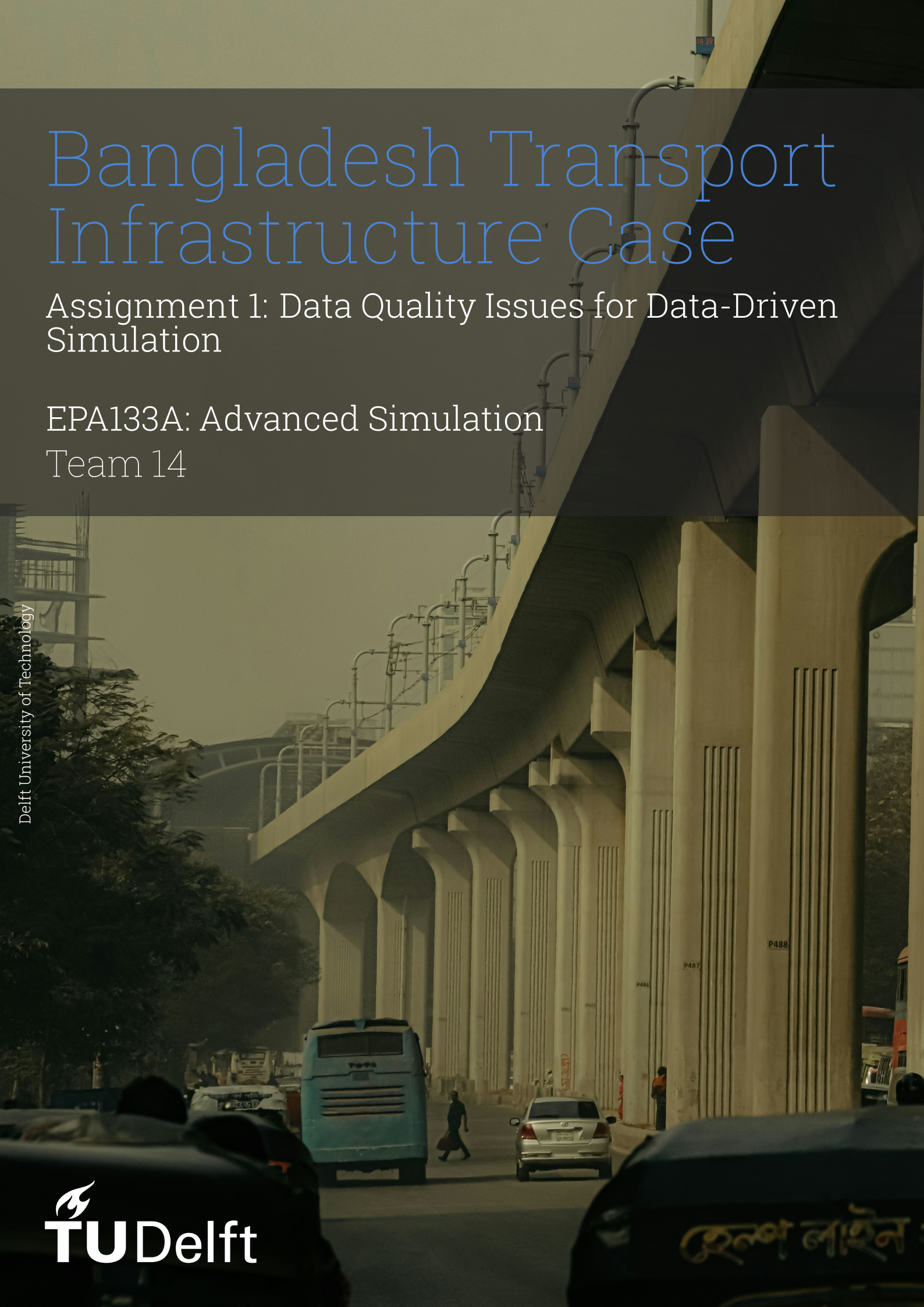
Bangladesh Transport Infrastructure Case

Assignment 1: Data Quality Issues for Data-Driven Simulation

EPA133A: Advanced Simulation

Team 14



Delft University of Technology

TU Delft

Bangladesh Transport Infrastructure Case

Assignment 1: Data Quality Issues for Data-Driven Simulation

by

Team 14

Bayu Jamalullael	6367984
Brenda Escobar	6512191
Brian Parsaoran	6147674
Taufik M. Yusup	6378056
Zhafran Sidik	5658411

20 February 2026

Faculty of Technology, Policy and Management (TPM)

TU Delft

Contents

1

Introduction

1.1. Context and Motivation

Bangladesh is one of the most disaster-prone countries in the world, regularly facing three types of disasters: earthquakes, river flooding, and cyclones. Its low-lying delta geography and dense population make the country and its infrastructure especially vulnerable to extreme events.

A prime example is such an event as Cyclone Sidr in 2007. It killed thousands of people and swept away more than a million houses. More than 5,000 bridges were washed away, thousands of kilometres of roads were impassable due to the floods, and 8 million people suffered from the cyclone.

The World Bank wants to know where infrastructure investments will have the greatest impact. This means investigating which roads, waterways, train tracks, and bridges are vulnerable to damage or destruction from natural hazards and would be critical from humanitarian and economic perspectives.

1.2. Datasets and Files Used

We use two datasets, one with road data and another with bridge data, both from the Ministry of Transport in Bangladesh. These files include geographic, physical, and identification information on each infrastructure element. However, these data files have many quality issues that need to be corrected before using the data.

1.3. Goal and Scope of This Assignment

The purpose of this report is to clean and process the data files for further analysis of the criticality and vulnerability of infrastructure elements, and to identify possible robust interventions. In particular, we identify and distinguish different types of data quality issues in the road and bridge data files and fix the most relevant issues to allow the simulation runs to use accurate data.

2

Data Quality Assessment

2.1. Data Quality Diagnosis

Prior to utilising the dataset for a simulation analysis aimed at identifying critical road and bridge infrastructure in Bangladesh, a comprehensive quality assessment is required. This process ensures that the data itself is usable so that the output of the simulation is reliable and robust enough to inform policy decisions.

The roads and bridges dataset was obtained from the Bangladeshi Ministry of Transport, and subsequently processed by researchers from TU Delft. The dataset's provenance can be assessed following the "Seven W's" framework outlined by Marsden and Pingry [3]. The *What* component fails to fully capture the data's reality due to errors. Meanwhile a lack of explicit description about the *When, Where, How, Who, Which and Why* hampers utility of the dataset without further processing. Furthermore, there is a misalignment in the *Why*: the dataset was presumably collected for the Ministry's internal administrative objectives rather than for a simulation study. Therefore, the dataset may contain extraneous information.

The issues identified are categorised according to the framework introduced by Huang and Verbraeck [2] in table 2.1 below. In the following subsection 2.2, a prioritisation of these issues is explained.

2.2. Prioritisation of Data Quality Issues

Given the time and resource constraints, not all errors are addressed in this exercise. We chose to focus on the errors that would enable us to run a reliable simulation to identify the critical roads and bridges infrastructure as outlined in the introduction. This means that a particular focus was taken on national highways N1 and N2. For the bridges dataset, a pragmatic approach is taken, as the combination of duplicated data and missing information makes it difficult to reliably assess the significance of the bridges for the research. As shown later, a few bridges remain in incorrect positions, and it is difficult to ascertain their actual positions without external data (cross-referencing with the roads dataset proved futile). Furthermore, these bridges lie on district roads (*Z roads*) that are less important for this research.

Table 2.1: Data quality assessment

Category	Criterion	Description
Syntactics	Accuracy	No data points were found to be in a different format than it should be, therefore no modifications were needed.
	Consistency	Some variables used were not consistently named. In the roads dataset, the LRPs were not consistently labelled as it was initially in a long format (e.g., ”lrp1”, ”lrp2”). This does not correspond to the ”LRPName” used in the bridges dataset.
Semantics	Accuracy	• In the bridges dataset, the chainage value was not always monotonous. • Both datasets contained points with ”out of bounds” coordinates, erroneously labelled (0,0) coordinates, and inverted latitudes and longitudes. • In the bridges dataset, the width and construction year were missing. Overall, these missing values correlated with an interpolation as the chosen method for estimating the location.
	Completeness	• In the roads dataset, there were data points with missing LRPs and coordinates. • In the bridges dataset, missing coordinates were the most relevant issue. Some bridges were in roads that could not be located in the roads dataset.
	Consistency	• In both datasets, there were identical row entries. • In the bridges dataset, some rows had identical coordinates/LRPs but different names.
Pragmatics	Completeness	Assumed to be complete for the research, as there is no need for any extra data.
	Timeliness	Could be fixed as the dataset appeared to be collected at different years. Some details have changed over time (e.g., bridge name, condition), leading to duplicates of the same bridge.
	Suitability	Assumed to fit the need for research.
	Precision	Coordinates are excellent because the number of decimal points increases spatial accuracy. However, issues with inverted, out-of-bounds, or missing coordinates can significantly alter the plotting of roads and bridges and the simulation results.
	Type-sufficiency	Assumed to fit.

