarios, and compares the perturbed outputs with the corresponding no-noise baseline. This keeps the analysis close to the implemented DSS rather than treating robustness as an abstract property detached from the code.

1.4 Research Objective

The overall objective of the thesis is to assess the robustness of optimization-based decision support for sustainable urban planning using the Bologna Green Cells case study. In order to do so rigorously, Chapters 2–5 establish the baseline system on which the later assessment depends. This includes clarifying which data are used, how they are transformed into model inputs, what each optimization model actually implements, and where the implementation differs from the mathematical description in the original project report.

More specifically, the thesis asks how the implemented recommendations change under controlled perturbations of environmental, demographic, and spatial inputs; which models and input sources appear more sensitive within the tested scenarios; and what the implemented robustness metrics can and cannot say about the stability of Green Cell placement.

1.5 Research Questions

The robustness assessment is organized around three research questions:

- **RQ1:** How much do the selected Green Cell locations deviate from the no-noise baseline under synthetic perturbations of the input data sources?

- **RQ2:** Which model-source-perturbation combinations produce the largest spatial or quantitative deviations from the baseline?
- **RQ3:** How do spatial similarity metrics and quantitative output-ratio metrics converge or diverge in diagnosing robustness or fragility?

RQ1 concerns spatial deviation relative to the no-noise baseline, measured by changes in selected cell footprints and nearby-cell matches. RQ2 concerns model-source-perturbation sensitivity and includes both spatial and quantitative deviations. RQ3 concerns convergence or divergence between spatial metrics and output-ratio metrics, where the latter are evaluated through utility, yard-allocation, and street-allocation ratios.

1.6 Thesis Organization

Chapter 2 introduces the conceptual background required to understand the thesis, including sustainable urban planning, urban green infrastructure, environmental indicators, and optimization-based decision support. Chapter 3 describes the original TALEA project and the Bologna Green Cells case study. Chapter 4 reconstructs the baseline data and preprocessing workflow. Chapter 5 presents the implemented optimization models and compares them with the formulations reported in the original project report.

Chapter 6 defines the robustness-evaluation methodology. Chapter 7 describes the perturbation and evaluation implementation. Chapter 8 presents the observed robustness metrics and selected plots. Chapter 9 interprets those results cautiously in relation to the model formulations and perturbed variables, and Chapter 10 summarizes the findings and limitations.

Chapter 2

Background and Related Work

2.1 Sustainable Urban Planning and Climate Adaptation

Sustainable urban planning concerns the organization of urban space in ways that account for environmental, social, and infrastructural objectives. In climate-adaptation terms, cities are central sites of risk reduction, infrastructure planning, and uneven vulnerability [7, 8]. In the context of the TALEA baseline project, this broad planning concern is narrowed to the placement of new green interventions in Bologna. The original TALEA report by Di Buò and Reale [6] motivates the problem through climate change, biodiversity loss, pollution, and unequal access to public space.

Climate adaptation is relevant to this thesis because the original DSS is not merely a geometric placement tool. It combines urban form, existing green coverage, population density, and heat-related indicators into optimization models that prioritize areas according to explicit utility functions. The technical question addressed later by the thesis depends on this planning context: if a DSS uses spatial and environmental data to support climate-sensitive decisions, then the reliability of those recommendations is linked to the quality and uncertainty of the underlying data.

2.2 Urban Green Infrastructure

Urban green infrastructure refers to the planned use of vegetation and related natural or semi-natural spaces within cities, understood as part of a wider urban green-space system with ecological and human-health functions [14]. Nature-based-solutions literature similarly frames urban nature as a way to connect adaptation, ecosystem services, and social co-benefits [9]. The original report frames Green Cells as modular interventions that can reconnect fragmented green areas and provide local climate refuges. This thesis uses that project-specific concept in Chapter 3; the present section only establishes the general conceptual role of green infrastructure.

The original TALEA project also treats some blue spaces as part of the baseline green-space layer. According to the report, river and water-covered areas are collapsed with green spaces because the available data and project assumptions did not require a separate treatment of blue and green areas. This is a modelling choice of the baseline project, not a general claim about urban ecological planning.

2.3 Green Accessibility and Green Justice

The report states that TALEA aims to promote "green justice" by targeting vulnerable areas and involving residents in planning and maintenance. This establishes an equity-oriented motivation for the project, but the available repository material does not provide enough evidence to define or operationalize green justice as an academic concept within this thesis.

Academic literature on urban green space treats unequal access to green space as an environmental justice issue and warns that greening strategies can have distributional consequences if equity is not considered explicitly [16].

For the foundation chapters, the term is therefore used only when describing the original TALEA project report. The implemented optimization models do not contain a direct variable named green justice, social vulnerability, or accessibility. They instead use population density, existing green-space coverage, UHEI, and micro-scale spatial variables as the implemented prioritization criteria.

2.4 Urban Heat and Environmental Indicators

The Urban Heat Island effect refers to the tendency for urbanized areas to exhibit higher heat exposure than surrounding or less built-up environments, with implications for heat-related health risk [12]. Heat-wave research in European cities also documents associations between heat-wave events and mortality [4]. The original TALEA report links UHI and UHW phenomena to climate risk in Bologna and uses heat-related indicators to prioritize interventions. The implemented system includes the Urban Heat Exposure Index (UHEI) and the Normalized Difference Vegetation Index (NDVI) as environmental indicators.

The report describes UHEI as an index showing how heat accumulates and persists in different areas of the city. It also states that UHEI is a composition of other indicators, including land surface temperature, NDVI, and albedo. The repository contains UHEI as a spatial input layer under the raw data index directory, and the grid-creation code aggregates it onto the model grid.

2.5 UHEI, NDVI, and Fractional Vegetation Cover

NDVI is used in the baseline system as an indicator more directly related to vegetation. The report states that the NDVI values available for this study are normalized in the range from 0 to 0.8, and the implemented `ndvi` model fixes `ndvi_norm_min`=0.0 and `ndvi_norm_max`=0.8 in the Python instance-generation code.

The connection between NDVI and Fractional Vegetation Cover (FVC) is supported in the baseline report through Carlson and Ripley [1], Choudhury et al. [2], and Gillies and Carlson [11]. These works are used by the report to motivate a relation between scaled NDVI and vegetation cover. The report also cites de la Iglesia Martínez and Labib [5] to justify caution when interpreting NDVI across different areas of interest.

This thesis does not generalize beyond that evidence. It uses the cited literature only to explain why the original project introduces NDVI and FVC-inspired model terms. The implemented formulas are analyzed separately in Chapter 5, where the MiniZinc files are treated as the source of truth for the actual optimization models.

2.6 Spatial Data and Indicators for Urban Planning

Spatial decision support relies on the transformation of heterogeneous data into comparable indicators. GIS-based decision-analysis and planning-support literature emphasizes the role of spatial data, modelling, visualization, and evaluation tools in structuring planning problems [13, 10]. In the TALEA baseline system, the transformation includes selecting the Bologna study area, creating a regular grid, computing overlaps between the grid and urban layers, estimating population density per statistical area, and aggregating UHEI and NDVI to the model cells.

This thesis distinguishes between raw spatial data, intermediate derived layers, and optimization parameters. That distinction is important because the implemented models do not consume raw maps directly; they consume numerical arrays written to MiniZinc data files after preprocessing.

2.7 Decision Support Systems in Urban Planning

A decision support system for urban planning can be understood as a computational aid that organizes data, models alternatives, and supports decision-makers in comparing possible interventions. Planning support systems are described as computer-based geo-information instruments that combine data, algorithms, modelling capabilities, and display facilities to support planning activities [10]. The TALEA baseline DSS supports the choice of candidate Green Cell locations by ranking grid cells through optimization models.

The original report emphasizes transparency and interpretable optimization for the TALEA baseline system. More generally, planning-support and mathematical-programming literature treats models as explicit structures for organizing data, alternatives, decision variables, objectives, and constraints [10, 13, 15]. The interpretability claim in this thesis is therefore limited to the inspected implementation: its recommendations can be traced through written MiniZinc variables, constraints, utility functions, and input parameters rather than through opaque predictions.

2.8 Optimization-Based and Interpretable Decision Support

Optimization-based decision support formulates planning alternatives through decision variables, objectives, and constraints, following the basic structure of mathematical-programming model building [15]. In the TALEA implementation, each grid cell has decision variables for the amount of new Green Cell area assigned to street space and yard-like free space. The models maximize a mean utility over selected cells, constrained by available space and by a maximum number of active cells.

The interpretability of the TALEA baseline system is implementation-level interpretability: the utility functions, constraints, and input parameters are written explicitly in MiniZinc. Chapter 5 documents these formulations directly from the source files.

2.9 Spatial Optimization and Location-Allocation Concepts

The TALEA baseline problem resembles spatial selection and location-allocation problems because it asks which spatial units should receive new interventions under explicit limits. Classic location-allocation literature formulates related questions about selecting locations to serve or cover spatial demand under constraints [3]. In the implementation, the candidate locations are grid cells, and the selected cells are those with non-zero `street_cells` or `yard_cells`.

This connection is used cautiously. The implemented TALEA models are documented as the specific optimization formulations contained in the repository rather than as canonical instances of a named location-allocation model.

Chapter 3

The TALEA Project and the Bologna Green Cells Case Study

3.1 Project Context

The original TALEA project is presented by Di Buò and Reale [6] as a response to urban climate and sustainability challenges in Bologna. Their report states that the project is supported by the European Urban Initiative – Innovative Actions (EUI-IA), introduces TALEA Green Cells as modular green units intended to reconnect fragmented green areas, regenerate underused spaces, and create climate refuges, and formulates the placement problem as the search for the best n Green Cell locations according to a utility function.

The inspected repository is consistent with this description and implements a transparent methodology for planning TGC placement in Bologna using interpretable optimization techniques. This thesis treats the report by Di Buò and Reale and the inspected repository as project evidence. Because no external official TALEA or EUI-IA project source has been verified for this thesis, institutional-support details are treated as claims from the local project documentation rather than independently documented facts.

3.2 Bologna Green Cells Case Study

The report describes the case study as focusing on the urban area of Bologna. It identifies three pilot zones: Martiri-Boldrini, Mascarella, and Savena. These zones are presented in the report as examples of different applications of the Green Cell model, ranging from dense mobility corridors to peripheral areas with self-establishing vegetation.

Inspection of the implementation shows that the baseline DSS does not directly model those pilot zones as hard constraints in the MiniZinc formulations inspected for this thesis. Instead, it builds a city-scale or center-scale grid, computes indicators for each cell, and selects cells according to model-specific utility functions. This distinction is important: the pilot-zone discussion belongs to the project narrative in the report, while the implemented optimization system works over computed grid-cell attributes.

3.3 Planning Problem Addressed by the Baseline DSS

The baseline planning problem is to identify suitable locations for new TGCs. Inspection of the implementation shows a corresponding selection budget represented by the parameter `top_k` in MiniZinc and `max_cells` in the execution wrapper.

The planning decision supported by the system is therefore not the complete engineering design of a Green Cell, nor the legal feasibility of acquiring or transforming a parcel. It is a prioritization decision: which grid cells should be selected as candidate locations, and how much intervention area should be assigned to street and yard-like spaces in those cells.

3.4 Intended Outputs

Inspection of the solver module shows that the implemented solver writes GeoJSON outputs containing the selected grid-cell geometries and the fields `street_cells`, `yard_cells`, and `utility`. The repository documentation indicates that these outputs are intended to be spatially visualized, as reflected by the React/Leaflet frontend and the map-based interpretation emphasized in the original report.

The expected output of the baseline DSS can therefore be described as a spatial recommendation layer. It contains candidate cells for TGC placement, the model-assigned amount of new green intervention on street and yard-like areas, and the utility value associated with each selected cell.

3.5 Baseline Workflow

Figure 3.1 summarizes the baseline workflow reconstructed from the original report and from the implementation. The report supplies the project-level description of the workflow, while inspection of the repository implementation specifies the executable sequence that transforms raw data into optimization outputs.

Open urban, demographic, and environmental data

$\downarrow$

GIS preprocessing and 100 m grid construction

$\downarrow$

Macro-scale and micro-scale indicator computation

$\downarrow$

MiniZinc instance generation

$\downarrow$

Optimization model solution

$\downarrow$

GeoJSON layer of selected Green Cell candidate locations

Figure 3.1: Schema of the baseline TALEA decision-support workflow reconstructed from the project report and the implementation files.

The workflow begins from raw spatial and statistical datasets. It creates an analysis grid, computes the spatial overlap between grid cells and urban layers, estimates macro-scale factors at the level of statistical areas, writes MiniZinc data files, and solves one of the implemented models. The selected cells are then exported as GeoJSON files.

3.6 Assumptions and Limitations of the Baseline Project

The original report identifies several simplifying assumptions. The free-space concept is defined as the remaining space after removing buildings, roads, human infrastructure, and existing green space; in many cases the report interprets this as yard court space. Roads are also considered potential intervention spaces, but highways and large squares are excluded to make this approximation more realistic.

The report also notes that the baseline models omit information such as budget, the topology of new Green Cells, and the structure of paths connecting green areas. It further states that the absence of some real-world constraints may limit direct feasibility. These limitations belong to the original baseline project and motivate careful interpretation of its outputs.

Later thesis phases will use the baseline DSS as the object of analysis, but the baseline assumptions described here are not themselves findings.

Chapter 4

Data and Preprocessing

4.1 Overview

The baseline TALEA DSS is built from heterogeneous spatial, environmental, and demographic data. The original report describes the main data categories and their planning motivation; the repository implementation defines how those data are converted into the numerical arrays used by the optimization models. This chapter describes the baseline data pipeline only.

The final optimization dataset is a grid-based GeoJSON layer in which each cell contains model inputs such as existing green area, road area, free-space area, number of free-space fragments, tree count, weighted UHEI, weighted NDVI, and derived inverse UHEI. Figure 4.1 summarizes this transformation.

The baseline data inventory is reconstructed from the original TALEA report by Di Buò and Reale [6] and from the inspected implementation files. Tables 4.1 and 4.2 list the baseline raw data categories, while metadata such as acquisition date, coordinate reference system, or spatial resolution are marked as unavailable or requiring verification when they are not explicit in the inspected sources.

Raw spatial and statistical datasets

↓

QGIS/PyQGIS cleaning, clipping, difference, union,
and overlap operations

↓

Regular grid and statistical-area intersections

↓

Macro indicators and micro indicators

↓

`final_grid` layer and MiniZinc data arrays

Figure 4.1: Schema of the data-to-model pipeline for the baseline TALEA DSS, reconstructed from the project report and the implementation.

Table 4.1: Raw spatial and tabular datasets used by the baseline TALEA system: file, format, and CRS metadata.

Dataset	Inspected file	Nature / format	CRS metadata
Land use	`uso_suolo_bologna.geojson`	Spatial polygon GeoJSON	File declares OGC CRS84
Municipal green	`un_gest.fgb`	Spatial polygon FGB	File metadata indicates WGS 84 / EPSG:4326
Private green	`verde_privato_urbanizzato.fgb`	Spatial polygon FGB	File metadata indicates WGS 84 / EPSG:4326
Forest areas	`aree-boschive.gpkg`	Spatial polygon GPKG layers	GPKG metadata indicates EPSG:4326 for used layers
River / water areas	`aree-fluviali.gpkg`	Spatial GPKG; implementation uses polygon layer	GPKG metadata indicates EPSG:4326 for used layer
Buildings	`rifter_edif_pl.geojson`	Spatial polygon GeoJSON	File declares EPSG:3857
Street areas	`aree-stradali.geojson`	Spatial polygon GeoJSON	No explicit CRS member in inspected file
Trees	`alberi-manutenzioni.fgb`	Spatial point FGB	File metadata indicates WGS 84 / EPSG:4326

Dataset	Inspected file	Nature / format	CRS metadata
Statistical areas	`aree-statistiche.geojson`	Spatial polygon GeoJSON	No explicit CRS member in inspected file
Inhabited localities	`localita_abitative.gpkg`	Spatial polygon GPKG layer	GPKG metadata indicates EPSG:4326
Population by statistical area	`popolazione_per_area_statistica.csv`	Tabular CSV	Not spatial; linked by statistical-area code
UHEI	`urban_heat_exposure_index.geojson`	Spatial polygon GeoJSON index	File declares OGC CRS84
NDVI	`ndvi_mean.geojson`	Spatial polygon GeoJSON index	File declares OGC CRS84

Table 4.2: Raw dataset source status and baseline role in the TALEA decision-support pipeline.

Dataset	Source, date, and resolution status	Baseline role
Land use	Geoportale Emilia-Romagna, as reported in the original TALEA report; report states database year 2020; original resolution not documented	Select residential free-space base; identify highways and railways
Municipal green	Open Data Bologna, as reported in the original TALEA report; acquisition date and resolution not documented	Existing green-space layer
Private green	Open Data Bologna, as reported in the original TALEA report; acquisition date and resolution not documented	Existing green-space layer
Forest areas	Geoportale Emilia-Romagna, as reported in the original TALEA report; acquisition date and resolution not documented	Additional green-space layer

Dataset	Source, date, and resolution status	Baseline role
River / water areas	Geoportale Emilia-Romagna, as reported in the original TALEA report; acquisition date and resolution not documented	Treated together with green areas in the baseline project
Buildings	Open Data Bologna, as reported in the original TALEA report; acquisition date and resolution not documented	Removed from candidate free space
Street areas	Open Data Bologna, as reported in the original TALEA report; acquisition date and resolution not documented	Candidate street-space intervention area after cleaning
Trees	Open Data Bologna, as reported in the original TALEA report; acquisition date and resolution not documented	Count trees outside existing green areas
Statistical areas	Open Data Bologna, as reported in the original TALEA report; acquisition date and resolution not documented	Macro-scale aggregation unit and center-area definition
Inhabited localities	Source not clearly documented in the report or repository documentation; acquisition date and resolution not documented	City-wide clipping before grid construction
Population by statistical area	Bologna metropolitan statistical data portal, as reported in the original TALEA report; acquisition date not documented	Compute population density and macro factor
UHEI	TALEA UHI analysis described in the original report; source resolution and acquisition date require verification	Heat-related macro and model input

Dataset	Source, date, and resolution status	Baseline role
NDVI	TALEA UHI analysis described in the original report; source resolution and acquisition date require verification	Vegetation indicator for the NDVI model

4.2 Spatial Unit of Analysis

The implementation creates a regular grid with horizontal and vertical spacing of 100 meters. For the city-wide Bologna case, the grid is clipped to the inhabited area extracted from the localities layer and filtered statistical areas. For the center case, the grid is clipped to a subset of central statistical areas. The grid geometry is created in EPSG:3857 in the implementation.

The grid is the unit on which the optimization models operate. Each candidate cell is assigned an identifier, geometric area, and derived indicators. Inspection of the saved processed files shows that the grid-level and solver-output GeoJSON files for both the center and city-wide cases declare EPSG:3857, while several non-grid intermediate layers keep CRS84 metadata and one railway intermediate lacks an explicit CRS member in the inspected copy. The thesis therefore treats EPSG:3857 as the grid and output CRS, not as a universal CRS claim for every intermediate file.

4.3 Preprocessing Operations

The baseline preprocessing combines GIS operations and tabular computations. Conceptually, the main operations are:

- cleaning and merging green and blue spatial layers into a unified green-space layer;

- removing highways and large square-like street elements from the street layer;
- defining free space from residential land-use classes after subtracting existing green space, roads, buildings, and railways;
- creating the 100 m grid and intersecting it with statistical areas;
- computing overlap areas between grid cells and green, building, road, free-space, and railway layers;
- counting free-space fragments and trees outside existing green spaces;
- aggregating UHEI and NDVI values onto grid cells;
- computing population density and the macro utility factor.

These operations are implemented primarily by the grid-creation, preprocessing, and utility modules. The chapter avoids a line-by-line code description because the research purpose is to document the data design, not every processing command.

4.4 Macro-Scale and Micro-Scale Indicators

The baseline system separates macro-scale and micro-scale indicators. The macro scale is associated with statistical areas and captures broader planning priority. The micro scale is associated with grid cells and captures the local spatial feasibility and existing conditions of each candidate cell.

Macro scale	**Micro scale**
Statistical areas	100 m grid cells
Population density	Road space
Existing green shortage	Free / external space
Average UHEI	Green area, trees, fragmentation

Figure 4.2: Schema of the macro-scale and micro-scale indicator structure of the baseline DSS.

The macro factor is computed from normalized population density, lack of green space, and UHEI using weights `density_param`, `green_param`, and `uhei_param`. The implementation requires these weights to sum to one. The micro indicators enter the model-specific utility functions described in Chapter 5.

4.5 Derived Variables

The implementation distinguishes raw datasets, processed spatial layers, derived indicators, and final model parameters. Table 4.3 lists the main processed spatial layers inspected in the baseline path, while Table 4.4 lists the principal derived variables that become model inputs or solver outputs.

Table 4.3: Processed spatial layers used before MiniZinc instance generation.

Processed layer	Inspected layer / file	Created from	Category
Full inhabited area	`area_abitata_full.geojson`	Statistical areas clipped with inhabited localities, then filtered	Processed study-area layer
Unified green layer	`verde.gpkg`	Municipal green, private green, forest layers, river/water layer, and Giardini Margherita area	Derived spatial layer
Cleaned street layer	`aree-stradali-modificate.geojson`	Street areas after square/highway filtering and removal of green and railway overlaps	Derived spatial layer

Processed layer	Inspected layer / file	Created from	Category
Free-space layer	free_space.geojson	Residential land-use areas after removing green, streets, buildings, and railways	Derived spatial layer
Grid with area overlaps	grid_with_areas.geojson	100 m grid with overlaps against green, building, road, free-space, and railway layers	Intermediate model-input layer
Final grid	final_grid.geojson	Grid with areas, UHEI, NDVI, tree counts, and inverse UHEI	Final model-parameter layer

Table 4.4: Derived-variable inventory for the baseline model inputs and outputs.

Variable	Derived from	Created / used by	Role
density	Population and statistical-area area	Preprocessing and utility modules	Macro population factor
macro_utility_factor	Density, green percentage, UHEI	Preprocessing module	Multiplicative macro factor
green_area	Unified green layer overlap	Grid-creation stage	Existing green-space penalty / space constraint
road_area	Cleaned road layer overlap	Grid-creation stage	Available street intervention space
free_space_area	Free-space layer overlap	Grid-creation stage	Available yard-like intervention space

Variable	Derived from	Created / used by	Role
`free_space_number`	Free-space fragment count	Grid-creation stage	Fragmentation penalty / denominator term
`tree_number`	Trees outside green areas	Grid-creation stage	Existing vegetation proxy
`weighted_uhei`	UHEI-grid intersections	Grid-creation stage	Heat indicator for macro and `diff` model
`weighted_ndvi`	NDVI-grid intersections	Grid-creation stage	Vegetation indicator for `ndvi` model
`inverse_uhei`	Max UHEI minus weighted UHEI	Grid-creation and parser stages	Input percentile basis for `inverse_uhei`
`street_cells`	Solver decision variable	MiniZinc models and solver output	Amount assigned to street space
`yard_cells`	Solver decision variable	MiniZinc models and solver output	Amount assigned to free / yard-like space
`utility`	Model-specific utility equation	MiniZinc models and solver output	Objective contribution and output attribute

4.6 Dataset, Code, and Parameter Mapping

Table 4.5 maps the main data sources to preprocessing components and model parameters.

Table 4.5: Mapping between datasets, preprocessing functions, and model parameters.

Data category	Main code evidence	Derived model fields	Dependent model parameters
Population	Density estimation and cell-level density weighting	`density`, `macro_factors`	`density_param`
Green and blue areas	Green-layer construction and overlap computation	`green_area`, `green_pc`	`green_param`; utility denominators
Land use	Residential selection, highway extraction, and free-space derivation	`free_space_area`, railway and road exclusions	Space feasibility parameters
Streets	Street cleaning and overlap computation	`road_area` / `street_space`	`beta_streets`
Free space	Spatial difference and fragment count	`free_space_area`, `free_space_number`	`alpha_yard`; fragmentation terms
Trees	Trees-outside-green count	`tree_number`	Utility denominator or penalty terms
UHEI	Weighted aggregation and inverse-UHEI percentile construction	`weighted_uhei`, `inverse_uhei`	`uhei_param`; `alpha_uhei`; inverse UHEI percentiles
NDVI	Weighted aggregation and parser constants	`weighted_ndvi`	`ndvi_norm_max`, `ndvi_norm_min`

4.7 Missing Metadata and Data Limitations

Table 4.6 records missing metadata and limitations that should be resolved before final submission.

Table 4.6: Missing metadata and baseline data limitations.

Issue	Current evidence	Required follow-up
Raw CRS metadata	File-level CRS metadata is available for some raw files, but several GeoJSON files have no explicit CRS member in the inspected repository copy	Verify official CRS metadata for each raw source file
Dataset acquisition dates	Report states land-use database refers to 2020; other acquisition dates are unclear	Verify source metadata from official portals
UHEI and NDVI generation details	The original report describes the TALEA UHI analysis and source satellite data generally	Verify exact UHI processing documentation
Spatial resolution of source indices	Grid resolution is 100 m; source UHEI/NDVI cell size is not fully documented in inspected text	Verify source index resolution
Green and blue aggregation assumption	Report collapses green and water areas for this study	Clarify implications and cite supporting planning literature if generalized
Free-space interpretation	Report interprets free space as often corresponding to yard courts	Avoid claiming legal or construction feasibility without further data

Chapter 5

Optimization Models

5.1 Implementation Sources and Modelling Scope

The baseline TALEA repository implements four MiniZinc model files: `std`, `diff`, `inverse_uhei`, and `ndvi`. The parser builds the corresponding MiniZinc data instances from the final grid, while the solver exports the selected grid cells as GeoJSON.

This chapter treats the MiniZinc files as the source of truth for the implemented formulations. The original report is used to explain the conceptual motivation of the models, but differences between the report and the implementation are documented explicitly in Section 5.9.

5.2 Sets, Parameters, and Decision Variables

Let $I = \{1, \ldots, n\}$ be the set of grid cells. The implementation names n as `len`. The notation used in this chapter is summarized in Table 5.1.

Table 5.1: Notation for the implemented optimization models.

Symbol	Meaning
$i \in I$	Grid-cell index

Symbol	Meaning
K	Maximum number of active cells, implemented as `top_k`
S_i	Street space in cell i, implemented as `street_space`
E_i	External or free-space area in cell i, implemented as `ext_space`
G_i	Existing green-space area in cell i, implemented as `green_space`
A_i	Number of free-space fragments in cell i, implemented as `num_areas`
T_i	Number of trees in cell i, implemented as `num_trees`
M_i	Macro factor in cell i, implemented as `macro_factors`
F_i	Full available grid-cell area used by the model, implemented as `full_space`
H_i	Weighted UHEI in cell i, implemented as `uhei` in the `diff` model
N_i	Weighted NDVI in cell i, implemented as `ndvi` in the `ndvi` model
$\bar{S}, \bar{E}, \bar{A}, \bar{T}, \bar{G}, \bar{H}$	Maximum values of the corresponding arrays, computed inside the MiniZinc models
β	Street-space availability parameter, implemented as `beta_streets`
α	Yard-space parameter, implemented as `alpha_yard`
α_H	UHEI parameter used only by the `diff` model, implemented as `alpha_uhei`
I_{95}, I_5	95th and 5th percentiles of `inverse_uhei`, computed in Python for the `inverse_uhei` model
$N_{\max}, N_{\min}$	NDVI normalization constants, implemented as `ndvi_norm_max`=0.8 and `ndvi_norm_min`=0.0
s_i	Continuous MiniZinc decision variable `street_cells[i]`, declared as $\text{var } 0..\bar{S}$

Symbol	Meaning
y_i	Continuous MiniZinc decision variable `yard_cells[i]`, declared as $\text{var } 0..\bar{E}$
u_i	Continuous MiniZinc decision variable `utility[i]`, declared as var float

The Python parser adds one to `green_area` and `tree_number` before writing them as model arrays. It also conditionally adds one to `free_space_number` when street space is positive. These implementation choices should be preserved in the final technical description because they affect the numerical arrays passed to MiniZinc.

5.3 Common Objective and Constraints

All four models maximize the same objective structure:

$$\max \quad \frac{1}{K}\sum_{i\in I} u_i. \tag{5.1}$$

The denominator is the fixed parameter K, not the realized number of active cells. The active-cell limit is implemented as:

$$\sum_{i\in I} \mathbf{1}\left(s_i \neq 0 \vee y_i \neq 0\right) \leq K. \tag{5.2}$$

All models also include street-space and total-space feasibility constraints:

$$s_i \leq \beta S_i \qquad \forall i \in I, \tag{5.3}$$

$$s_i + y_i + G_i \leq F_i \qquad \forall i \in I. \tag{5.4}$$

The yard-space constraint differs by model. In `std` and `diff`, the implementation uses:

$$y_i \leq E_i \quad \forall i \in I. \tag{5.5}$$

In `inverse_uhei` and `ndvi`, the implementation uses:

$$y_i \leq \alpha E_i \quad \forall i \in I. \tag{5.6}$$

5.4 Standard Model

The `std` model is the baseline utility formulation implemented in `std_model.mzn`. It rewards available yard and street allocations and penalizes existing green space, trees, and fragmentation. The implemented utility is:

$$u_i = \begin{cases} M_i \dfrac{\alpha \dfrac{y_i}{\bar{E}} + \dfrac{s_i}{\bar{S}}}{\dfrac{G_i}{\bar{G}} + \dfrac{T_i}{\bar{T}} + 10^{-8}}, & A_i = 0, \\ M_i \dfrac{\alpha \dfrac{y_i}{\bar{E}} + \dfrac{s_i}{\bar{S}}}{\dfrac{A_i}{\bar{A}} + \dfrac{G_i}{\bar{G}} + \dfrac{T_i}{\bar{T}} + 10^{-8}}, & A_i \neq 0. \end{cases} \tag{5.7}$$

Urban-planning interpretation: the model assigns higher utility to cells where new intervention space is available and where existing green, tree presence, and free-space fragmentation are lower. The small constant 10^{-8} is an implementation safeguard in the denominator.

Limitation: if normalization maxima are zero, the MiniZinc formulation does not define an explicit safeguard for the corresponding division. The implementation appears to rely on the data having positive maxima.