3

Methodology for Data Cleaning

This section outlines the set of custom functions used to detect, flag and, where possible, repair issues identified in the dataset. These functions allowed the reproduction of the data cleaning processes in both data sets.

3.1. Data Restructuring and Preprocessing

The original road dataset (`_roads.tsv`) is stored in a wide format, where each road is represented as a single row containing repeating triples of *(LRP name, latitude, longitude)*. To enable systematic quality assessment and geometry-based cleaning, the dataset was first transformed into a tidy long format with one row per LRP point (`road`, `lrp`, `lat`, `lon`). This restructuring allows for (i) computing distances between consecutive reference points, (ii) detecting abnormal spatial jumps, and (iii) applying neighborhood-based repair strategies, while still processing each road independently.

As an initial consistency step, duplicate LRP entries within the same road were removed based on the (`road`, `lrp`) key to ensure uniqueness of reference points and to avoid ambiguous sequencing. Furthermore, roads containing three or fewer LRPs were excluded from the cleaning process, as they provide insufficient spatial context for distance-based anomaly detection and interpolation (the repair algorithm requires surrounding points to evaluate and correct a segment).

3.2. Chainage Monotony Issues

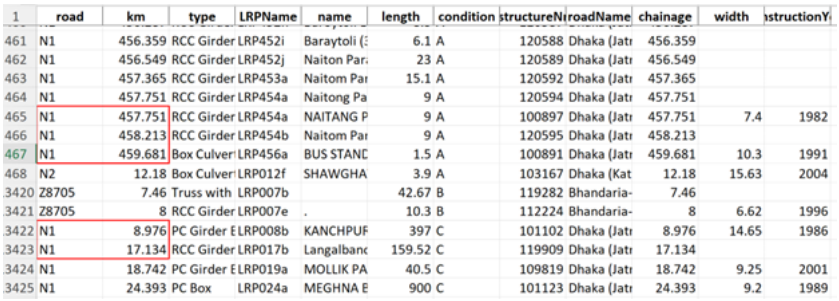
Chainage refers to the distance measurement along the centerline of the road. It is a cumulative measurement, meaning it continuously increases in one direction. To ensure the simulation research has mathematically consistent spatial references, we check whether chainage consistently increases along each road. If not, we investigate why and apply a repairing algorithm.

The chainage monotony check consisted of a function that applied the following steps:

1. Read the Excel file road and bridge dataset into Python.
2. Group all the rows that correspond to the same road
3. For each group, iterate over every row to check if the current chainage has a higher or equal value than the previous one
4. If True or False assign a value to a report. The output is a report of the roads that have broken monotony, if any.

The roads dataset was already grouped coherently by road. In contrast, the bridges dataset contained multiple clusters of the same road distributed throughout the data file. This in some cases, broke the consistency of increase in chainage values as shown in Figure 3.1.

Since this difference in organisation between the files could produce chainage monotony breaks in the bridge data file, but not necessarily due to inaccurate values; rather, because of the ordering. The



1	road	km	type	LRPName	name	length	condition	structureN	roadName	chainage	width	structionY
461	N1	456.359	RCC Girder	LRP452i	Barayotoli (:	6.1	A	120588	Dhaka (Jatı	456.359		
462	N1	456.549	RCC Girder	LRP452j	Naiton Par:	23	A	120589	Dhaka (Jatı	456.549		
463	N1	457.365	RCC Girder	LRP453a	Naitom Pai	15.1	A	120592	Dhaka (Jatı	457.365		
464	N1	457.751	RCC Girder	LRP454a	Naitong Pa	9	A	120594	Dhaka (Jatı	457.751		
465	N1	457.751	RCC Girder	LRP454a	NAITANG P	9	A	100897	Dhaka (Jatı	457.751	7.4	1982
466	N1	458.213	RCC Girder	LRP454b	Naitom Pai	9	A	120595	Dhaka (Jatı	458.213		
467	N1	459.681	Box Culver	LRP456a	BUS STANI	1.5	A	100891	Dhaka (Jatı	459.681	10.3	1991
468	N2	12.18	Box Culver	LRP012f	SHAWGHA	3.9	A	103167	Dhaka (Kat	12.18	15.63	2004
3420	Z8705	7.46	Truss with	LRP007b		42.67	B	119282	Bhandaria-	7.46		
3421	Z8705	8	RCC Girder	LRP007e	.	10.3	B	112224	Bhandaria-	8	6.62	1996
3422	N1	8.976	PC Girder E	LRP008b	KANCHPUF	397	C	101102	Dhaka (Jatı	8.976	14.65	1986
3423	N1	17.134	RCC Girder	LRP017b	Langalbanc	159.52	C	119909	Dhaka (Jatı	17.134		
3424	N1	18.742	PC Girder E	LRP019a	MOLLIK PA	40.5	C	109819	Dhaka (Jatı	18.742	9.25	2001
3425	N1	24.393	PC Box	LRP024a	MEGHNA E	900	C	101123	Dhaka (Jatı	24.393	9.2	1989

Figure 3.1: Broken monotony in road N1 identified in disordered bridge dataset

chainage monotony repair function consisted of sorting the entries first by road and then by chainage. The output is a new sorted DataFrame.

Part of ensuring data quality in this step is to revise the output to ensure spatial consistency.

3.3. Invalid Roads in Bridge Dataset

Some roads in the bridge dataset were not located in the roads dataset. Since simulation research relies on modelling a network, isolated bridges will not contribute and could lead to errors during simulation. Therefore, we implemented two functions, one to identify which roads from the bridges dataset were also in the roads dataset, and another function to drop the roads flagged as invalid from the bridges dataframe. The ones that were not were flagged as invalid and dropped from the dataframe.

We declare that removing these bridges does not imply that they are unimportant for our purposes, but that without a link to a road, they are of no use to creating a road network model, and any creation of roads will require a detailed analysis that escapes the scope of this report.

We used the bridges dataset with sorted monotony issues, therefore, the output of this section of the cleaning process was an Excel file with corrected chainage monotony and no invalid roads.

3.4. Inverted Latitude and Longitude Issues

Some of the rows have the values of latitude and longitude inverted in the dataset. A preprocessing step to swap them is required before further processing because it helps reduce the number of outliers detected and corrected in later algorithms. These values themselves are not necessarily incorrect, nor do they need interpolation to correct the position.

Ideally, all bridges in Bangladesh should have latitude values in the range 20.86382 to 26.33338 and longitude values in the range 88.15638 to 92.30153 [1]. All latitudes that fall within the longitude range (and vice versa) are flagged. Afterwards, bridges with inverted coordinates are corrected to represent their true values. The detailed steps are as follows:

- **Create a copy of the dataframe for processing**
- **Detect inverted coordinates for all bridges**

  This is done using the function `detect_bridges_inverted_coordinates(df1)`.

  1. Check latitude values of each row to see whether they fall within the defined longitude range and vice versa.
  2. Flag rows where both longitude and latitude indicate a misplaced condition, excluding general out-of-bounds data.
- **Invert the values of latitude and longitude of detected incorrect bridges**

  The function `swap_lon_lat(df)` is used during this step.

  1. Check the flag column in the dataframe to apply the swap function only to applicable rows.
  2. Swap the values of latitude and longitude.

3. Remove the flag after successful swapping.

- **Log the swapped data for future reference**
- **Return swapped dataframe**

3.5. Out of Boundary Issues

Some of the latitude and longitude values lie outside the boundary of Bangladesh. The approach is similar approach to the inverted latitude-longitude detection, but here the latitudes and longitudes are compared to their respective boundary. This step is done after the inversion fixes to ensure those previously inverted rows are not also detected as out-of-bound.