5.5 Difference Model

The `diff` model is a separate utility variant implemented in `diff_model.mzn`. It uses a difference rather than a ratio between positive allocation terms and existing-condition terms. The implemented utility is:

$$u_i = \begin{cases} M_i \left[\alpha \frac{y_i}{\bar{E}} + \frac{s_i}{\bar{S}} + \alpha_H \frac{H_i}{\bar{H}} - \left(\frac{G_i}{\bar{G}} + \frac{T_i}{\bar{T}} \right) \right], & A_i = 0, \\ M_i \left[\alpha \frac{y_i}{\bar{E}} + \frac{s_i}{\bar{S}} + \alpha_H \frac{H_i}{\bar{H}} - \left(\frac{A_i}{\bar{A}} + \frac{G_i}{\bar{G}} + \frac{T_i}{\bar{T}} \right) \right], & A_i \neq 0. \end{cases} \tag{5.8}$$

Urban-planning interpretation: this model treats intervention space, UHEI, existing green, trees, and fragmentation as additive terms with positive or negative signs. It can therefore produce negative utility values when penalties dominate. This is a meaningful implementation difference from the ratio-based `std` model.

Assumption: the implemented model includes UHEI as a positive term scaled by α_H. Whether that term should be interpreted as an incentive to prioritize high heat-exposure cells depends on the chosen value of α_H and on the configuration actually passed to the solver.

5.6 Inverse UHEI Model

The `inverse_uhei` model is implemented in `inverse_uhei_model.mzn`. The grid-creation stage computes:

$$\text{inverse\_uhei}_i = \max_j(H_j) - H_i, \tag{5.9}$$

and then passes the 5th and 95th percentiles of this derived array to MiniZinc. The implemented utility is:

$$u_i = \frac{M_i}{\frac{A_i}{\bar{A}} + \frac{G_i}{\bar{G}} + \frac{T_i}{\bar{T}}} \cdot \frac{s_i + y_i}{F_i} \cdot (I_{95} - I_5). \tag{5.10}$$

This model differs from `std` and `diff` by using a percentile range derived from inverse UHEI as a scaling factor and by using $(s_i + y_i)/F_i$ as the intervention share. The implemented yard constraint is $y_i \leq \alpha E_i$.

Limitation: unlike the `std` model, the denominator in Equation 5.10 does not include an epsilon term. The implementation does not explicitly guard against a zero denominator.

5.7 NDVI Model

The `ndvi` model is implemented in `ndvi_model.mzn`. It uses the weighted NDVI value of each grid cell and the constants $N_{\max} = 0.8$ and $N_{\min} = 0.0$. The implemented utility is:

$$u_i = \frac{M_i}{N_i + \frac{A_i}{\bar{A}}} \cdot \left[\left(0.5 + 0.5 \frac{s_i + y_i}{F_i} \right) (N_{\max} - N_{\min}) + N_{\min} \right]. \tag{5.11}$$

The model is conceptually motivated by the relationship between NDVI and vegetation cover discussed in the original report and supported by Carlson and Ripley [1], Choudhury et al. [2], Gillies and Carlson [11], and de la Iglesia Martínez and Labib [5]. However, Equation 5.11 is the implemented linearized form, not the full square-root relation described in the background literature.

Implementation note: `green_space` appears in the space constraint, while `num_trees` is passed as an input but is not used in the `ndvi` utility expression.

5.8 Model Family Comparison

Figure 5.1 summarizes the relationship between the four formulations.

Table 5.2 compares the model inputs and distinctive features.

`std`	`diff`	`inverse_uhei`	`ndvi`
Ratio utility	Difference utility	Inverse-UHEI scaling	NDVI-based scal-ing
Green/tree/fragment penalty	Additive penalty	Percentile range	Linearized FVC inspiration

Figure 5.1: Comparison of the implemented model families for the baseline DSS.

Table 5.2: Comparison of the four implemented optimization models.

Model	Formulation type	Distinctive inputs	Main interpretation
`std`	Ratio-based base-line variant	$S_i, E_i, G_i, A_i, T_i, M_i, F_i$	Prioritize avail-able intervention space where exist-ing green, trees, and fragmentation are low
`diff`	Additive difference variant	Standard inputs plus H_i and α_H	Balance positive allocation terms against existing-condition penalties and UHEI term
`inverse_uhei`	Inverse-UHEI per-centile variant	I_{95}, I_5 from inverse UHEI	Scale intervention share by an observed inverse-UHEI range
`ndvi`	NDVI-based variant	$N_i, N_{\max}, N_{\min}$	Use NDVI and in-tervention share to represent vegetation-related benefit

5.9 Differences Between the Reported and Implemented Formulations

The comparison below contrasts the mathematical description in the original TALEA report [6] with the inspected MiniZinc implementation. The report and the implementation are broadly aligned in purpose, but they do not match perfectly in every mathematical detail. Table 5.3 documents the main discrepancies identified during this phase.

Table 5.3: Differences between the reported and implemented formulations.

Topic	Report formulation / statement	Implementation formulation	Evidence
Standard model name	The report's model section emphasizes Difference, Inverse UHEI, and NDVI after introducing the general model	A distinct `std_model.mzn` file is implemented	Original report, Section 3; MiniZinc model files
Street coefficient in formulas	Report formulas display $\beta_{\text{streets}} \cdot$ street_cells inside utility expressions	MiniZinc utility expressions use $s_i/\bar{S}$; β appears in the constraint $s_i \le \beta S_i$	Original report, Section 3.2; MiniZinc model files
Yard coefficient in constraints	Report text states α_{yard} regulates yard-space impact; formulas place it inside utility terms	In `std` and `diff`, $y_i \le E_i$; in `inverse_uhei` and `ndvi`, $y_i \le \alpha E_i$	MiniZinc model files
Difference UHEI sign	Report prose says existing UHEI is among terms decreasing utility in the difference utility	MiniZinc adds $+\alpha_H H_i/\bar{H}$ inside the positive part of the expression	Original report, Section 3.2.1; `diff_model.mzn`

Topic	Report formulation / statement	Implementation formulation	Evidence
Inverse UHEI numerator	Report formula shows $\alpha_{\text{yard}} \cdot y_i + \beta_{\text{streets}} \cdot s_i$ over full space	MiniZinc uses $(s_i + y_i)/F_i$, with α and β enforced through constraints	Original report, Section 3.2.2; `inverse_uhei_model.mzn`
NDVI/FVC relation	Report motivates the NDVI model through a square-root relation between FVC and scaled NDVI, then states that a first-order linear approximation is used	MiniZinc implements the linear expression in Equation 5.11	Original report, Section 3.2.3; `ndvi_model.mzn`
Tree input in NDVI model	Report states NDVI substitutes the sum of green space and trees in the previous utility functions	MiniZinc passes `num_trees` but does not use it in the NDVI utility expression	Original report, Section 3.2.3; `ndvi_model.mzn`; parser module
Active-cell objective denominator	Report describes finding the best n cells	Implementation maximizes average utility divided by K, regardless of the realized number of active cells	Original report, Section 3; MiniZinc model files

The table does not imply that the report or implementation is conceptually wrong. It records differences that must remain visible in the thesis because later analysis should refer to the implemented system, not an idealized formulation.

Chapter 6

Robustness Evaluation Methodology

6.1 Evaluation Objective

The robustness evaluation studies how stable the implemented Green Cell recommendations are when selected input data are perturbed in a controlled way. Robustness is therefore not treated as a general property of the TALEA DSS. In this thesis, robustness is evaluated as the degree to which a model preserves spatial selections and output quantities when controlled perturbations are applied to the input data, using the unperturbed baseline as the reference solution.

The evaluation follows a baseline-versus-scenario design. For each model, the no-noise output `no_noise.geojson` is treated as the reference solution. Perturbed solutions are then generated by modifying one input source at a time, rebuilding or reparsing the corresponding model input, solving the same optimization model, and comparing the resulting selected grid cells with the baseline output. This design makes the interpretation local: a model appears more stable under a given scenario when its perturbed solution remains close to the baseline according to the implemented spatial and ratio-based metrics.

This design directly addresses the research questions introduced in Section 1.5. Spatial overlap and centroid-buffer metrics address RQ1 by measuring how far selected Green Cell locations deviate from the no-noise baseline. Grouping results by model, source, perturbation type, and level addresses RQ2 by identifying model-source-perturbation combinations associated with spatial or quantitative deviations. Comparing spatial metrics with utility, yard-allocation, and street-allocation ratios addresses RQ3 by testing where spatial similarity and output-ratio metrics converge or diverge.

6.2 Perturbation Design

The perturbation strategy is based on controlled changes to selected data sources used by the optimization pipeline. The implementation distinguishes between *noise source*, meaning the data component being perturbed, and *noise type*, meaning the perturbation mechanism. All robustness experiments are conducted on the full-city Bologna configuration, which includes both central and peripheral areas of the study domain. Subsequent references to the experiments therefore refer to this configuration unless stated otherwise. The thesis uses the metric CSVs and associated plotting outputs generated for this configuration as the evidence base for the quantitative results.

The implemented perturbation types are:

- additive Gaussian perturbation, where a zero-mean normal error is added to numerical values;
- multiplicative Gaussian perturbation, where values are multiplied by a normally distributed factor centered on one;
- mislabelling, where attributes or values are swapped between records;
- missingness, where values are removed and then replaced using local or grouped imputation;

- aggregation changes, where UHEI or NDVI are aggregated by mean or median instead of the weighted baseline aggregation;
- geometric perturbation, where polygon geometries are scaled around their centroids before overlap computations.

Levels 1 to 5 generally represent increasing perturbation intensity within a given perturbation function. However, noise levels are ordinal scenario intensities within each perturbation family, not physically equivalent magnitudes across all sources. For example, a high mislabelling level for land use is not directly comparable to a high Gaussian perturbation level for NDVI or UHEI. Therefore, cross-source comparisons should be interpreted as relative empirical stress tests within the implemented perturbation design, not as physically commensurate uncertainty magnitudes. The exact numerical meaning of a level depends on the source and the function, and is documented in Chapter 7.

Although the perturbations are stochastic in mechanism, the inspected functions use fixed random seeds by default. Consequently, each perturbation level corresponds to a reproducible scenario, not to an independent Monte Carlo sample. This is important for interpretation: the thesis evaluates empirical robustness evidence under the tested perturbation scenarios, but it does not estimate full probability distributions, confidence intervals, or statistical uncertainty bounds.

6.3 Pipeline Logic

The robustness pipeline has four conceptual stages:

1. produce or select the no-noise baseline output for a given model and study size;
2. perturb one selected source while leaving the remaining scenario definition unchanged;

3. re-run the preprocessing, instance generation, and solver steps required by that source;

4. compare the perturbed GeoJSON output with the corresponding baseline GeoJSON output.

The analysis in this thesis covers the four implemented model families: `std`, `diff`, `inverse_uhei`, and `ndvi`.

Unperturbed inputs → baseline processing/parsing → baseline output

↓

Selected source and noise type → perturbed processing/parsing → perturbed output

↓

Baseline-versus-perturbed comparison using spatial and ratio metrics

↓

CSV summaries and plots for model, source, type, and level comparisons

Figure 6.1: Schema of the robustness comparison design. The same model is compared against its own no-noise baseline under each recognized perturbation scenario.

6.4 Comparison Design

The comparison design is deliberately paired. The baseline solution and the perturbed solution are compared within the same model family, rather than comparing a perturbed solution from one model with the baseline of another. This avoids mixing model-formulation differences with perturbation effects. The design supports several levels of interpretation:

- *model-by-model comparison*, where each model is evaluated against its own baseline, supporting RQ2;

- *scenario-by-scenario comparison*, where the source and perturbation type identify the tested uncertainty mechanism, supporting RQ2;

- *level-by-level comparison*, where increasing levels indicate increasing intensity within the same perturbation function, supporting RQ2 within a perturbation family;
- *metric-by-metric comparison*, where spatial similarity and output-ratio metrics are interpreted together, supporting RQ1 and RQ3.

Spatial metrics and output-ratio metrics are complementary. Spatial metrics assess whether the recommended cells remain in similar places. Ratio metrics assess whether the average utility, yard allocation, or street allocation remains similar for the selected cells. A model can therefore look stable in one metric and sensitive in another. Table 6.1 summarizes the interpretation used in the results chapter.

Table 6.1: Interpretation of the robustness metrics used in the baseline-versus-perturbed comparison.

Metric	Compared objects	Expected robust behaviour	Low or extreme values	Main limitation
Jaccard similarity	Unioned selected-cell polygons	Value close to 1, indicating a similar selected footprint	Values near 0 indicate little area overlap with the baseline	Strict for nearby but non-overlapping cells
Centroid-buffer IoU	Selected-cell centroids within the implemented distance tolerance	Value close to 1, indicating that perturbed selections can be matched near baseline selections	Low values indicate shifted locations, different selected-cell counts, or both	Depends on the 300 m tolerance and ignores exact polygon area

Metric	Compared objects	Expected robust behaviour	Low or extreme values	Main limitation
Utility ratio	Mean `utility` in perturbed and baseline outputs	Ratio close to 1	Very large, negative, or near-zero values indicate scale/sign sensitivity or denominator problems	Unstable when means are small, zero, non-finite, or sign-changing
Yard ratio	Mean `yard_cells` in perturbed and baseline outputs	Ratio close to 1	Values far from 1 indicate changed yard-like allocation magnitudes	Does not identify where allocation changed spatially
Street ratio	Mean `street_cells` in perturbed and baseline outputs	Ratio close to 1	Values far from 1 indicate changed street allocation magnitudes	Does not identify where allocation changed spatially

6.5 Spatial Robustness Metrics

The first group of metrics compares the spatial placement of Green Cell recommendations. Let B be the set of selected geometries in the baseline output and P the set of selected geometries in a perturbed output.

6.5.1 Jaccard Similarity

The implemented Jaccard metric unions the baseline geometries and the perturbed geometries, then compares their areas:

$$J(B,P) = \frac{\text{area}\,(G_B \cap G_P)}{\text{area}\,(G_B \cup G_P)}, \tag{6.1}$$

where G_B is the geometric union of all baseline selected cells and G_P is the geometric union of all selected cells in the perturbed solution. If both unions are empty, the implementation returns 1. If only one is empty, it returns 0.

The score ranges from 0 to 1 when valid geometries are available. A value close to 1 means that the perturbed solution occupies almost the same selected spatial footprint as the baseline. A value close to 0 indicates low overlap and therefore high spatial sensitivity. This metric is strict: nearby but non-overlapping cells are treated as different.

6.5.2 Centroid-Buffer IoU

The second spatial metric is named `iou_buffer` in the implementation. It is not a polygon area IoU. Instead, it compares selected-cell centroids using a distance tolerance. The implementation uses $d = 300$ metres by default.

Let C_B and C_P be the centroid sets of the baseline and perturbed selected cells. Candidate pairs are created when a baseline centroid and a perturbed centroid are at most d metres apart. A maximum bipartite matching is then computed so that each centroid can be matched at most once. If the matching routine is unavailable or fails, the implementation falls back to a greedy distance-ordered matching. If m_d is the number of matched pairs, the score is:

$$I_d(B,P) = \frac{m_d}{|C_B| + |C_P| - m_d}. \tag{6.2}$$

As in the Jaccard implementation, the score is 1 when both outputs are empty and 0 when only one is empty. Values close to 1 mean that most selected cells in the perturbed solution can be paired with baseline selections within the tolerance distance. Values much lower than 1 indicate that selected locations have moved beyond the buffer tolerance or that the selected-cell counts differ substantially. This metric is less strict than Jaccard, but it ignores the exact polygon overlap and depends on the chosen 300 m tolerance.

6.6 Ratio-Based Robustness Metrics

The second group of metrics compares average output attributes. For a column c, let $\bar{c}_B$ be its mean value over the baseline selected rows and $\bar{c}_P$ its mean value over the perturbed selected rows. The implementation writes two ratios:

$$R_{P/B}(c) = \frac{\bar{c}_P}{\bar{c}_B}, \qquad R_{B/P}(c) = \frac{\bar{c}_B}{\bar{c}_P}. \tag{6.3}$$

The evaluated columns are:

- `utility`, saved in `utility_diff.csv`;
- `yard_cells`, saved in `yard_util.csv`;
- `street_cells`, saved in `street_util.csv`.

A ratio of 1 means that the perturbed solution has the same mean output value as the baseline for that column. Values above 1 in $R_{P/B}$ mean that the perturbed mean is larger than the baseline mean, while values below 1 mean that it is smaller. These metrics do not have a fixed upper bound. They should be interpreted cautiously when the baseline or perturbed mean is close to zero, and especially when a model can produce negative utility values. This limitation is visible for the `ndvi` utility ratio under strong NDVI perturbations.

Spatial and ratio-based metrics answer different questions. A solution may select similar locations while changing the amount of yard or street allocation, or it may move to different locations while preserving a similar average utility. For this reason, the results chapter reports both metric families instead of reducing robustness to a single number.

6.7 Normalized Cross-Model Robustness Index

The comparison workflow also derives a normalized robustness percentage so that score metrics and ratio metrics can be summarized together across models. Let m be one of the implemented models and let s be a score metric, such as Jaccard similarity or centroid-buffer IoU. For these metrics, the normalized robustness value is

$$R_m(s) = \mathrm{clip}(s, 0, 1), \qquad R_{m,\%}(s) = 100R_m(s). \tag{6.4}$$

Thus a score of 1 corresponds to 100% robustness for that comparison, while lower scores indicate spatial deviation from the baseline.

For ratio metrics, let q denote the mean perturbed value divided by the mean baseline value. The comparison workflow treats positive increases and decreases away from the baseline symmetrically:

$$R_m(q) = \mathrm{clip}\left(\min\left(q, \frac{1}{q}\right), 0, 1\right), \qquad q > 0 \text{ and finite.} \tag{6.5}$$

The percentage form is again $R_{m,\%} = 100R_m$. For example, $q = 0.8$ and $q = 1.25$ both imply $R_m = 0.8$, because both represent a 20% departure from equality in opposite directions. Non-finite, zero, or negative ratios cannot be normalized by this rule and are excluded from the normalized comparison table. The corresponding drop from baseline similarity is

$$\mathrm{drop}_{\%} = 100 - R_{m,\%}. \tag{6.6}$$

The normalized comparison is a summary device, not a replacement for the metric-specific analysis. It allows the results chapter to compare the average and lower-tail behaviour of models across several metrics, but it hides the direction of ratio changes and inherits the limitations of the underlying metric files.

For a group of valid normalized comparisons, the workflow reports the mean robustness, median robustness, minimum robustness, 10th percentile robustness, maximum drop, and standard deviation. These summaries are computed for groups such as model, model–metric, model–noise type, and model–noise source. At the series level, where a series is defined by model, metric, noise type, and noise source, the workflow also records the available noise levels, the level-1 and level-5 values when present, and the range of robustness values.

For level-based perturbations, the degradation trend is summarized by fitting a first-degree polynomial to robustness percentage as a function of noise level:

$$R_{\%}(l) \approx \alpha + \beta l. \tag{6.7}$$

The slope β, reported in percentage points per noise level, is computed only when at least two valid levels are available. Negative slopes indicate that the normalized robustness tends to decrease as the perturbation level increases. Aggregation-based perturbations are handled separately because they compare alternative preprocessing choices and do not form an ordered level sequence.

Chapter 7

Implementation

7.1 Scope of the Implementation Analysis

This chapter documents the parts of the Python project that implement the robustness evaluation. It focuses on the functions that create perturbed data, pass those data into the optimization pipeline, and compute the robustness metrics from the generated GeoJSON outputs. The objective is not to describe every code line, but to explain how the implementation operationalizes the methodology in Chapter 6.

Three modules are responsible for creating perturbation scenarios:

- `process_data.py`, which injects noise into raw environmental, land-use, or population files;
- `grid_creation.py`, which builds the grid and applies perturbations that affect grid-level overlaps, counts, geometries, or aggregation methods;
- `data_parser.py`, which converts the processed grid into MiniZinc data and applies the `full_space` perturbation directly before instance generation.

The evaluation module then compares perturbed result files with the no-noise baseline and writes metric CSV files and PNG plots. The experimental result set used in this thesis consists of the evaluation outputs and corresponding plotting outputs generated for the study configuration defined in Chapter 6.

7.2 Raw-Data Perturbations

Raw-data perturbations are implemented in `process_data.py`. The script creates a scenario-specific raw-data copy and modifies the selected raw file in place. For raw-source scenarios, perturbed raw files are consumed downstream by the preprocessing and grid-creation stages. Grid-level and parser-level perturbations are instead applied directly within `grid_creation.py` and `data_parser.py`. The thesis therefore does not treat raw-data copies as the mechanism for every perturbation source.

All stochastic raw-data perturbations use a default seed of 42. Therefore, they are stochastic mechanisms with reproducible scenario outputs.

Table 7.1: Raw-data perturbation functions used to generate robustness scenarios.

Function	Source, type, and levels	Methodological role
`apply_noise_to_output`	Dispatch function for `uhei`, `ndvi`, `land_use`, and `population_density`.	Selects the appropriate perturbation function after the raw-data folder has been copied. Its output is the modified raw scenario folder used by the next preprocessing stage for raw-source scenarios.

Function	Source, type, and levels	Methodological role
`apply_noise_uhei`	Source `uhei`; types `gauss` and `multiplicative`; levels 1–5. Gaussian standard deviations are $0.1, 0.3, 0.5, 0.6, 0.7$. Multiplicative standard deviations are $0.1, 0.2, 0.3, 0.4, 0.5$.	Modifies the `value` property in `indexes/urban_heat_exposure_index.geojson`. Additive noise is clipped at zero; multiplicative noise uses factors centered on one and is also clipped at zero. The perturbed UHEI layer later affects weighted UHEI, inverse UHEI, and UHEI-sensitive models. The effect of these perturbations on macro-level optimization inputs depends on the preprocessing stage that produces the macro-level indicators.
`apply_noise_ndvi`	Source `ndvi`; types `gauss` and `multiplicative`; levels 1–5. Gaussian standard deviations are $0.1, 0.2, 0.3, 0.4, 0.5$. Multiplicative standard deviations are $0.05, 0.1, 0.2, 0.3, 0.4$.	Modifies the `value` property in `indexes/ndvi_mean.geojson`. Values are clipped to the NDVI interval $[-1, 1]$. The perturbed layer changes the weighted NDVI values used especially by the `ndvi` model.

Function	Source, type, and levels	Methodological role
apply_noise_ land_use	Source land_use; implemented as mislabelling; levels 1–5 with fractions $0.10, 0.15, 0.20, 0.30, 0.50$.	Swaps land-use classification fields between neighbouring features when both features are below the hectare threshold and have different descriptions. This perturbs the land-use layer before green, street, and free-space derivations are recomputed.
apply_noise_ population	Source population_density; types gauss, multiplicative, mislabelling, and missingness; levels 1–5.	Modifies the Residenti column in the population CSV. Gaussian noise uses a standard deviation proportional to each value; multiplicative noise uses factors centered on one; mislabelling swaps values within Quartiere and Zona; missingness removes values and imputes them using group, district, and global medians. The output affects density inputs. The repository indicates that these variables are read as part of the broader input structure, but the exact propagation path to any macro-level indicators should be treated cautiously unless the macro-generation stage is inspected directly.

7.3 Grid-Level Perturbations

Some perturbations are applied after raw layers have been transformed into grid-level indicators. These functions are implemented in `grid_creation.py`. Their inputs are QGIS layers, GeoJSON files, or GeoDataFrames created during grid construction; their outputs are updated processed layers and ultimately the processed grid passed to the parser.

Table 7.2: Grid-level perturbation functions used to generate robustness scenarios.

Function	Source, type, and levels	Methodological role
`apply_noise_trees`	Source `trees`; type `missingness`; levels 1–5 with fractions $0.10, 0.20, 0.30, 0.40, 0.50$.	Randomly masks a fraction of `tree_number` values in `trees_outside_green.geojson`, then imputes each masked value with the median of neighbouring grid cells in a 5×5 window. This perturbs the tree-count penalty used by several utility formulations.
`apply_num_areas_gauss`	Source `num_areas`; type `gauss`; levels 1–4 with standard deviations $0.5, 1.0, 1.5, 2.5$.	Adds Gaussian noise to `free_space_number`, clips negative values to zero, and converts the result to integers. The current implementation and result files do not include a level-5 Gaussian `num_areas` scenario.

Function	Source, type, and levels	Methodological role
`apply_num_areas_missingness`	Source `num_areas`; type `missingness`; levels 1–5 with fractions $0.10, 0.20, 0.30, 0.40, 0.50$.	Selects grid cells, finds neighbours within a 300 m centroid buffer, and replaces the selected `free_space_number` values by the local median. This tests sensitivity to uncertainty in free-space fragmentation counts.
`inject_polygon_geometric_noise`	Sources `green_area`, `street_space`, and `ext_space`; type `geometric`; levels 1–5. Maximum coordinate-scale deviations are $0.0025, 0.0050, 0.0100, 0.0150, 0.0250$.	Reads polygon layers, transforms them to EPSG:25832, scales each polygon around its centroid by a random factor above or below one, repairs geometries where possible, and writes a noisy polygon layer. The resulting layer is then used in overlap computations. Seeds differ by source in the main workflow: 42 for green areas, 43 for streets, and 44 for external/free space.
`verde_gauss_noise` and `verde_multiplicative_noise`	Source `green_area`; types `gauss` and `multiplicative`; levels 1–5. Gaussian standard deviations are $0.5, 1.0, 1.5, 2.5, 4.0$; multiplicative standard deviations are $0.05, 0.10, 0.20, 0.30, 0.40$.	Modify the computed `verde_area` and recalculate `verde_pc`. These perturb the existing-green component after grid overlap calculation.

Function	Source, type, and levels	Methodological role
_apply_area_noise and wrappers	Sources ext_space and street_space; types gauss and multiplicative; levels 1–5 with the same area-noise parameters used for green area.	General helper for perturbing area and percentage fields. The wrappers ext_space_gauss_noise, ext_space_multiplicative_noise, street_space_gauss_noise, and street_space_multiplicative_noise apply it to free-space and street-space overlap fields.
compute_mean_uhei, compute_median_uhei, compute_mean_ndvi, compute_median_ndvi	Sources uhei and ndvi; type aggregation; methods mean and median.	Replace the baseline weighted aggregation with mean or median aggregation for the selected environmental index. Unlike the other perturbations, this is deterministic and has no random seed. It tests sensitivity to an alternative preprocessing choice rather than to random measurement noise.

The main workflow in grid_creation.py controls how these perturbations enter the pipeline. Geometric noise is applied before overlap computation because it changes the source polygons. Area and count perturbations are applied after overlap computation because they modify grid-level attributes. Aggregation scenarios replace only the selected environmental aggregation while leaving the other environmental index on the weighted baseline computation.

7.4 Instance Generation and Full-Space Perturbation

The parser module converts the processed grid into MiniZinc data instances. It also implements the `full_space` perturbation through `add_noise_full_space`. This source is handled differently from most others because `full_space` is computed inside the parser by `utils.full_space_per_area`. It is therefore perturbed directly before the MiniZinc arrays are written.

For `full_space`, additive Gaussian noise uses standard deviations $100, 200, 300, 400, 500$ for levels 1–5, while multiplicative noise uses standard deviations $0.05, 0.10, 0.20, 0.30, 0.40$. Values are clipped below at one, preserving strictly positive total-space inputs for the models. The parser then constructs arrays for `green_space`, `street_space`, `ext_space`, `num_areas`, `num_trees`, `macro_factors`, and model-specific environmental fields. These arrays are written to MiniZinc instance files.

7.5 Perturbation Formulas and Code Snippets

The following short snippets illustrate the perturbation logic used in the implementation. They are intentionally partial: the complete functions also handle file loading, validation, and output writing.

7.5.1 Additive and Multiplicative Numerical Noise

The raw environmental perturbations use the same basic pattern for additive and multiplicative Gaussian noise. UHEI values are clipped below at zero; NDVI values are clipped to $[-1, 1]$.

```
if noise_type == "gauss":
    noise = np.random.normal(0, sigma, len(values))
    new_vals = np.clip(values + noise, 0, None)
```

```
elif noise_type == "multiplicative":
    noise = np.random.normal(1, sigma, len(values))
    new_vals = np.clip(values * noise, 0, None)
```

Mathematically, for additive Gaussian noise the implementation follows

$$x_i' = \mathrm{clip}(x_i + \epsilon_i, L, U), \qquad \epsilon_i \sim \mathcal{N}(0, \sigma_l^2), \tag{7.1}$$

where σ_l is selected by the perturbation level. For multiplicative noise it follows

$$x_i' = \mathrm{clip}(x_i \eta_i, L, U), \qquad \eta_i \sim \mathcal{N}(1, \sigma_l^2). \tag{7.2}$$

The bounds (L, U) depend on the variable: NDVI uses $(-1, 1)$, while UHEI and area-like variables use a lower bound of zero or one depending on the function. All these stochastic perturbations use fixed seeds in the inspected implementation, so the scenario files are reproducible.

7.5.2 Missingness and Median Imputation

For population missingness, the implementation removes a fraction of `Residenti` values and fills them through increasingly broad median groups.

```
df.loc[missing_idx, "Residenti"] = np.nan
df["Residenti"] = df.groupby(["Quartiere", "Zona"])["Residenti"] \
    .transform(lambda s: s.fillna(s.median()))
df["Residenti"] = df.groupby("Quartiere")["Residenti"] \
    .transform(lambda s: s.fillna(s.median()))
df["Residenti"] = df["Residenti"].fillna(global_median)
```

If M_l is the set of rows selected as missing at level l, then for each $i \in M_l$ the intended imputation is

$$x_i' = \mathrm{median}\{x_j : j \in G(i),\ x_j \text{ observed}\}, \tag{7.3}$$

where $G(i)$ is first the same `Quartiere–Zona` group, then the same `Quartiere`, and finally the whole dataset if needed. The result is clipped at zero and rounded because `Residenti` is a count.

The tree-count missingness perturbation applies the same idea at grid level, but the group is spatial rather than administrative. For a masked grid cell i, the replacement is the median of observed `tree_number` values in a 5×5 grid-cell window:

$$t'_i = \mathrm{median}\{t_j : j \in N_{5\times5}(i),\ t_j \text{ observed}\}. \tag{7.4}$$

7.5.3 Land-Use Mislabelling

The land-use perturbation is not numerical noise. It swaps classification fields between selected neighbouring features when both features are below the hectare threshold and have different descriptions.

```
if pair_valid(i, j, hect_thresh):
    for field in fields_to_swap:
        features[i]["properties"][field], \
        features[j]["properties"][field] = \
        features[j]["properties"][field], \
        features[i]["properties"][field]
```

For a valid selected pair (i, j), this can be written as

$$(a'_i, a'_j) = (a_j, a_i), \tag{7.5}$$

where a_i and a_j denote the exchanged land-use attribute vector. The geometry is not moved by this function; the scenario tests the effect of classification uncertainty on later spatial indicators.