Here, we will use the corrected road data frame to obtain the value of latitude and longitude data. However, not all the LRPs are present in the dataframe, and even if it exists, it does not guarantee it is the correct values, but for the purpose of roughly correcting the values before further processing it should be sufficient. This condition pose as limitation of this correction method.

The proposed further correction mechanism is to interpolate the adjacent chainage information to calculate the rough position of said LRPs. Alternatively, it could also scrap raw data for the remaining out of bound data. These proposed methods are yet to be implemented in this report. The steps are as the following details:

The steps are as follows:

- **Create a copy of the dataframe for processing**
- **Detect out-of-bound coordinates for all bridges**

  This is done using the function `detect_bridges_out_coordinates(df)`.

  1. Check latitude values of each row to determine whether they fall within the defined latitude range and vice versa.
  2. Flag rows where either longitude or latitude indicates a misplaced condition.
- **Lookup the latitude and longitude values of detected bridges using external reference data**

  The function `fix_far_out_coords(df1, df2)` is used during this step.

  1. Check the flag column in the dataframe to apply the function only to applicable rows.
  2. Query the second dataframe to obtain latitude and longitude values from the same **road** and **LRP** columns.
  3. Remove the flag if the data is found and successfully replaced.
  4. Interpolate approximate latitude and longitude using chainage relationships with adjacent LRPs (*not yet implemented*).
  5. For remaining unreplaced data, scrape the information from raw sources (*not yet implemented*).
- **Log the replaced data for future referencing**
- **Return replaced dataframe**

3.6. Missing Value and Incompleteness Issues

In the datasets, several rows are missing the latitude and longitude information entirely. The algorithm to detect these errors is directly flagging rows where either the latitude or longitude values is NaN. The approach to correct the data is divided into two steps, combining duplicate and out of bounds approach.

Firstly, we check for all rows with NaN values to see whether they have duplicate rows with the same road and LRP name with valid latitude and longitude value. If found, we will use the row with valid value and drop the rows with NaN in the latitude and longitude column. We are aware that there could be instances where the dropped rows may be more updated or contain information on bridge condition, but we deem it sufficient for the next analysis by using the worse case scenario for the bridges condition.

Secondly, we will refer to the corrected road data frame to obtain the value of latitude and longitude data. However, not all the LRPs are present in the dataframe, and even if it exists, it does not necessarily contain the correct values, but for the purpose of roughly correcting the values before further processing it should be sufficient. This condition is a limitation of this correction method.

The proposed further correction mechanism is to interpolate the adjacent chainage information to calculate the rough position of said LRPs. Alternatively, it is also possible to scrap raw data for the remaining out-of-bound data. These proposed methods are yet to be implemented in this report. The steps are as follows:

- **Create a copy of the dataframe for processing**
- **Detect NaN coordinates for all bridges and drop duplicated values**

  The function `drop_redundant_nan_coords(df)` is used during this step.

  1. Filter all rows containing NaN values in latitude and longitude columns.
  2. Check whether the filtered rows contain duplicate entries (same road and same LRPName) but with valid latitude and longitude values elsewhere.
  3. Drop rows with NaN latitude and longitude if duplicate valid entries exist; otherwise retain the rows.
- **Lookup the latitude and longitude values of NaN bridges using external reference data**

  The function `fill_nan_coords(df1, df2)` is used during this step.

  1. Query the second (reference) dataframe to obtain latitude and longitude values using the same **road** and **LRP** columns for all rows containing NaN in either latitude or longitude.
  2. Interpolate approximate latitude and longitude using chainage relationships with adjacent LRPs (*not yet implemented*).
  3. For remaining unreplaced data, scrape the information from raw sources (*not yet implemented*).
- **Log the replaced NaN data for future referencing**
- **Return replaced NaN dataframe**

3.7. Detection of Spatial Outliers

To identify anomalous coordinates in both the road and bridge datasets, the distance between consecutive LRPs was computed using the Haversine distance, which measures great-circle distance on the Earth's surface. This method is preferred over planar Euclidean distance because the dataset spans a national scale, and spherical geometry provides more reliable segment length estimation [4].

For each road or bridge alignment, the median segment length was calculated and used to define a dynamic anomaly threshold:

$$\text{Threshold} = K \times \text{Median Segment Length}.$$

Segments exceeding this threshold were classified as spatial outliers ("jump segments"). A relative threshold was chosen instead of a fixed kilometer value because infrastructure geometries vary considerably across regions; a static threshold would either fail to detect extreme anomalies or incorrectly classify valid segments as errors.

An initial value of $K = 15$ was selected to detect extreme coordinate errors, such as typographical mistakes resulting in points located hundreds of kilometers away. The cleaning procedure was applied iteratively using the same parameter value. Repeating the process allows major distortions to be corrected first, after which additional inconsistencies may become detectable under the same relative threshold. This iterative approach improves spatial consistency while limiting the risk of overcorrection.

3.8. Spatial Outlier Repair Strategy

After detecting spatial outliers based on abnormal segment distances, a structured repair strategy was implemented. The goal is to correct erroneous coordinates while preserving genuine geometric variation. This procedure is applied to both the road and bridge datasets, since both represent linear infrastructure and rely on ordered coordinate sequences.

The detailed steps are as follows:

- **Sort LRPs according to chainage order**

  Before repairing, LRPs are sorted using a custom sorting key to ensure correct spatial sequencing (e.g., LRPS, LRP001, ..., LRPE). This guarantees that distance calculations and corrections follow logical progression.

- **Compute segment distances**

  The Haversine distance between consecutive points is calculated using the function `compute_segments(df)`.

  1. Calculate the median segment length for each road or bridge.
  2. Define a dynamic threshold as $K\times$ median segment length.
  3. Flag segments exceeding this threshold as spatial jumps.

- **Repair consecutive jump segments (Block Repair)**

  When multiple consecutive segments are flagged, the function `block_repair(df)` is applied.

  1. Identify continuous runs of abnormal segments.
  2. Preserve the boundary points surrounding the block.
  3. Linearly interpolate intermediate coordinates between valid endpoints.
  4. Log all modified points.

- **Repair isolated jump segments (Smart Local Repair)**

  For single abnormal segments, the function `smart_single_seg_repair(df)` is used.

  1. Evaluate two correction options: adjusting point i or point $i+1$.
  2. Compute the total local path distortion for both options.
  3. Apply the correction that minimizes geometric distortion.
  4. Log the applied modification.

- **Correct end-point anomalies**

  If the final segment (LRPE) is detected as abnormal, the function `end_point_anomaly_repair(df)` is applied.

  1. Evaluate whether the last segment exceeds the threshold.
  2. Extrapolate the final point using the direction of the previous two points.
  3. Replace the erroneous coordinate and log the correction.

- **Iterative refinement**

  The full repair procedure is executed multiple times using the same threshold parameter ($K = 15$).

  1. Recalculate segment distances after each pass.
  2. Detect newly exposed inconsistencies.
  3. Repeat until no significant spatial jumps remain.

- **Return cleaned dataset**

  The final cleaned dataframe is returned for further analysis and validation.

4

Results and Discussion

4.1. Road Data

Following the preprocessing and cleaning procedures described in Section 3, the roads dataset was transformed into a structurally consistent and spatially analysable format. During preprocessing, duplicate entries and roads with insufficient LRP points were removed, reducing the total number of roads from 877 to 852. The remaining dataset preserves structural integrity and forms the basis for the quality assessments presented in this section.

4.1.1. Chainage Monotony in Roads

After applying the chainage monotony check, the road data showed a consistent increase in values. No rearrangement or modification was needed.

4.1.2. Spatial Outliers and Geometric Distortions

To illustrate the impact of spatial outliers and the effectiveness of the repair strategy, two representative national roads are examined: N1 and N2. These roads were selected not only because they exhibit clear spatial distortions in the raw dataset, but also because they are used extensively in subsequent assignments and they are the main arterial highways of Bangladesh [5], making their geometric correctness particularly important.