7.5.4 Geometric Polygon Perturbation

Geometric perturbation scales polygon features around their centroids after transforming the layer to a metric CRS. The function preserves attributes, repairs geometries where possible, and writes a noisy polygon layer used in later overlap computations.

```
metric_gdf = source_gdf.to_crs("EPSG:25832")
rng = np.random.default_rng(seed)
scale_factor = _sample_geometric_scale(rng, maximum_delta)
noisy_geometry = shapely_scale(
    clean_geometry,
    xfact=scale_factor,
    yfact=scale_factor,
    origin="centroid"
)
```

For each polygon geometry g_i, the transformation is

$$g_i' = S(g_i; c_i, q_i), \tag{7.6}$$

where S scales the polygon around centroid c_i. The scale factor is $q_i = 1 \pm u_i$, with u_i sampled between one half of the level-specific maximum deviation and the full maximum deviation. In the main workflow, the seeds differ by source: green areas, street space, and external/free space use separate fixed seeds.

7.5.5 Full-Space Perturbation

The `full_space` source is perturbed inside the parser rather than by copying raw inputs.

```
if noise_type == "gauss":
    noise = np.random.normal(0, sigma, size=full_space.shape)
```

```
    new_values = np.clip(full_space + noise, 1, None)
else:
    noise = np.random.normal(1, sigma, size=full_space.shape)
    new_values = np.clip(full_space * noise, 1, None)
```

The implemented formulas are

$$F_i' = \max(1, F_i + \epsilon_i) \tag{7.7}$$

for additive Gaussian noise and

$$F_i' = \max(1, F_i \eta_i) \tag{7.8}$$

for multiplicative noise. The lower bound of one prevents zero or negative total-space values from entering the MiniZinc instance.

7.6 Evaluation Script

The script evaluation.py is the source of truth for the robustness metrics used in this thesis. It expects the baseline file to be named no_noise.geojson. For each model in the experimental result set, the script loads the baseline GeoJSON and all recognized perturbed GeoJSON files, computes the requested metrics, writes one CSV file per metric, and generates plots from those CSV files.

build_columns extracts noise_type, noise_source, noise_level, and aggregation_method from result filenames. It supports names of the form <noise_type>_<noise_source>_lvl<1-5>.geojson and aggregation_<noise_source>_<method>.geojson. This parsing is important because sources such as full_space, num_areas, green_area, ext_space, street_space, and population_density contain underscores.

`iter_result_files` skips the baseline file and ignores malformed filenames with a warning instead of stopping the evaluation. `validate_paths` raises an error if the result directory or baseline is missing. `load_geodata` checks required columns, requires a coordinate reference system, reprojects when requested, removes empty geometries, and repairs invalid geometries with `buffer(0)` when necessary.

The metric functions then compute:

- `metric_jaccard`, which writes `jaccard.csv`;
- `iou_buffer`, which writes `iou_buffer.csv` using the default 300 m centroid-matching distance;
- `utility_difference`, which writes `utility_diff.csv`;
- `utility_yard_cells`, which writes `yard_util.csv`;
- `utility_street_cells`, which writes `street_util.csv`.

For `iou_buffer`, the implementation creates candidate centroid pairs within the 300 m distance tolerance and then computes a maximum bipartite matching. If the matching routine is not available or fails, the code uses a greedy distance-ordered fallback. The resulting score is therefore based on one-to-one centroid matches, not on overlapping polygon area.

For plotting, `create_requested_plots` reads the CSV files and generates combined model-level plots as well as source-specific plots. Aggregation scenarios are plotted separately as changes from the neutral baseline value, because they do not have ordered noise levels. The plots used in the results chapter should therefore be read as visual summaries of the metric CSV files rather than as independent calculations.

7.7 Cross-Model Comparison Outputs

After the per-model metric files are available, the evaluation runner performs a quantitative comparison across models. This comparison reads the five metric CSV files produced for each model:

- jaccard.csv;
- iou_buffer.csv;
- utility_diff.csv;
- yard_util.csv;
- street_util.csv.

For each row, the comparison uses the metadata fields noise_type, noise_source, noise_level, and aggregation_method. The first two metric files are treated as score metrics and are normalized by clipping the score to the interval $[0, 1]$. The three ratio files are treated as perturbed-over-baseline ratios and are normalized with the symmetric rule described in Section 6.7. This means that a ratio above one is not automatically considered better; departures above and below the baseline are both counted as loss of robustness.

The comparison writes a long-format robustness table containing one normalized row per valid model–metric–scenario observation. It then produces summary CSV files by model, by model and metric, by model and noise type, by model and noise source, and by model, noise type, and source. A separate series-level table groups each model–metric–type–source combination across available noise levels and records the minimum value, maximum drop, range, and degradation slope. Worst-series and best-series tables are derived from this series-level summary. Aggregation rows are written separately because they do not have ordered noise levels.

The same comparison pass also generates six plot types used in Chapter 8: a model-level bar chart, heatmaps by model and metric, by model and noise type, and by model and noise source, a worst-series chart, and a strongest-degradation-slope chart. These plots are visual summaries of the comparison CSV files. They are useful for ranking patterns, but the numerical claims in the results chapter are taken from the CSV summaries.

7.8 Handling of Missing or Inconsistent Cases

The implementation contains several safeguards that affect the interpretation of the results. Missing baseline files stop evaluation because every metric is defined relative to the baseline. Malformed scenario filenames are skipped with warnings, so the number of rows in a metric CSV reflects only recognized result files. Missing required columns in GeoJSON outputs raise errors for the affected metric. For ratio metrics, divisions by non-finite or near-zero denominators return `NaN`.

These behaviours are methodological choices. They make the evaluation robust to auxiliary files and minor geometry problems, but they also mean that the final result tables must be interpreted as the set of successfully recognized and evaluated scenarios, not as a theoretical exhaustive list of every possible perturbation. Because the stochastic perturbation functions use fixed seeds where inspected, the generated rows are reproducible scenario records rather than Monte Carlo samples for estimating confidence intervals or statistical uncertainty bounds.

Chapter 8

Results

8.1 Evaluated Result Set

The results in this chapter use the metric CSV files and corresponding plotting outputs generated for the experimental configuration defined in Chapter 6. Each metric CSV is produced by the evaluation script from recognized perturbed GeoJSON files and the corresponding `no_noise.geojson` baseline. The analysis covers the four implemented model families: `std`, `diff`, `inverse_uhei`, and `ndvi`.

The chapter is organized around the research questions in Section 1.5. Spatial-similarity results address RQ1 by measuring deviations from the no-noise baseline, model-source-perturbation comparisons address RQ2, and the comparison between spatial metrics and output-ratio metrics addresses RQ3.

Table 8.1 summarizes the spatial metric files. The row counts differ because not every model has the same set of valid perturbation scenarios. In particular, environmental sources such as `uhei` and `ndvi` are only meaningful for the model variants that use them.

With respect to RQ1, the means in Table 8.1 suggest that many perturbation scenarios produce outputs close to the baseline. However, the minima show that this average behaviour masks strong sensitivity for specific combinations of model, source, and noise type. In the plots, ordered perturbations

Model	Rows	Mean Jaccard	Minimum Jaccard	Minimum IoU buffer
std	94	0.8966	0.4452	0.4706
diff	108	0.8972	0.2120	0.3514
inverse_uhei	106	0.8915	0.4213	0.4493
ndvi	107	0.8388	0.0529	0.2484

Table 8.1: Summary of spatial robustness metrics in the experimental result set. IoU buffer refers to the centroid-matching metric implemented with a 300 m default tolerance.

are shown by level, while aggregation scenarios represent alternative preprocessing choices rather than increasing noise levels. The following sections therefore focus on the sources that produce the largest changes rather than treating the mean alone as a robustness conclusion. Some combined plots contain many scenario lines, so they should be interpreted together with the CSV summaries and the source-specific plots.

8.2 Spatial Similarity by Model

For the `std` model, the lowest Jaccard value is 0.4452 under level-5 land-use mislabelling. Other low values occur for multiplicative `ext_space` perturbation at level 5, with Jaccard 0.4792, and Gaussian `num_areas` perturbation at level 4, with Jaccard 0.5233. This indicates that the `std` model is not uniformly stable: it appears comparatively stable under several numerical perturbations, but it changes substantially when land-use labels or free-space structure are strongly perturbed.

For the `diff` model, the strongest spatial sensitivity is associated with multiplicative `ext_space` perturbation. The Jaccard score decreases from 0.7543 at level 1 to 0.2120 at level 5. The source-specific plot in Figure 8.3 makes this pattern clearer than the combined plot in Figure 8.2. The IoU-buffer metric is less strict but tells a similar story: multiplicative `ext_space` reaches 0.3889 at level 5, and population-density mislabelling reaches the lowest IoU-buffer score of 0.3514.

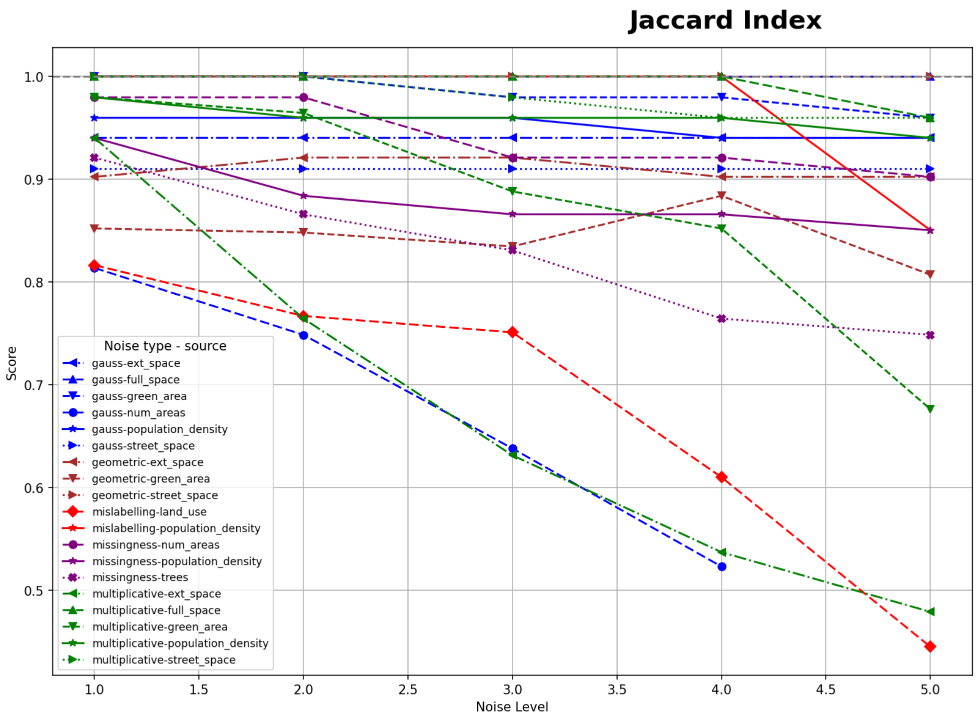

Figure 8.1: Jaccard similarity for the `std` model. The plot shows high similarity for many scenarios but visible drops for land-use mislabelling, multiplicative external-space perturbation, and `num_areas` perturbation.

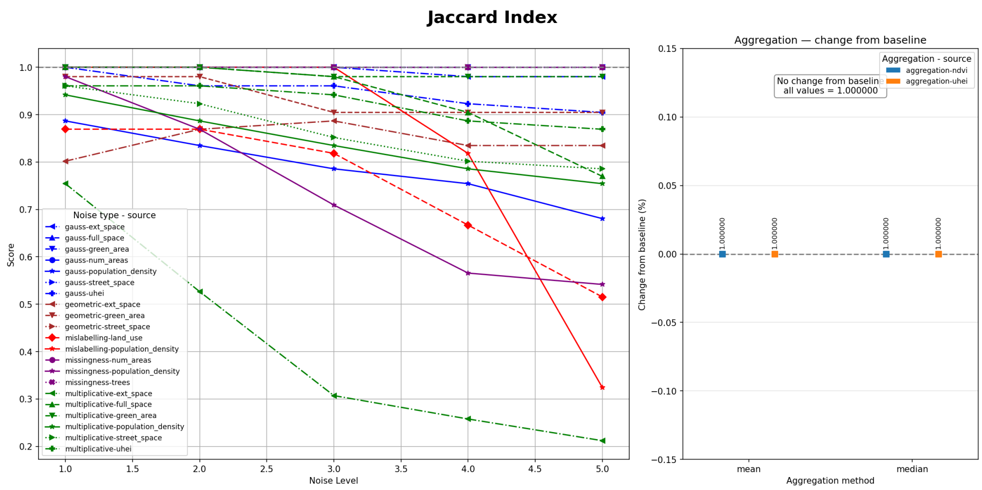

Figure 8.2: Jaccard similarity for the `diff` model. Aggregation scenarios for UHEI and NDVI remain at 1.0 in this metric, while several perturbations with ordered noise levels reduce spatial overlap.

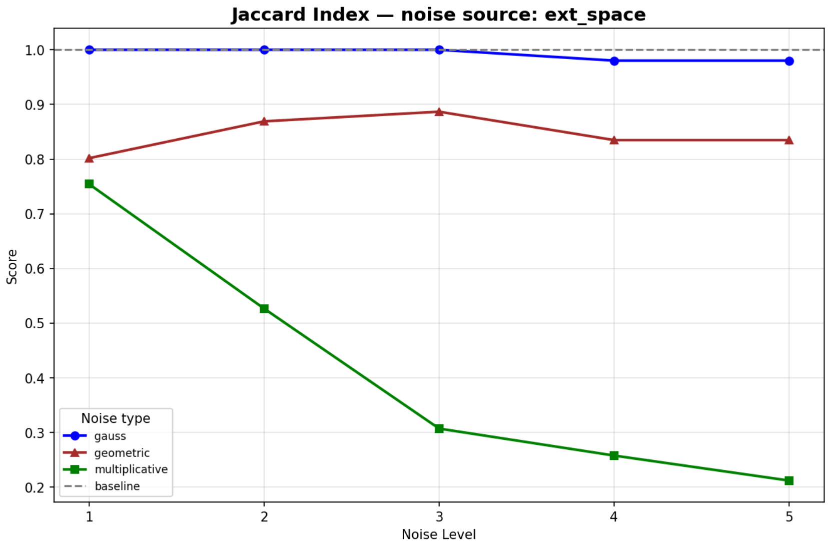

Figure 8.3: Source-specific Jaccard plot for `ext_space` perturbations in the `diff` model. Multiplicative perturbation produces the strongest decrease among the tested `ext_space` scenarios.

For the `inverse_uhei` model, the lowest spatial scores occur under land-use mislabelling and `full_space` perturbation. The minimum Jaccard is 0.4213 for level-5 land-use mislabelling, closely followed by 0.4252 for level-5 multiplicative `full_space`. The full-space result is consistent with the role of available total space in feasibility and utility scaling; land-use mislabelling likely acts through upstream spatial indicators.