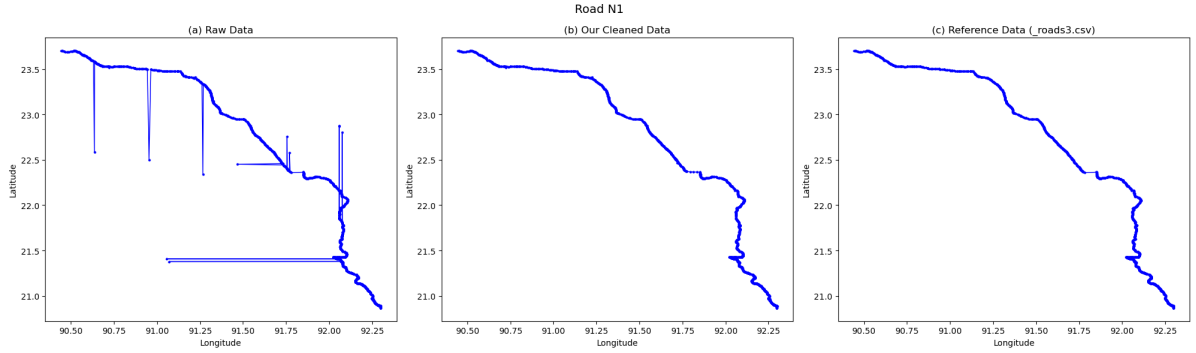


Figure 4.1: Geometry of Road N1: (a) raw data, (b) cleaned data, and (c) reference dataset.

Figure 4.1 presents the geometry of Road N1 before and after cleaning. In the raw dataset (Figure 4.1a), several LRPs are visibly displaced from the main alignment, producing vertical spikes and artificially long segments. After applying the outlier detection and repair procedure (Figure 4.1b), these distortions are removed and the alignment becomes smoother. The cleaned geometry closely matches the provided reference dataset (Figure 4.1c), indicating that the corrections preserve realistic road structure without oversmoothing the geometry.

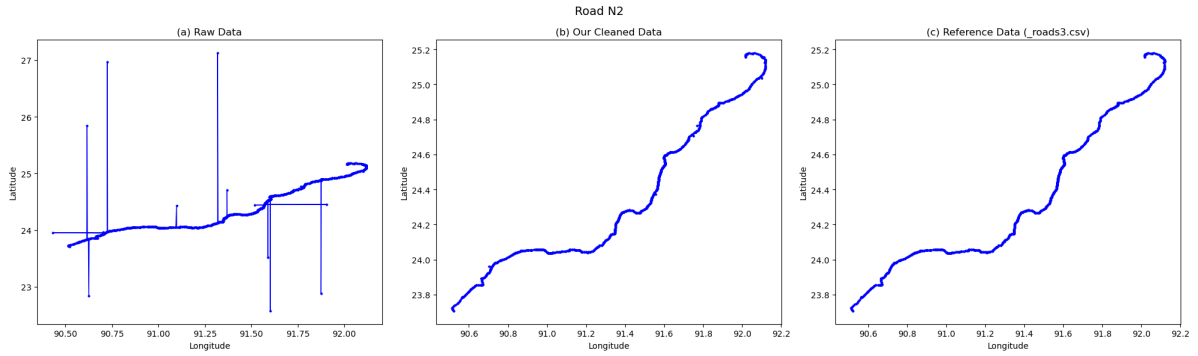


Figure 4.2: Geometry of Road N2: (a) raw data, (b) cleaned data, and (c) reference dataset.

A more extreme case is observed in Road N2 (Figure 4.2). The raw geometry (Figure 4.2a) contains severe discontinuities where individual points are located far from neighboring LRPs, resulting in artificial vertical lines and highly inflated segment lengths. In some instances, segment distances exceeded several hundred kilometers, which is physically implausible within a national road network. After cleaning (Figure 4.2b), these anomalies are eliminated, and the road alignment becomes spatially coherent. The resulting geometry closely resembles the reference dataset (Figure 4.2c).

4.1.3. Network-Level Improvement

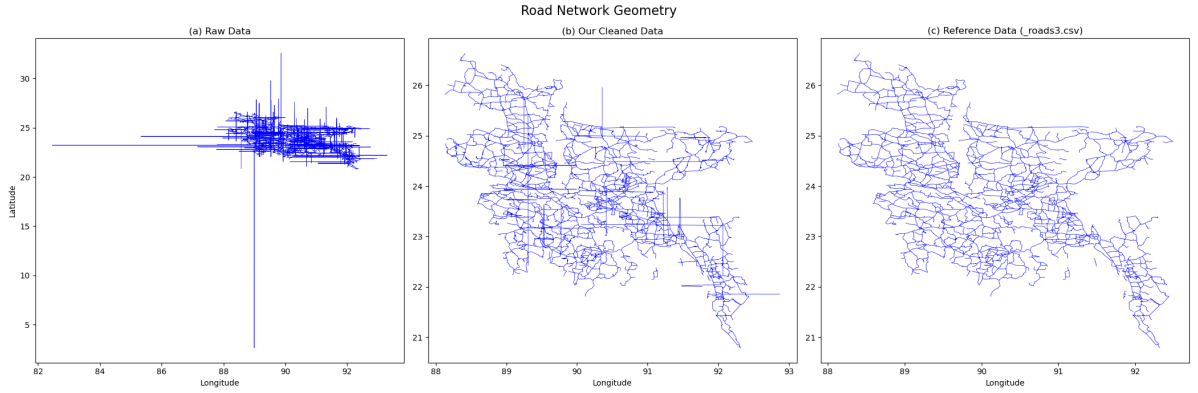


Figure 4.3: Road network geometry: (a) raw dataset, (b) cleaned dataset, and (c) reference dataset.

While the case studies of Roads N1 and N2 demonstrate the correction of severe local distortions, the broader impact of the cleaning procedure is best observed at the national scale. Figure 4.3 shows the geometry of the entire road network before and after cleaning, alongside the provided reference dataset. In the raw data (Figure 4.3a), numerous artificial spikes and long disconnected segments are visible, caused by misplaced LRPs and extreme coordinate errors. These anomalies significantly distort the network's spatial structure.

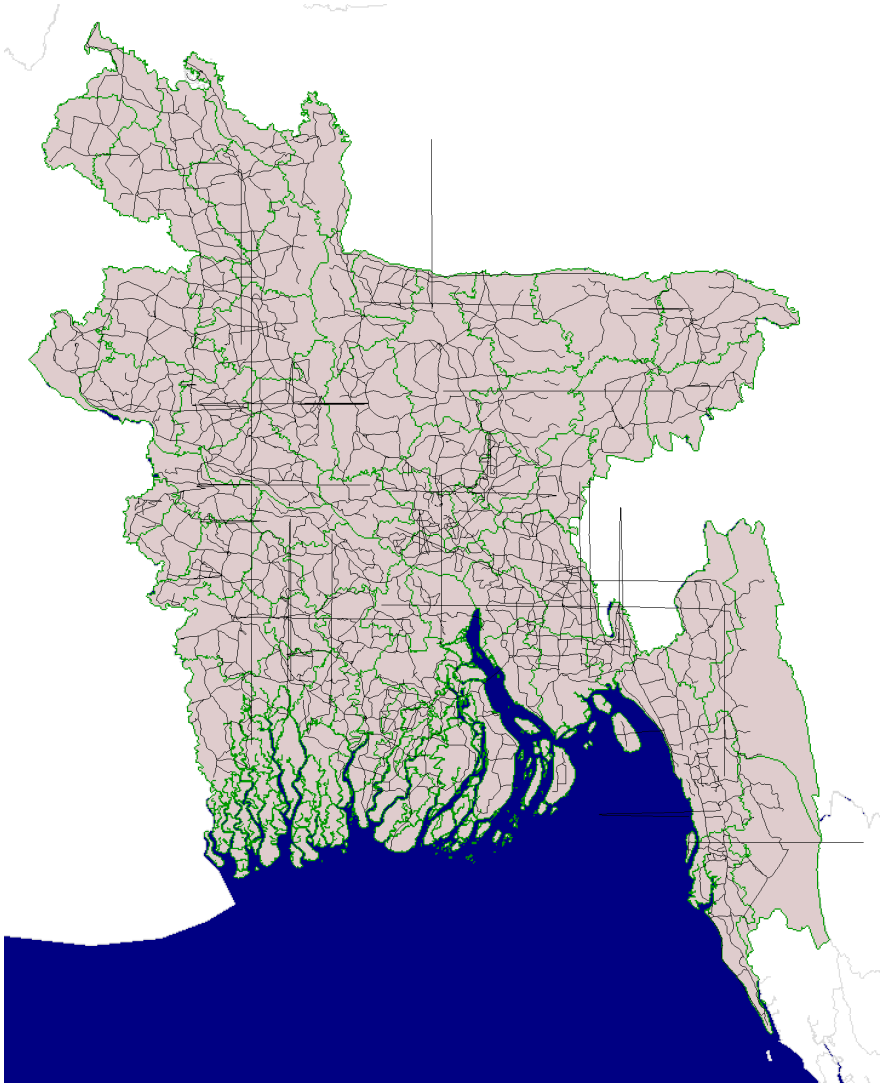


Figure 4.4: Visualisation of the cleaned road dataset in the provided Java-based simulation environment.

After applying the iterative outlier detection and repair procedure (Figure 4.3b), the majority of these extreme distortions are removed. The cleaned network exhibits substantially improved spatial continuity, with road alignments forming coherent and geographically plausible trajectories. Most large vertical spikes and implausibly long segments are eliminated, resulting in a geometrically more consistent national network.

However, the current algorithm does not resolve all anomalies. Minor irregularities remain in certain regions, particularly where local geometry is complex or where LRP spacing is sparse. This limitation stems from the use of a median-based adaptive threshold, which is designed to robustly detect extreme outliers but may miss subtler distortions, risking overcorrection.

The cleaned dataset was subsequently loaded into the provided Java-based simulation environment by replacing the original road file. As shown in Figure 4.4, the simulation visualises the cleaned geometry directly without additional modification. This confirms that the dataset is structurally consistent and compatible with the intended modelling framework. Although the Java visualisation reflects the same underlying cleaned data, it demonstrates that the repaired network integrates smoothly into the simulation workflow.

Taken together, the results indicate that the applied repair strategy substantially improves the geometric integrity of the road dataset while maintaining structural consistency for downstream applications, even though complete anomaly removal cannot be guaranteed.

4.2. Bridge Data

The bridge dataset does not require any specific transformation before processing. However, given the amount of information in it, preprocessing needs to be carried out in more detailed steps than for the roads dataset. The handling of duplicate values, monotony, and missing values needs to consider the information required for further analysis.

4.2.1. Duplicate Values

Some rows were fully identical, meaning the same bridge was recorded multiple times without any extra detail. In these cases, the first occurrence sufficiently represented the bridge, so the remaining identical rows were removed.

Another group consisted of rows duplicated across the core analytical attributes (*road, name, chainage, condition, LRPName, latitude, longitude*) but differing only in secondary columns such as zone, circle, or division. These additional fields do not influence structural or spatial analysis. Therefore, one representative row was kept, and the others were removed.

The cleaned table shows that only one instance of each coordinate pair remains. Each road reference point is now represented once, preserving the spatial structure while eliminating repetition.

The duplicated rows that were retained are shown in Table 4.1.

Table 4.1: Representative rows retained after duplicate removal

Road	LRPName	Latitude	Longitude
R556	LRP015a	25.582305	89.134611
Z5509	LRP026a	25.120222	89.145611
N603	LRPS c	24.370527	88.722694
Z8052	LRP009c	22.362222	90.234444
N5	LRP388b	25.753417	88.713667
N8	LRP123a	23.003639	90.222194
Z6034	LRPS b	24.163500	89.441972
N2	LRP275a	25.158917	92.100139
N502	LRP029c	24.592473	89.236168

4.2.2. Chainage Monotony in Bridges

After applying the chainage monotony check, the bridge data showed 507 roads with broken monotony. After applying the chainage monotony repair function, the new DataFrame showed only one road with broken monotony, road R750.

After checking this road, the issue is that one entry has no chainage value. However, it does have an assigned LRP (LRP006b), length, type of structure, among other information. Given that there is another row entry with the same LRP, length, type of structure and name, we conclude it is a duplicate. Therefore, we drop this entry with no chainage value. After this, chainage monotony is maintained across all bridges in the dataset, and an Excel file is exported containing the processed data.

4.2.3. Removal of Invalid Roads in Bridge Dataset

This step removed 29 roads from the bridges dataset because they do not exist in the roads dataset. Four roads are Regional roads, and the remaining 24 are Zilla roads. 551 bridges were removed in total. Since most of the bridges removed were on minor roads and none were on National roads, this could be considered non-critical, and we can expect this removal not to affect the simulation results. In the following Figure 4.5 we can observe the list of roads removed.

```
551 bridges are in roads that don't exist on the roads dataset
These roads don't exist on the roads dataset:
 ['R505' 'R680' 'R750' 'R856' 'Z1006' 'Z1090' 'Z1211' 'Z1463' 'Z1503'
 'Z1613' 'Z1632' 'Z1705' 'Z2022' 'Z2033' 'Z2063' 'Z3614' 'Z5071' 'Z5073'
 'Z5208' 'Z5458' 'Z5459' 'Z5478' 'Z6801' 'Z6814' 'Z6815' 'Z7048' 'Z7049'
 'Z8711' 'Z8948']
```

Figure 4.5: Results from cleaning algorithm to remove invalid roads from bridge data

4.2.4. Inverted Latitude and Longitude Values

The detection step identified rows where the latitude fell within the expected longitude range and the longitude fell within the expected latitude range for Bangladesh.

After processing, the values were inverted, producing latitude values around 24–25 and longitude values around 90–91. This placed the bridges back within the country's geographic bounds without interpolation or relocation. The positions were not recalculated or estimated, improving spatial accuracy while preserving the original measurement data.

The corrected coordinate samples are presented in Table 4.2.

Table 4.2: List of swapped inverted coordinates

Road	LRPName	Old Lat	Old Lon	New Lat	New Lon
R241	LRP026a	91.544194	24.773694	24.773694	91.544194
R241	LRP027a	91.542389	24.786833	24.786833	91.542389
R241	LRP028a	91.541778	24.790944	24.790944	91.541778
R241	LRP028c	91.542083	24.796694	24.796694	91.542083
R241	LRP029a	91.543889	24.805111	24.805111	91.543889
R241	LRP031a	91.538722	24.816556	24.816556	91.538722
R241	LRP032a	91.534361	24.823833	24.823833	91.534361
R241	LRP032b	91.530556	24.828056	24.828056	91.530556
R241	LRP033a	91.527917	24.830944	24.830944	91.527917
R241	LRP033b	91.524083	24.835222	24.835222	91.524083
R241	LRP034a	91.519389	24.842889	24.842889	91.519389
R241	LRP035a	91.518194	24.844972	24.844972	91.518194
R241	LRP035c	91.516167	24.849861	24.849861	91.516167
R241	LRP036a	91.514083	24.855083	24.855083	91.514083
R241	LRP038a	91.500417	24.867917	24.867917	91.500417

4.2.5. Detection of Out-of-Bounds Coordinates

After correcting inverted coordinates, another group of records still appears outside Bangladesh's geographic extent. Many entries contained coordinates of 0, while others were located far from the country.

The procedure did not attempt to reposition these points because their locations cannot be reliably inferred from the available information. Instead, they were marked as invalid and prepared for later handling, either interpolation using road chainage or removal. The list of detected out-of-bounds records is presented in Table 4.3.

Table 4.3: List of detected out-of-bounds coordinates

Index	Road	LRPName	Latitude	Longitude
10716	Z1405	LRP054a	0.000000	0.000000
11958	Z1811	LRP149a	0.000000	0.000000
11960	Z1811	LRP149d	0.000000	0.000000
11966	Z1811	LRP150b	0.000000	0.000000
11967	Z1811	LRP151a	0.000000	0.000000
13072	Z2812	LRP023a	15.13799	56.12014
13074	Z2812	LRP024a	14.75649	54.70584
13083	Z2812	LRP026d	13.38416	49.61830
13090	Z2812	LRP029b	12.11781	44.92361
13093	Z2812	LRP029f	11.75221	43.56824
13095	Z2812	LRP030a	11.61974	43.07717
13120	Z2813	LRP011a	0.000000	0.000000
13122	Z2813	LRP012b	0.000000	0.000000
13125	Z2813	LRP018a	0.000000	0.000000
13142	Z2813	LRP024c	0.000000	0.000000
19863	Z7718	LRP038a	0.000000	0.000000
20453	Z8203	LRPSa	19.09852	72.87707
20455	Z8203	LRP001a	0.000000	0.000000
20457	Z8203	LRP001c	0.000000	0.000000
20459	Z8203	LRP002a	0.000000	0.000000
20462	Z8203	LRP003a	0.000000	0.000000
20464	Z8203	LRP003c	0.000000	0.000000
20466	Z8203	LRP004a	0.000000	0.000000
20468	Z8203	LRP005a	0.000000	0.000000

4.2.6. Handling Missing Spatial Coordinates

Another issue found in the dataset was the absence of spatial information. Several rows contained no latitude and longitude values, meaning the location of the corresponding LRP could not be determined directly.