The `ndvi` model is the most sensitive model in the spatial metrics. Its minimum Jaccard is 0.0529 under level-5 Gaussian NDVI perturbation, and the same source produces low scores already at level 3 and level 4. The IoU-buffer metric confirms that the perturbed selections are not only an artefact of exact polygon overlap: under Gaussian NDVI perturbation, the IoU-buffer scores fall to 0.2903 at level 3, 0.2484 at level 4, and 0.2579 at level 5.

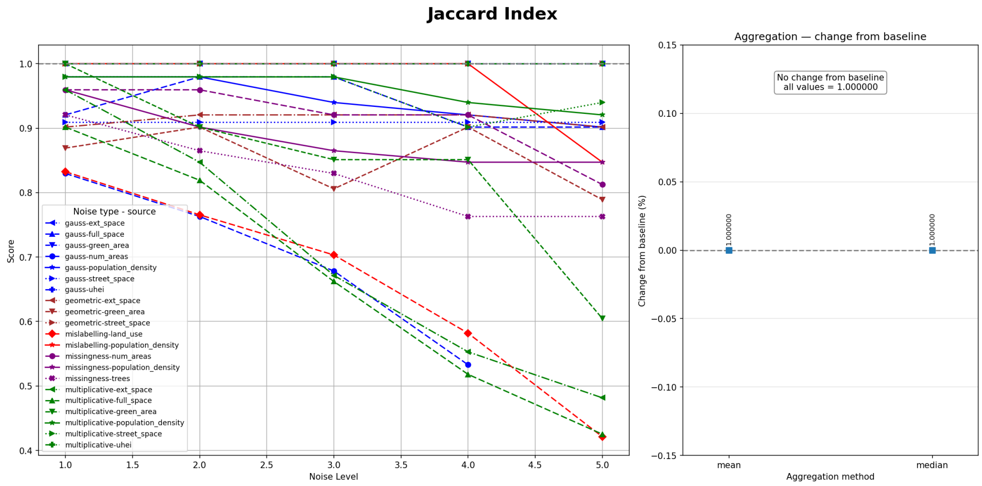

Figure 8.4: Jaccard similarity for the `inverse_uhei` model. The largest spatial changes appear for land-use mislabelling and full-space perturbations.

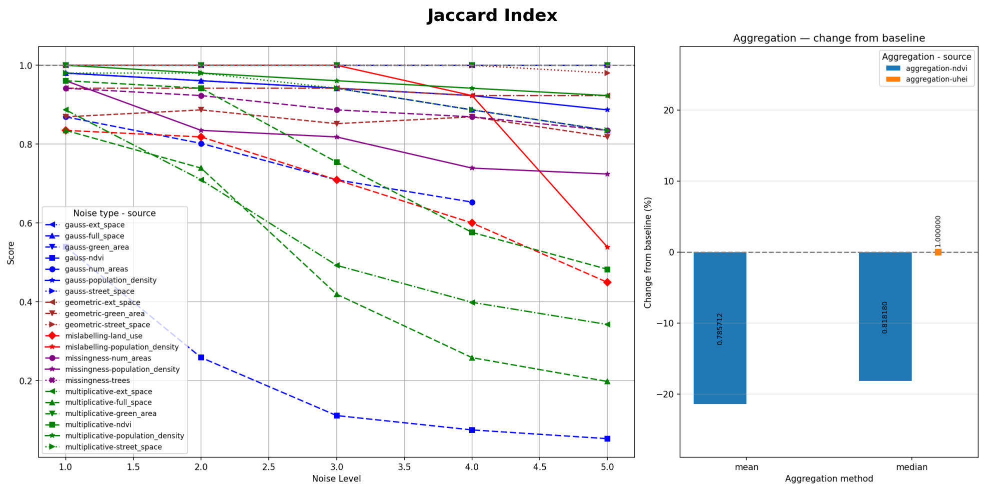

Figure 8.5: Jaccard similarity for the `ndvi` model. The strongest drop is associated with Gaussian perturbation of the NDVI source.

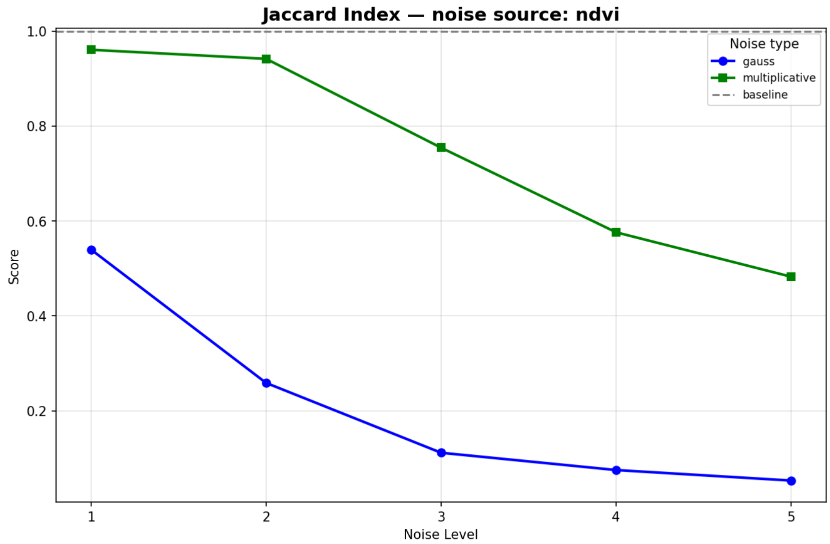

Figure 8.6: Source-specific Jaccard plot for NDVI perturbations in the `ndvi` model. Gaussian NDVI perturbation has a much stronger effect than multiplicative NDVI perturbation within the tested scenarios.

8.3 Aggregation Scenarios

Aggregation scenarios do not use ordered noise levels, so they are evaluated as alternative preprocessing choices. In the `diff` model, both mean and median aggregation for UHEI and NDVI give Jaccard 1.0. In the `inverse_uhei` model, UHEI mean and median aggregation also give Jaccard 1.0. In contrast, the `ndvi` model changes under NDVI aggregation: the Jaccard score is 0.7857 for mean aggregation and 0.8182 for median aggregation. The UHEI aggregation scenario present for `ndvi` remains at 1.0 in the inspected Jaccard file.

These results suggest that changing the aggregation method has clearer effects when the perturbed index is directly used by the model utility. This interpretation should remain cautious because only the implemented aggregation alternatives and available result files are considered.

8.4 Ratio-Based Metrics

With respect to RQ3, the output-ratio metrics add information about output magnitudes that is not captured by spatial overlap alone. Table 8.2 reports the range of `mean_noise_over_baseline` for each model and metric. A value of 1 is neutral. Values below 1 indicate a lower perturbed mean than the baseline mean, and values above 1 indicate a higher perturbed mean.

Model	Utility ratio range	Yard ratio range	Street ratio range
`std`	0.5213–1.1025	0.7478–1.1096	0.7902–1.2890
`diff`	1.0000–2.4720	0.9817–1.3968	0.8320–1.1833
`inverse_uhei`	0.8199–2.3009	0.8719–1.2825	0.7076–1.0797
`ndvi`	-808.2308–176.5561	0.6533–1.4854	0.7671–1.2849

Table 8.2: Range of the `mean_noise_over_baseline` ratio in the three ratio-based metric files for each model.

For `std`, multiplicative `ext_space` perturbation produces the largest utility-ratio decrease, reaching 0.5213 at level 5. The same source also affects allocation magnitudes: the yard ratio reaches 0.7478 and the street ratio reaches 1.2890 in the largest observed deviations. This indicates that perturbed solutions can reallocate the relative amount of yard and street intervention even when some spatial metrics remain moderate.

For `diff`, the utility-ratio range is high partly because the baseline mean utility is small, approximately 0.2236 in the CSV file. The largest utility ratios occur for geometric `ext_space` and missingness in `trees`, both above 2.4. Because the denominator is small, these values should be interpreted as signs of sensitivity in the implemented metric rather than as proportional planning benefits.

For `inverse_uhei`, the largest utility-ratio increases occur under UHEI perturbations: multiplicative UHEI reaches 2.3009 at level 5, while Gaussian UHEI reaches 1.8553 at level 5. This evidence is consistent with the model's dependence on inverse-UHEI scaling and percentiles derived from the processed data.

For `ndvi`, the utility ratio becomes difficult to interpret under Gaussian NDVI perturbation. The CSV records a utility ratio of -808.2308 at level 4 and 176.5561 at level 5 because the perturbed mean utility changes sign or scale relative to the baseline mean. Figure 8.7 therefore should not be read as a conventional robustness curve. It shows that the implemented utility-ratio metric is unstable for this scenario and should be interpreted together with the spatial metrics.

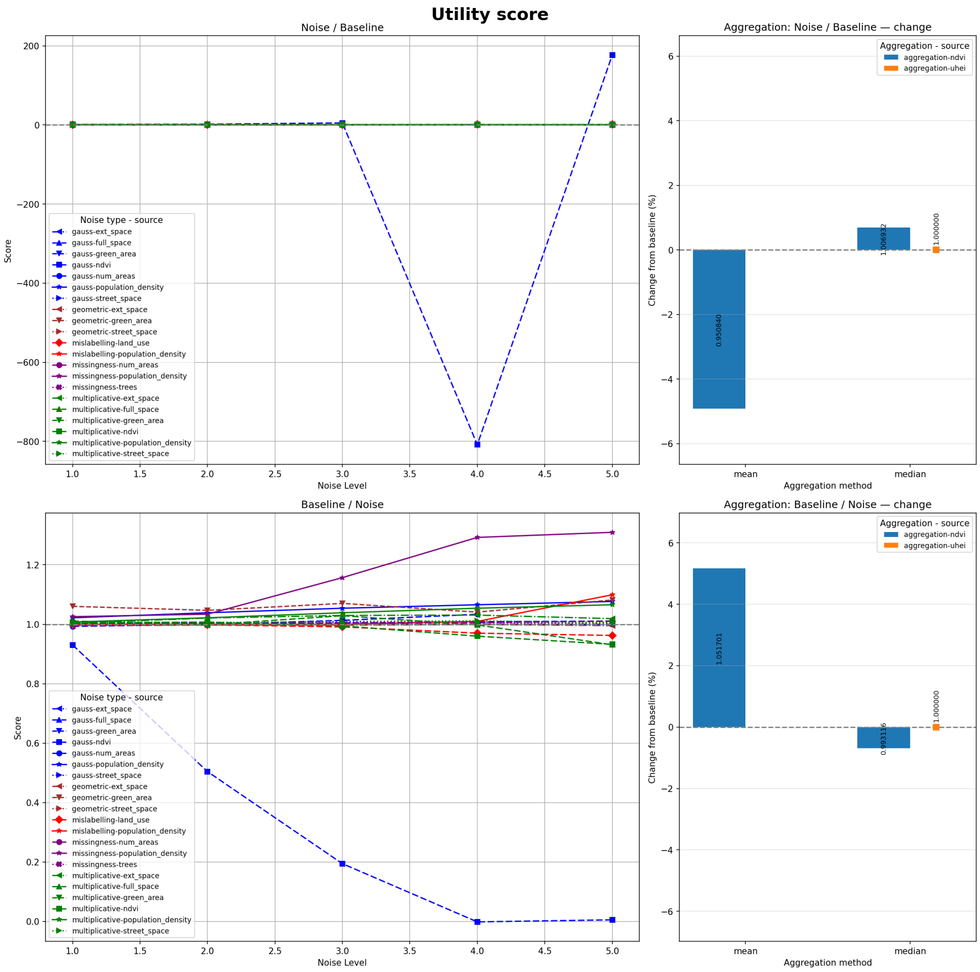

Figure 8.7: Utility-ratio plot for the `ndvi` model. The extreme Gaussian NDVI values show a limitation of ratio-based utility comparison when the perturbed mean utility changes sign or becomes very large relative to the baseline.

8.5 Cross-Model Quantitative Comparison

The comparison summaries normalize the five metric files to a 0–100 robustness scale, as defined in Section 6.7. Score metrics are converted directly from their similarity score, while ratio metrics are converted symmetrically so that proportional increases and decreases away from the baseline both reduce robustness. The values in this section summarize level-based perturbation scenarios; aggregation rows are retained separately because they do not have ordered noise levels. The comparison also addresses the lower-tail part of RQ2 by distinguishing average robustness from worst observed robustness.

Model	Points	Mean $R_{\%}$	Median $R_{\%}$	P10 $R_{\%}$	Worst $R_{\%}$	Max drop
`inverse_uhei`	520	94.0	98.1	83.5	42.1	57.9
`std`	470	93.9	97.9	81.8	44.5	55.5
`ndvi`	519	92.0	98.8	75.3	0.6	99.4
`diff`	520	84.1	98.0	42.3	21.2	78.8

Table 8.3: Normalized robustness summary by model for level-based perturbation scenarios. The table combines spatial scores and ratio metrics, so it should be interpreted together with the metric-specific results.

Table 8.3 and Figure 8.8 show that `inverse_uhei` has the highest mean normalized robustness, at 94.0%, followed very closely by `std` at 93.9%. The difference between these two averages is small and should not be overinterpreted. The `ndvi` model has a high median robustness of 98.8%, but its worst normalized value is only 0.6%, showing that high central tendency hides severe lower-tail sensitivity. The `diff` model has the lowest mean normalized robustness, 84.1%, even though its median is 98.0%; this reflects a smaller number of strongly damaging series rather than uniformly poor behaviour.

The model–metric heatmap in Figure 8.9 explains part of this ranking. The `diff` model retains high yard and street robustness, with mean values of 98.7% and 98.0%, but its mean normalized utility robustness is only 43.8%. The `ndvi` model has lower mean spatial robustness than the other models, with 83.8% for Jaccard and 86.5% for centroid-buffer IoU, while its ratio metrics

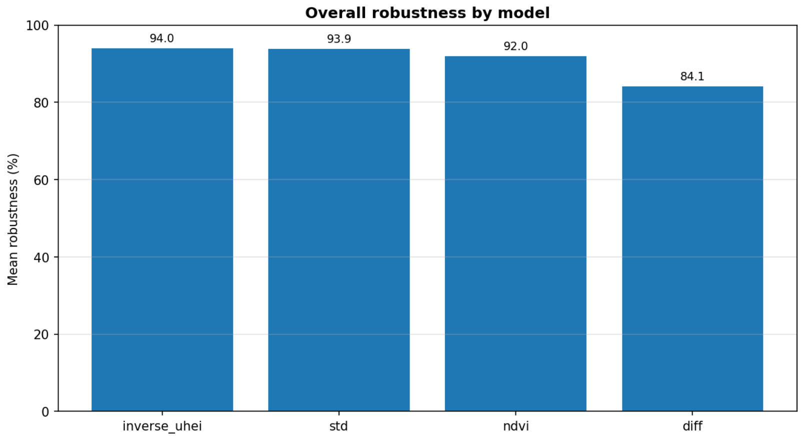

Figure 8.8: Mean normalized robustness by model. The plot summarizes all valid level-based metric rows and should be read as an aggregate comparison, not as a replacement for metric-specific evidence.

are high on average but include a severe utility outlier. This confirms that the normalized average combines different kinds of sensitivity and should not be read as a single planning-quality score.

The heatmap by noise type in Figure 8.10 indicates that the `diff` model has the lowest mean robustness under mislabelling and multiplicative perturbations, at 79.9% and 81.6% respectively. For `ndvi`, Gaussian perturbation has a mean of 91.0% but includes the worst normalized value in the comparison. Thus mean robustness by noise type is useful for screening patterns, but the worst-case and slope summaries are needed to identify fragile series.

For the source-level part of RQ2, the source heatmap in Figure 8.11 is more diagnostic. The lowest source-level mean is the `ndvi` model under NDVI perturbation, at 71.9%. Other low source-level means are `diff` under `ext_space` perturbation at 78.2%, `diff` under land-use mislabelling at 78.4%, and `diff` under population-density perturbation at 80.0%. These values support the interpretation that the most important sensitivities are localized in specific model–source combinations.

The worst-series summary, shown in Figure 8.12, identifies the most fragile model–metric–type–source combinations. The lowest series is `ndvi` utility robustness under Gaussian NDVI perturbation: across the valid positive-ratio

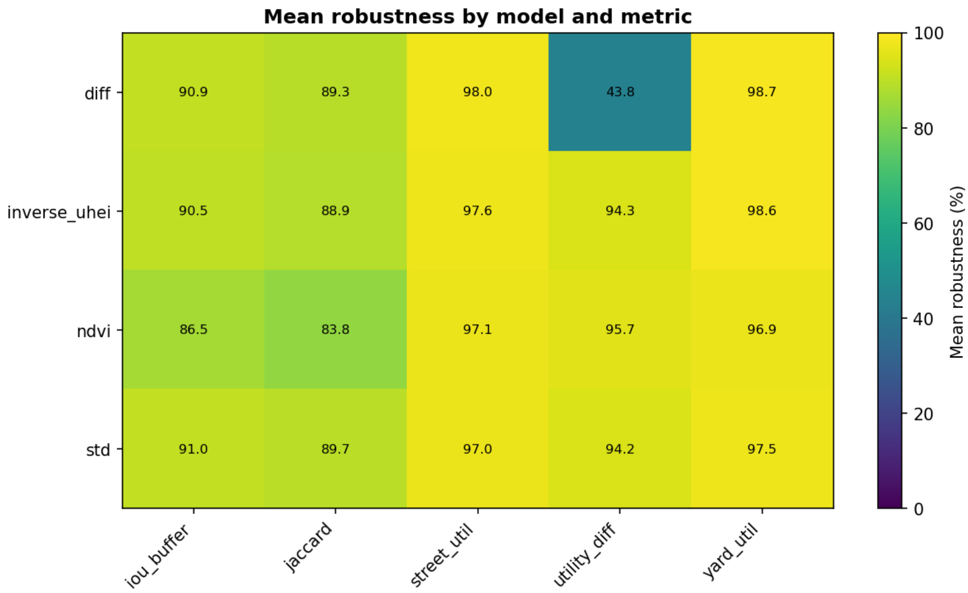

Figure 8.9: Mean normalized robustness by model and metric. Darker or lower cells identify metric-specific sensitivity that can be hidden in the overall model average.

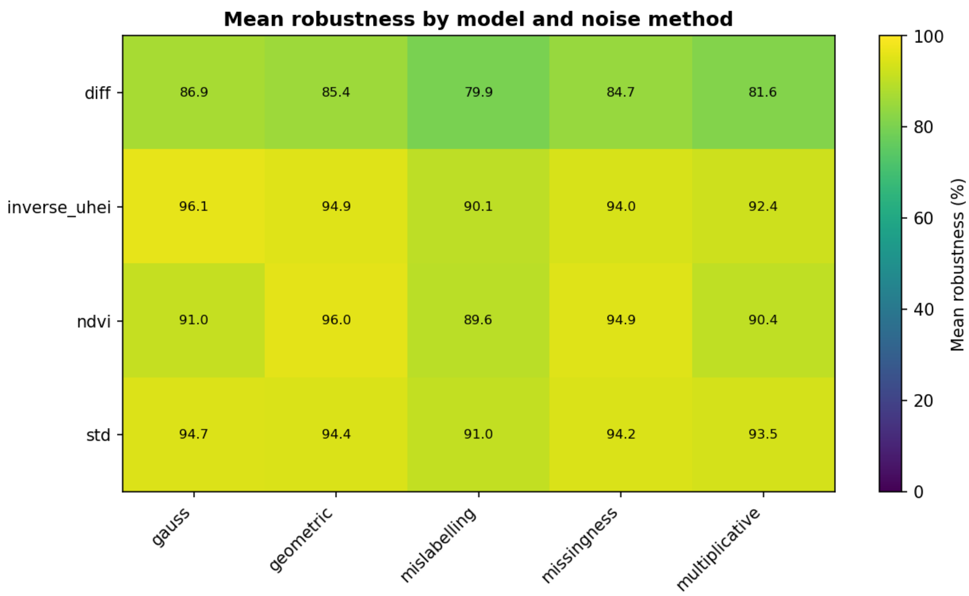

Figure 8.10: Mean normalized robustness by model and noise type. The plot highlights broad method-level patterns but does not show whether a low value is caused by one severe source or several moderate ones.

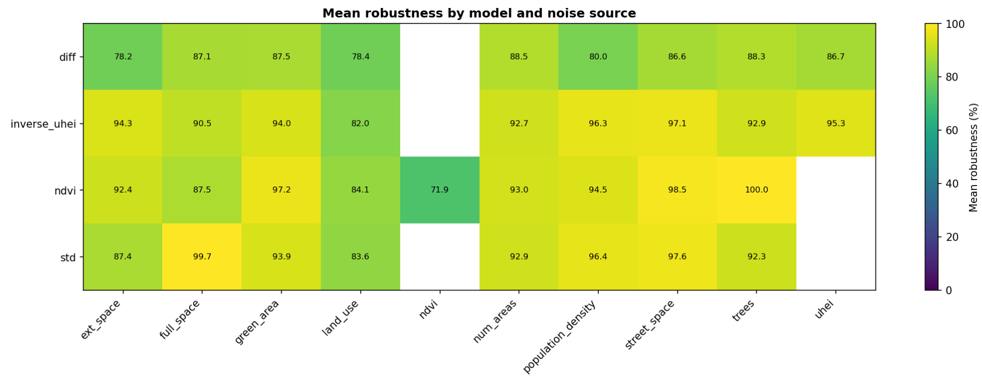

Figure 8.11: Mean normalized robustness by model and noise source. Source-level summaries help distinguish broad model sensitivity from sensitivity to specific input components.

levels listed in the comparison table, its mean normalized robustness is 40.9%, its minimum is 0.6%, and its maximum drop is 99.4 percentage points. The next worst spatial series is `ndvi` Jaccard under Gaussian NDVI perturbation, with mean robustness 20.8% and minimum 5.3%. The `diff` Jaccard series under multiplicative `ext_space` perturbation also has low robustness, with mean 41.2% and minimum 21.2%.

The degradation-slope summary in Figure 8.13 measures how quickly normalized robustness decreases as the perturbation level increases. The steepest negative slope is again `ndvi` utility robustness under Gaussian NDVI perturbation, at -22.23 percentage points per noise level. Other strong negative slopes include `ndvi` Jaccard under multiplicative `full_space` perturbation at -17.55, `diff` Jaccard under population-density mislabelling at -15.33, and `inverse_uhei` utility robustness under multiplicative UHEI perturbation at -14.94. These slopes indicate fitted degradation trends in the available scenario sequence, but they do not prove that the same trend would hold for untested perturbation levels or random seeds.

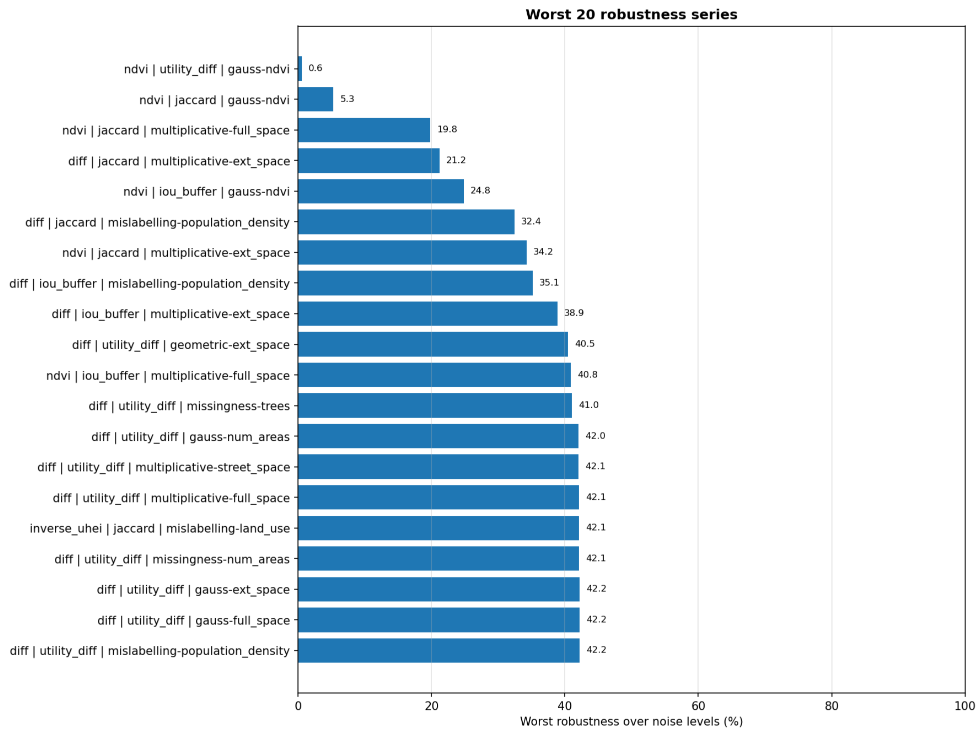

Figure 8.12: Worst 20 normalized robustness series, ordered by the worst value reached across available noise levels. This plot emphasizes lower-tail fragility rather than average model behaviour.

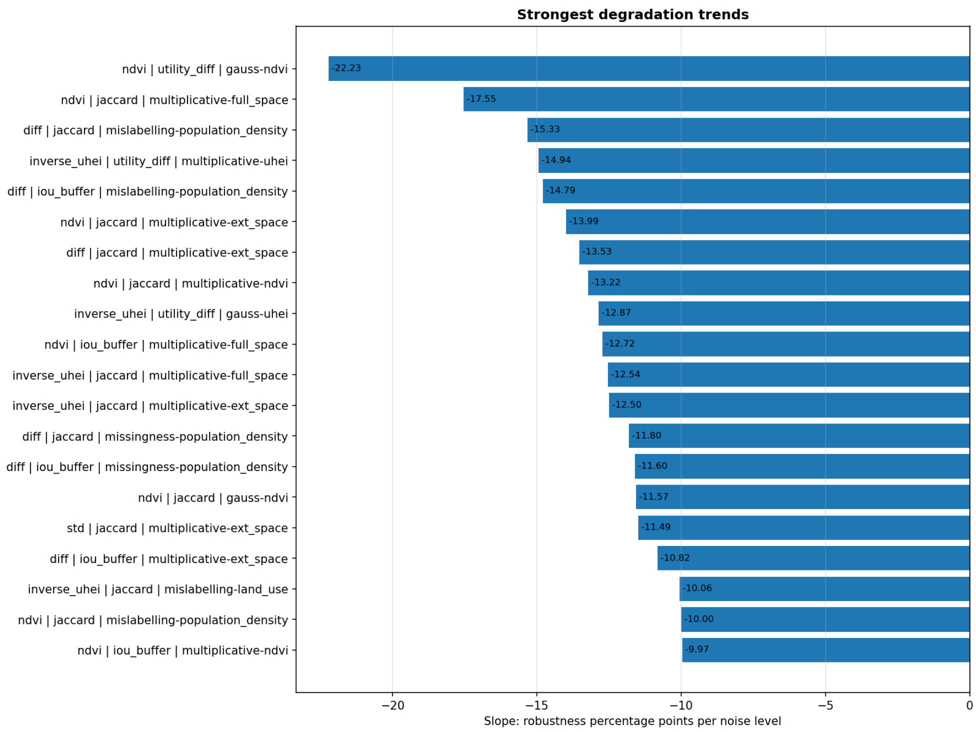

Figure 8.13: Series with the strongest negative degradation slopes. Slopes are fitted only for level-based perturbations with at least two valid levels and are reported in robustness percentage points per noise level.

The cross-model comparison therefore reinforces the metric-specific findings. Average robustness is high for several models, but severe fragility appears in selected model–source combinations. Ratio-based normalized robustness is especially useful for comparing deviations from the baseline in both directions, but it is still sensitive to invalid or sign-changing ratios; for example, the raw `ndvi` utility-ratio plot remains necessary to understand the extreme Gaussian NDVI case. The comparison plots are dense and should be interpreted together with the CSV summaries rather than as standalone evidence.

8.6 Summary of Main Observations

Within the tested perturbation scenarios, the result files answer the research questions in three main ways. For RQ1, many scenarios retain high spatial similarity to the no-noise baseline, but specific perturbation sources produce substantial drops. For RQ2, several of the strongest spatial or quantitative deviations occur where the perturbed variable enters the model directly or changes feasibility/allocation inputs: NDVI perturbations affect the `ndvi` model, UHEI perturbations affect the `inverse_uhei` utility ratio, and external/free-space perturbations affect models through feasibility and allocation terms. For RQ3, spatial and output-ratio metrics do not always tell the same story. A model may preserve similar locations while changing allocation magnitudes, or it may show large utility-ratio changes because of the scale of the baseline mean.

Chapter 9

Discussion

9.1 Interpreting Robustness in This Case Study

The results suggest that the implemented optimization models often preserve similar recommendations under many tested perturbations, but they are not uniformly stable across all tested sources and noise types. This distinction is important. The mean Jaccard and IoU-buffer values are relatively high for all models, but the minimum values show that some perturbations substantially alter the selected Green Cell locations. Therefore, the evidence supports a cautious statement: within the tested scenarios, robustness depends strongly on the perturbed input source, the model formulation, and the metric used for comparison.

The analysis should not be interpreted as proof that the DSS is reliable under all real-world uncertainty. The perturbations are controlled, synthetic, and generated with fixed seeds. They evaluate reproducible scenarios rather than a distribution of repeated random experiments. They also depend on the available result files and on the implementation choices in the evaluation script.

9.2 Meaning of the Cross-Model Comparison

The normalized comparison adds a compact view of the same evidence, but it should be interpreted as a diagnostic summary rather than as a final ranking of planning value. On average, `inverse_uhei` and `std` show the highest average stability under these tests, with mean normalized robustness close to 94%. The `ndvi` model has a similarly high median but a much weaker lower tail, while `diff` has the lowest mean because several utility and external-space-related series are strongly affected. This pattern reinforces the point that high mean robustness can hide localized fragility.

The comparison also clarifies why model-level averages are not enough. The `ndvi` model is not uniformly unstable: many of its scenarios remain close to the baseline, but Gaussian NDVI perturbation produces the most fragile series in both utility robustness and spatial overlap. The `diff` model likewise has high yard and street robustness on average, but its normalized utility robustness and multiplicative `ext_space` spatial series lower the aggregate score. These cases suggest that a planner should inspect source-specific robustness before relying on a baseline optimized map.