The resulting table shows LRPs whose latitude and longitude are entirely missing. Since no matching valid entries were available for these cases, they could not be corrected automatically. They were therefore left identified for later handling using road reference data.

Rows that could be resolved through duplication were cleaned, while rows without any spatial reference were separated from usable spatial data. This prepares the dataset for later processing without introducing artificial coordinates.

List of records with missing spatial coordinates are shown in Table 4.4.

Table 4.4: Records with missing latitude and longitude values

Road	LRPName	Latitude	Longitude
N208	LRP058b	NaN	NaN
R505	LRP007a	NaN	NaN
R680	LRP030a	NaN	NaN
R680	LRP034b	NaN	NaN
R680	LRP041b	NaN	NaN
R750	LRPSb	NaN	NaN
R750	LRP025a	NaN	NaN
R856	LRP009c	NaN	NaN
R856	LRP009e	NaN	NaN
R856	LRP010b	NaN	NaN

4.2.7. Spatial Outlier and Geometric Distortions

To evaluate the impact of spatial repair, several representative national roads were examined, namely N1, N2, and N6. N1 and N2 were selected because the raw data showed clear geometric inconsistencies that would affect distance calculation and network connectivity, while N6 was used to demonstrate that the algorithm does not distort already correct information.

As shown in Figure 4.6, the raw geometry of road N1 contains long vertical spikes and straight segments disconnected from the main alignment. After the repair process, the spikes disappear and the road becomes a continuous curve following a realistic trajectory. The corrected alignment forms a coherent path, however small section are not fully corrected which may require further manual check.

A stronger distortion is visible in road N2. The raw dataset contains extreme vertical segments and sharp jumps across large distances, as shown in Figure 4.7. After correction, the resulting geometry better represents a physical road structure.

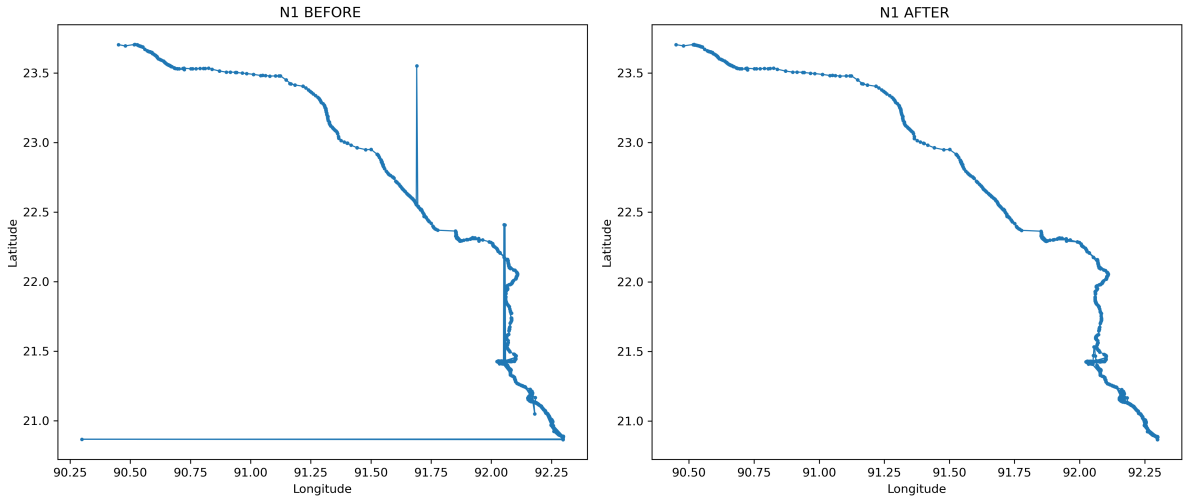


Figure 4.6: Road N1 before and after spatial repair

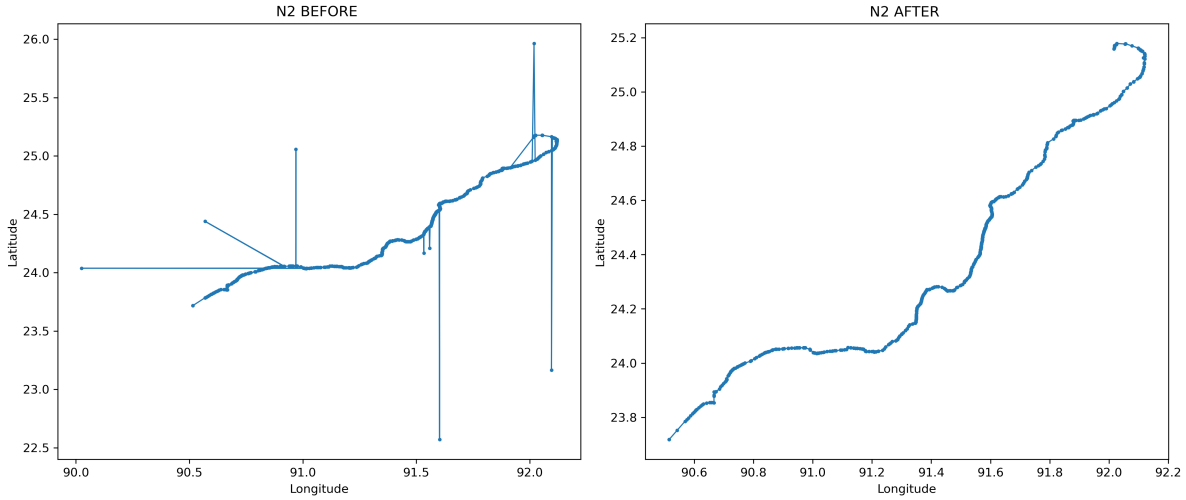


Figure 4.7: Road N2 before and after spatial repair

Road N6 (Figure 4.8) shows a case where the correction mainly adjusts local irregularities while keeping the original curvature intact. This indicates that the repair process only process outlier points rather than altering the full geometry.

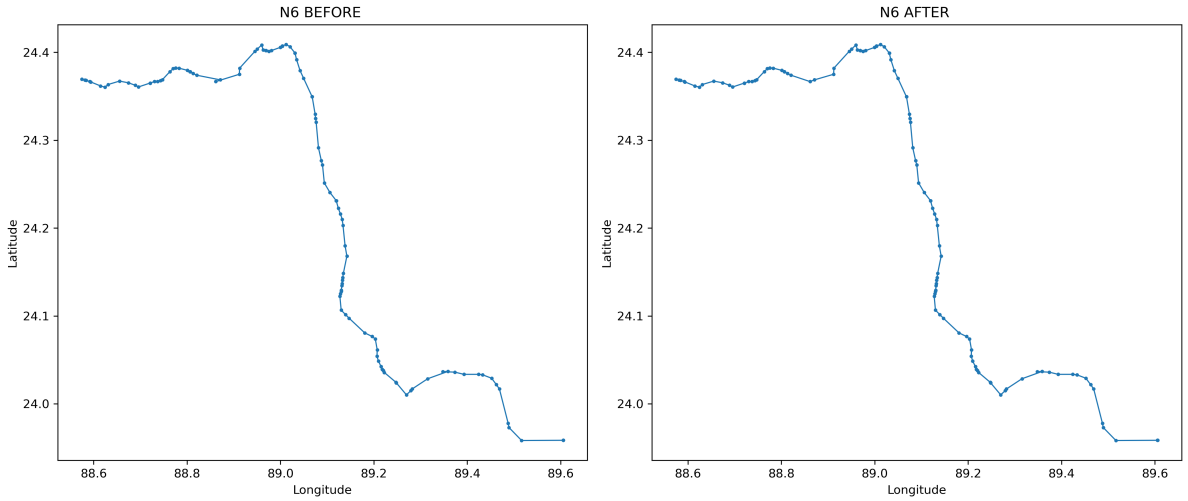


Figure 4.8: Road N6 before and after spatial repair

4.2.8. Network Level Improvement

The effect of the repair is visible at the national network scale. After the procedure, the points align across the country following road as seen in Figure 4.9. Some minor irregularities remain in sparse regions, but the major geometric distortions are eliminated.

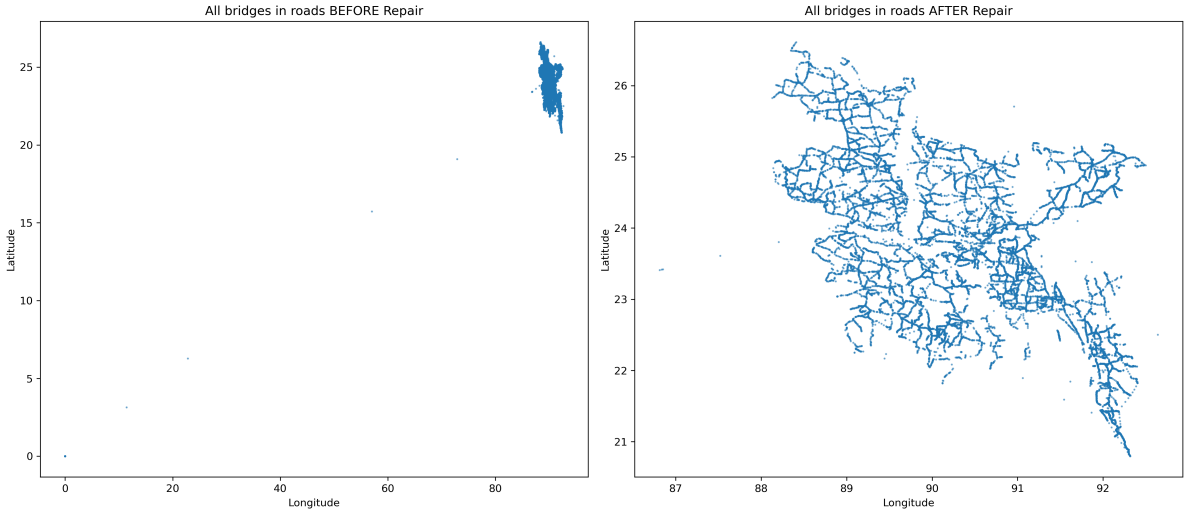


Figure 4.9: Bridges network before and after spatial repair

In the original dataset, as shown in Figure 4.10, many bridges are clearly positioned in the ocean. After applying the repair process, the number of offshore bridges is significantly reduced as shown in Figure 4.11. The spatial distribution becomes more consistent with the actual infrastructure network.

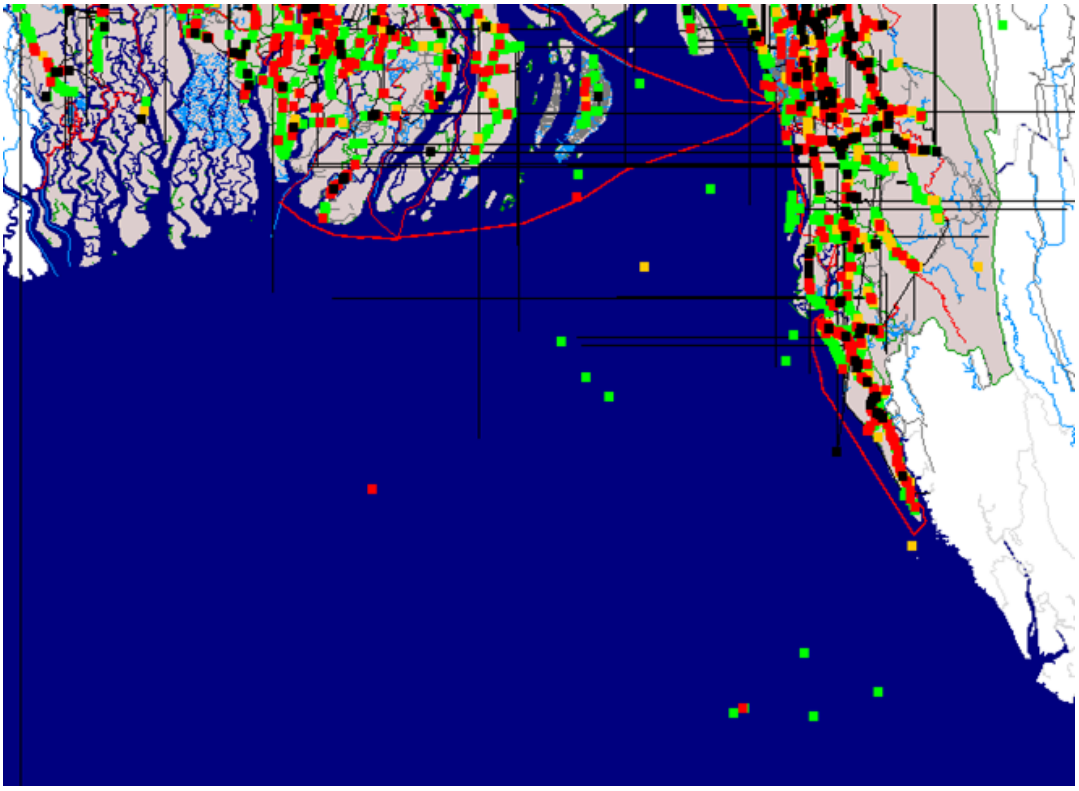


Figure 4.10: Bridge locations near the coastal region before cleaning

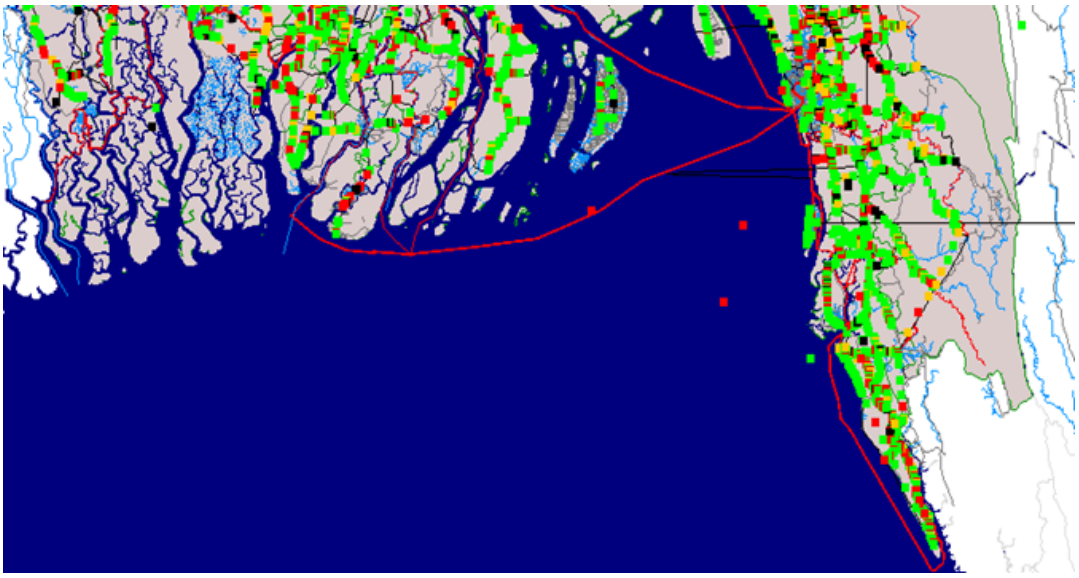


Figure 4.11: Bridge locations near the coastal region after cleaning

4.2.9. Remaining Issues, Missing Values and Out-of-Bound Bridges

Several bridges remain in the ocean, while additional points are still outside the Bangladesh area. This reflects a limitation where the correction relies on nearby road reference points and logical proximity. When no reliable reference exists or when the original record lacks sufficient contextual information, the method cannot confidently reposition the bridge. As a result, some incorrect points persist. The remaining points are shown in table 4.5.