The ratio normalization is useful because it does not reward larger perturbed values simply because they exceed the baseline. A ratio of 1.25 and a ratio of 0.8 are treated as equally distant from baseline equality. However, this symmetry also removes direction and cannot handle zero, negative, or non-finite ratios. For that reason, the normalized comparison should be read alongside the raw ratio plots, especially for the `ndvi` utility metric where sign and scale changes make the raw ratio difficult to interpret.

The degradation slopes provide another useful warning signal. Strong negative slopes indicate fitted decreases in normalized robustness as the implemented perturbation level increases, as seen for Gaussian NDVI perturbation in the `ndvi` model and for several external-space or population-density series. This is consistent with scenario-dependent sensitivity, but it is not a statistical trend estimate over repeated random draws because each level is represented by the available fixed-seed scenario.

9.3 Spatial Stability of Selected Cells

The spatial-stability plane addresses RQ1. Jaccard similarity asks whether the selected cell footprint remains similar to the no-noise baseline footprint, while centroid-buffer IoU asks whether selected cells can still be matched near baseline selections within the implemented tolerance. High values in these metrics mean that the optimized map would look broadly similar to the baseline recommendation. Low values mean that the perturbation has changed the recommended places, either because cells move, because selected-cell counts differ, or because both occur.

This distinction matters for interpretation. A strict Jaccard drop can occur when nearby cells do not overlap exactly, whereas centroid-buffer IoU allows limited spatial displacement. When both metrics fall in the same scenario, as in the strongest NDVI and external-space cases, the evidence is more consistent with meaningful placement change than with a minor polygon-overlap artefact.

9.4 Quantitative Stability of Output Values

The quantitative-stability plane addresses the output-ratio side of RQ3. Utility, yard, and street ratios measure whether average output quantities remain close to the baseline, not whether the same cells are selected. They are therefore about magnitude stability rather than spatial coincidence. A scenario can preserve many baseline-like locations while changing the balance between yard and street allocations, or it can change utility ratios strongly because the baseline mean is small.

These ratio metrics are useful for identifying allocation or utility shifts, but they need cautious interpretation. Ratios become less stable when denominators are small, zero, non-finite, or when utility values can change sign. This is especially relevant for the `diff` and `ndvi` utilities, where sign and scale can make a ratio look extreme even when the planning meaning is not a simple proportional gain or loss.

9.5 Model-Source-Perturbation Sensitivity

The strongest effects relevant to RQ2 are generally consistent with the model formulations documented in Chapter 5. External-space perturbations act through yard-space bounds and allocation terms; total-space constraints also limit feasible allocations. This interpretation is consistent with why `ext_space` perturbations are important for the `diff` model and also affect allocation ratios in the other models, although the experiment does not isolate this mechanism independently.

The `ndvi` model appears especially sensitive to Gaussian NDVI perturbation. This interpretation is consistent with the formulation described in Chapter 5, because NDVI enters directly into the implemented utility expression. When the NDVI field itself is perturbed, the ranking of cells can plausibly change sharply, and the spatial metrics show that the selected locations can diverge strongly from the baseline. Multiplicative NDVI perturbation is less disruptive in the available plots, suggesting that the exact perturbation mechanism matters, not only the source.

The `inverse_uhei` model shows sensitivity to UHEI in the utility-ratio metric and to `full_space` and land-use mislabelling in the spatial metrics. This split is consistent with the implemented formulation: UHEI affects the inverse-UHEI scaling, while total space and land use affect feasibility and the construction of the spatial indicators. The result therefore illustrates why robustness cannot be assessed from a single metric alone, although it should not be read as an independently isolated causal mechanism.

Land-use mislabelling is also important. Because land-use data contribute to the derivation of green areas, free spaces, and other spatial layers, label swaps can propagate through several later computations. In the result files, level-5 land-use mislabelling is among the worst spatial cases for both `std` and `inverse_uhei`. This suggests that classification errors in upstream land-use data can matter even when the final optimization model does not explicitly contain a variable named `land_use`.

9.5.1 Model-Level Patterns

The `std` model appears relatively stable for many perturbations, but not for all. Its lowest spatial scores occur for strong land-use mislabelling, multiplicative `ext_space`, and Gaussian `num_areas`. This interpretation is consistent with a model whose utility depends on available intervention space, existing green space, tree counts, and free-space fragmentation. The model is not highly sensitive to every perturbation, but it appears more sensitive when the perturbation changes core spatial availability or denominator terms.

The `diff` model has a high mean spatial similarity but a very low worst-case Jaccard under multiplicative `ext_space`. The observed behaviour is consistent with the implemented formulation, where additive differences may react differently when available yard-like space changes. The results suggest sensitivity to external-space magnitude, especially under multiplicative perturbation, but this should be treated as a scenario-specific observation rather than a general statement about all difference formulations.

The `inverse_uhei` model shows two forms of sensitivity. Spatially, land-use mislabelling and `full_space` perturbation produce the strongest changes. In the ratio metrics, UHEI perturbation has a strong effect on mean utility. This distinction is consistent with the model structure: UHEI affects the scaling of the utility, while total space and land-use-derived layers affect feasibility and candidate-cell attractiveness.

The `ndvi` model is the least stable model in the tested spatial metrics. Gaussian NDVI perturbation produces extremely low Jaccard and IoU-buffer values. The model also exposes a limitation of the utility-ratio metric: when the perturbed utility mean changes sign or becomes much larger than the baseline, the ratio can become very large or negative. For this model, spatial metrics are more directly interpretable for judging placement stability than the raw utility ratio alone.

9.6 Relationship Between Spatial and Quantitative Metrics

RQ3 asks how spatial similarity metrics and quantitative output-ratio metrics converge or diverge in diagnosing robustness or fragility. The results show that they are complementary but not interchangeable. In some scenarios, both metric families indicate sensitivity; in others, one metric family changes more strongly than the other. This means that robustness should not be reported as a single undifferentiated score.

The most useful interpretation is therefore layered. Spatial metrics answer whether the optimized locations remain close to the baseline. Ratio metrics answer whether average utility or allocation magnitudes remain close to the baseline. Model-specific summaries then show which formulations and input sources are associated with the largest deviations within the tested scenarios.

9.7 Practical Implications for Decision Support

For planning support, the main practical implication is that robustness should be assessed before treating an optimized map as a stable recommendation. Within the tested perturbation scenarios, some model-source-perturbation combinations preserve similar spatial selections, while others change sharply. This suggests that an optimization-based DSS should report not only the preferred baseline solution but also which input sources are most capable of changing that solution.

The results also suggest where data validation would be most valuable. NDVI uncertainty matters strongly for the `ndvi` formulation, external/free-space perturbations matter for allocation-sensitive models, and land-use mislabelling can propagate through upstream spatial indicators. These observations do not prove that those datasets are inaccurate in the Bologna case study. They indicate that, for this implementation, those sources deserve careful checking before recommendations are used in planning discussions.

Finally, the divergence between spatial and ratio-based metrics has a communication implication. A decision-support interface should avoid presenting robustness as a single score. Spatial similarity, utility stability, and allocation stability answer different questions, and planners may care about them differently depending on whether the priority is preserving candidate locations, preserving expected benefits, or preserving the balance between street and yard-like interventions.

9.8 Methodological Limitations

Several limitations constrain the conclusions. First, the perturbation levels are implementation-defined and differ by source. Level 5 for land-use mislabelling is not directly comparable to level 5 for NDVI Gaussian noise or geometric polygon scaling. The levels represent increasing intensity within each function, not a universal uncertainty scale.

Second, the stochastic perturbations use fixed seeds. This supports reproducibility, but it does not quantify variability across repeated random draws. A stronger statistical robustness analysis would run multiple seeds for each perturbation level and report distributions of the metrics.

Third, the evaluation uses the available generated outputs and corresponding plots as the main evidence base. The result counts should therefore be interpreted as the successfully evaluated scenarios in that result set, not as all conceivable perturbations.

Fourth, the models are evaluated against their own no-noise baselines. This is appropriate for robustness, but it does not decide which baseline model is normatively preferable for urban planning. A model can be stable under perturbation while still encoding assumptions that planners may want to revise.

Finally, the robustness metrics compare implemented outputs, not real-world planning outcomes. A spatially stable recommendation is not automatically feasible, equitable, or desirable in practice. The analysis should be understood as a technical robustness assessment of an optimization-based DSS component within the Bologna Green Cells case study.

Chapter 10

Conclusion

This thesis assessed the robustness of an optimization-based decision support workflow for the Bologna Green Cells case study. The work did not redesign the TALEA DSS. Instead, it reconstructed the implemented data pipeline and optimization models, then evaluated how the generated recommendations change under controlled perturbations of selected input data.

The main methodological contribution is the code-grounded robustness framework. Perturbed scenarios are created by modifying environmental, demographic, land-use, geometric, and grid-level inputs; the optimization models are re-executed; and the resulting Green Cell placements are compared with the corresponding no-noise baseline. The evaluation uses Jaccard similarity, centroid-buffer IoU, and ratios of mean utility, yard allocation, and street allocation. For the cross-model comparison, these metrics are also transformed into a bounded robustness percentage that treats positive and negative ratio deviations from the baseline symmetrically.

For RQ1, the spatial metrics show that many tested perturbations leave selected locations relatively close to the no-noise baseline, but this average stability hides substantial spatial shifts in specific scenarios. The strongest spatial changes appear for combinations such as Gaussian NDVI perturbation in the `ndvi` model, multiplicative external/free-space perturbation in the `diff` model, and strong land-use mislabelling in the `std` and `inverse_uhei` models.

For RQ2, the evidence indicates that spatial and quantitative deviations are model-source-perturbation specific rather than absolute. In the normalized comparison, `inverse_uhei` and `std` show the highest average stability, both close to 94%, while `diff` has the lowest mean because selected utility and external-space series are fragile. The `ndvi` model has a high median but the most severe lower-tail failure under Gaussian NDVI perturbation.

For RQ3, the analysis shows that spatial similarity metrics and quantitative output-ratio metrics are complementary but not interchangeable. Spatial metrics reveal changes in selected locations, while output-ratio metrics reveal changes in average utility, yard allocation, and street allocation. These views can diverge, especially when baseline means are small or perturbed utilities change sign, as observed for the `ndvi` model under strong NDVI perturbation.

The conclusions should be read cautiously. The perturbations are synthetic, the stochastic scenarios use fixed seeds, and the evaluation relies on the available result set and corresponding plotting outputs. The results therefore do not prove that the DSS is robust in general. They show how the inspected implementation behaves under the tested perturbation scenarios and identify which sources deserve particular attention in future validation.

Future work should repeat stochastic scenarios with multiple seeds, add confidence intervals or distributional summaries, test additional perturbation families and spatial matching tolerances, verify uncertainty assumptions with official dataset metadata, and connect technical robustness with planning criteria such as feasibility, maintenance, equity, and stakeholder acceptability.

Bibliography

[1] T. Carlson and D. Ripley. On the relation between NDVI, fractional vegetation cover, and leaf area index. *Remote Sensing of Environment*, 62:241–252, 1997. DOI: 10.1016/S0034-4257(97)00104-1.

[2] B. Choudhury et al. Relations between evaporation coefficients and vegetation indices studied by model simulations. *Remote Sensing of Environment*, 50(1):1–17, 1994. DOI: 10.1016/0034-4257(94)90090-6.

[3] R. Church and C. ReVelle. The maximal covering location problem. *Papers of the Regional Science Association*, 32:101–118, 1974. DOI: 10.1007/BF01942293.

[4] D. D'Ippoliti, P. Michelozzi, C. Marino, F. de'Donato, B. Menne, K. Katsouyanni, U. Kirchmayer, A. Analitis, M. Medina-Ramón, A. Paldy, R. Atkinson, S. Kovats, L. Bisanti, A. Schneider, A. Lefranc, C. Iñiguez, and C. A. Perucci. The impact of heat waves on mortality in 9 european cities: results from the EuroHEAT project. *Environmental Health*, 9, 37, 2010. DOI: 10.1186/1476-069X-9-37.

[5] A. de la Iglesia Martínez and S. M. Labib. Demystifying normalized difference vegetation index (NDVI) for greenness exposure assessments and policy interventions in urban greening. *Environmental Research*, 220:115155, 2023. DOI: 10.1016/j.envres.2022.115155.

[6] C. Di Buò and S. Reale. Sustainable City via Trustworthy Digital Twin: A Use Case. Project report. Year and institution not visible in the inspected PDF.

[7] D. Dodman, B. Hayward, M. Pelling, V. Castan Broto, W. Chow, E. Chu, R. Dawson, L. Khirfan, T. McPhearson, A. Prakash, Y. Zheng, and G. Ziervogel. Cities, settlements and key infrastructure. In H.-O. Pörtner, D. C. Roberts, M. Tignor, E. S. Poloczanska, K. Mintenbeck, A. Alegría, M. Craig, S. Langsdorf, S. Löschke, V. Möller, A. Okem, and B. Rama, editors, *Climate Change 2022: Impacts, Adaptation and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 907–1040. Cambridge University Press, Cambridge, UK and New York, NY, USA, 2022. DOI: 10.1017/9781009325844.008.

[8] European Environment Agency. Urban Adaptation in Europe. EEA Report 12/2020, European Environment Agency, 2020. URL: https://www.eea.europa.eu/en/analysis/publications/urban-adaptation-in-europe (visited on 07/08/2026).

[9] N. Frantzeskaki, T. McPhearson, M. J. Collier, D. Kendal, H. Bulkeley, A. Dumitru, C. Walsh, K. Noble, E. van Wyk, C. Ordóñez, C. Oke, and L. Pintér. Nature-based solutions for urban climate change adaptation: linking science, policy, and practice communities for evidence-based decision-making. *BioScience*, 69(6):455–466, 2019. DOI: 10.1093/biosci/biz042.

[10] S. Geertman and J. Stillwell. Planning support systems: an inventory of current practice. *Computers, Environment and Urban Systems*, 28(4):291–310, 2004. DOI: 10.1016/S0198-9715(03)00024-3.

[11] R. Gillies and T. Carlson. Thermal remote sensing of surface soil water content with partial vegetation cover for incorporation into climate models. *Journal of Applied Meteorology*, 34, 1995. DOI: 10.1175/1520-0450(1995)034<0745:TRSOSS>2.0.CO;2.

[12] C. Heaviside, H. Macintyre, and S. Vardoulakis. The urban heat island: implications for health in a changing environment. *Current Environmental Health Reports*, 4:296–305, 2017. DOI: `10.1007/s40572-017-0150-3`.

[13] J. Malczewski. GIS-based multicriteria decision analysis: a survey of the literature. *International Journal of Geographical Information Science*, 20(7):703–726, 2006. DOI: `10.1080/13658810600661508`.

[14] K. Tzoulas, K. Korpela, S. Venn, V. Yli-Pelkonen, A. Kaźmierczak, J. Niemelä, and P. James. Promoting ecosystem and human health in urban areas using green infrastructure: a literature review. *Landscape and Urban Planning*, 81(3):167–178, 2007. DOI: `10.1016/j.landurbplan.2007.02.001`.

[15] H. P. Williams. *Model Building in Mathematical Programming*. John Wiley & Sons, Chichester, 5th edition, 2013. ISBN: 978-1-118-44333-0.

[16] J. R. Wolch, J. Byrne, and J. P. Newell. Urban green space, public health, and environmental justice: the challenge of making cities "just green enough". *Landscape and Urban Planning*, 125:234–244, 2014. DOI: `10.1016/j.landurbplan.2014.01.017`.