Table 4.5: Missing and Invalid Coordinate Records

Index	Road	LRP	Latitude	Longitude
5327	R203	LRP082b	NaN	NaN
11906	Z1811	LRP149a	0.000000	0.000000
11907	Z1811	LRP149d	0.000000	0.000000
11913	Z1811	LRP150b	0.000000	0.000000
11914	Z1811	LRP151a	0.000000	0.000000
13061	Z2813	LRP011a	0.000000	0.000000
13062	Z2813	LRP012a	0.000000	0.000000
13063	Z2813	LRP012b	0.000000	0.000000
13064	Z2813	LRP012b	3.144097	11.40420
13065	Z2813	LRP012d	6.288194	22.80840
13066	Z2813	LRP018a	15.73659	57.02781
16061	Z5211	LRP030c	NaN	NaN
20374	Z8203	LRPSa	19.09852	72.87707
20376	Z8203	LRP001a	0.000000	0.000000
20378	Z8203	LRP001c	0.000000	0.000000
20380	Z8203	LRP002a	0.000000	0.000000
20383	Z8203	LRP003a	0.000000	0.000000
20385	Z8203	LRP003c	0.000000	0.000000
20387	Z8203	LRP004a	0.000000	0.000000
20389	Z8203	LRP005a	0.000000	0.000000
20395	Z8204	LRP009f	NaN	NaN
20930	Z8708	LRP074a	NaN	NaN

Moreover, after reviewing the Java visualisation output, several LRPs were identified as still containing coordinate inaccuracies that could be feasibly manually corrected.

For Road Z1071, the bridge dataset coordinates were not aligned with the corrected LRP roads dataset. Therefore, the bridge coordinates were adjusted to match the verified coordinates from the updated LRP roads dataset.

In addition, several typographical errors in latitude and longitude values were detected and corrected as follows:

- R164 – LRPSa: Longitude corrected from 92.640028 to 92.140028.
- Z1076 – LRP001a: Latitude corrected from 21.410806 to 21.910806.
- Z1220 – LRP002b: Latitude corrected from 25.70527778 to 23.70527778.
- Z1619 – LRPSa: Latitude corrected from 23.52455556 to 22.52455556.

However, for the remaining problematic LRPs, more intensive manual verification and correction would be required. These limitations, including constraints related to missing contextual information and reference inconsistencies, are discussed in greater detail in Chapter 5

5

Conclusion and Reflection

5.1. Conclusion

This report examined data quality issues identified in Bangladesh's road and bridge datasets. The analysis focused on syntactic and semantic inconsistencies most relevant to spatial referencing. A set of functions was implemented to systematically detect, flag, and repair these issues.

The corrections addressed broken chainage monotony, invalid roads, inverted coordinates, out-of-boundary values, missing (NaN) entries, duplicate records, and inconsistencies in bridge locations. Priority was given to issues we could correct in an automated or semi-automated manner and that affected a substantial portion of the datasets. This approach ensured that implemented repairs had a higher impact on the overall data quality.

5.2. Limitations

Certain issues were not fully resolved or even fully identified due to the level of detail and external validation required. For example, verifying bridge locations against satellite imagery was not carried out in this report.

On the other hand, data quality issues, such as duplicate bridges, were identified, but not all were resolved because they required different repair treatments. A duplicate could refer to an improvement in infrastructure, a different part of the same bridge, or a register of the same bridge generated in two different ways. Consequently, the report focused on identifying issues and developing processes that could be addressed reliably within the available resources and time.

Additionally, representing Bangladesh's territorial extent as a rectangular bounding box poses a limitation. While this approach is computationally efficient, it may incorrectly classify certain LRPs as being inside or outside the national boundary.

Furthermore, at the end of the processing pipeline, several LRPs could not be corrected through interpolation or coordinate adjustment. These include LRPs with missing (NaN) or zero latitude and longitude, as well as LRPs with coordinates that remain outside the Bangladesh boundary and were not successfully handled by the algorithm in which will require manual checking given the available data during the processing.

5.3. Possible Improvements and Extensions

The selection of corrections was guided by the requirements of the subsequent reports, particularly those involving the modelling of roads N1 and N2, and our available resources. However, many data quality issues can still be improved, as mentioned, by verifying with satellite data and with more detailed approaches.

Future improvements could address the limitation of spatial boundaries by integrating an official Bangladesh

polygon shapefile directly into the preprocessing workflow. In our Python test implementation, projecting all LRPs individually into the Bangladesh polygon took approximately 35 minutes. Although computationally intensive, optimised spatial indexing techniques or parallel processing could reduce execution time and make full point-in-polygon validation feasible.

For LRPs that remain unresolved after interpolation, a more robust solution would be to refer back to the original raw datasets in the RMMS and BMMS folders. By extracting and cross-validating the original coordinate information, these LRPs could potentially be repositioned more accurately. This would strengthen spatial reliability while preserving important attribute data needed for downstream modelling tasks.

A more accurate approach would involve using a polygon shapefile (SHP) of Bangladesh and performing a point-in-polygon projection for each LRP. This would allow a precise determination of whether each LRP lies within the actual national boundary.

Additionally, implementing a structured flagging system for spatial anomalies would improve transparency in subsequent modelling stages. Rather than excluding uncertain data, explicitly tracking unresolved LRPs enables sensitivity analyses and ensures that model results can be interpreted with appropriate caution.

Overall, the work shows that targeted data cleaning efforts can substantially improve the usability of the dataset, while highlighting the limitations and trade-offs inherent in real-world data quality management.

We also highlight the importance of including data quality reports as part of any research to ensure transparency into the initial data quality and the cleaning process to improve it, as well as the replicability of the methods.

References

[1] *Bangladesh latitude and longitude*. https://latitudelongitude.org/bd/. Accessed: 2026-02-20.

[2] Yilin Huang and Alexander Verbraeck. *2.1 Data Quality*. Feb. 2026.

[3] James R Marsden and David E Pingry. “Numerical data quality in IS research and the implications for replication”. In: *Decision Support Systems* 115 (2018), A1–A7.

[4] Dwi Arman Prasetya et al. “Resolving the shortest path problem using the haversine algorithm”. English. In: *Journal of Critical Reviews* 7.1 (Jan. 2020), pp. 62–64. ISSN: 2394-5125. DOI: 10.22159/jcr.07.01.11.

[5] Saifuddin Saif and Yashab Osama Rahman. “The Asian Highway: A pipe dream on paper”. In: *The Business Standard* (Sept. 28, 2021). URL: https://www.tbsnews.net/bangladesh/infrastructure/asian-highway-pipe-dream-paper-308839 (visited on 02/20/2026).

A

Data and Code Availability

All data processing, analysis, and visualization code used in this study are openly available to ensure transparency and reproducibility. The complete codebase, along with documentation and links to the underlying datasets, is hosted in a public GitHub repository at:

`https://github.com/taufik-my/ASIM_Team14/tree/main/EPA133a-G14-A1`

B

Use of AI Tools

B.1. How AI Was Used

- AI (Gemini and ChatGPT) was used to improve writing style and spell check. Grammarly was also used to review our writing. We reviewed the output to ensure consistency with our writing style.
- AI (Gemini and ChatGPT) was used to reformat our writing in LaTeX where needed.
- AI (Gemini and ChatGPT) was used to help debug issues in our code.

B.2. How AI Was Not Used

- AI was not used to generate the final results or interpret them.
- AI was not used to identify errors in the dataset.
- AI was not used to make unverified claims.

C

Team Contributions

This assignment was completed collaboratively by all members of Team 14. The table below summarises the primary contributions of each team member. All members participated in discussions, validation of results, and revision of the final report.

Table C.1: Overview of team member contributions

Student Name	Main Contribution
Bayu Jamalullael	Integrating the algorithm into one notebook, adding extra processing to check the boundary with the SHP file and processing after the first visualisation, write a discussion about limitations, possible improvements, and extensions
Brenda Escobar Arriaga	Chainage monotony, invalid road removal, introduction and conclusion
Brian Parsaoran	Bridge data cleaning, preprocessing, spatial correction, and Java simulation
Taufik M. Yusup	Road data cleaning and spatial correction; Github repository management; report preparation and LaTeX formatting
Zhafran Sidik	Bridge data cleaning, writing Chapter 2 and overall report consistency review