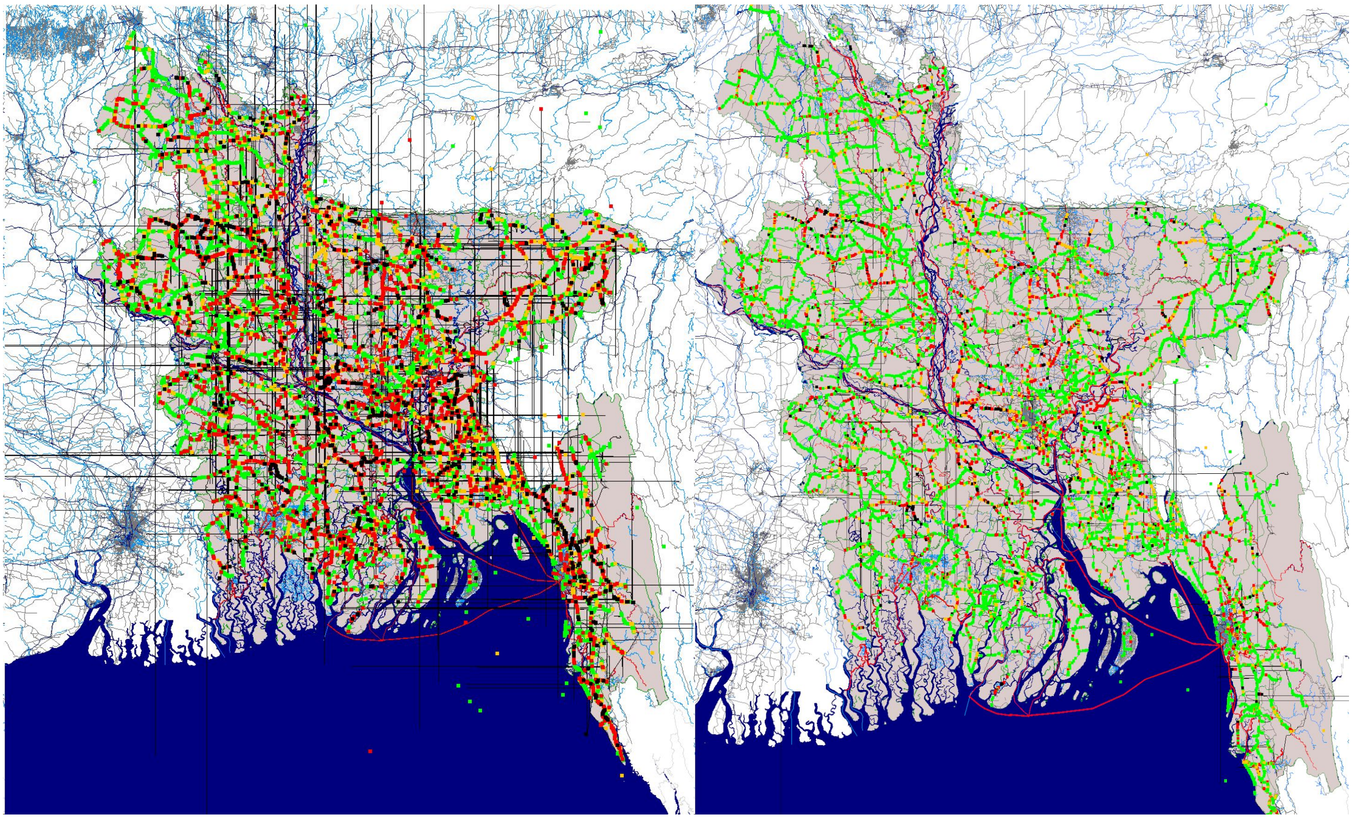
Assignment 1. Preparing Data for Data-Driven Simulation



Brennen Bouwmeester	4446461
Omar Quispel	4107950
Irene Schmidt	4288009
Kevin Su	4438108

In order to investigate the criticality and vulnerability of Bangladesh’s transport infrastructure and testing possible policies for this system, it is important to have a reliable data stream. In this report, the process of cleaning this data for the infrastructure is explained. As it is important to use up to date data for the transport infrastructure, the cleaning of the data is done in a reproducible way rather than a one-time process. This is done by creating algorithms that clean the data for the user, which can be used over and over again if updated data arises.

The report is split up into two parts; the first part focuses on the bridges in Bangladesh and their position and other attributes. The processing of the data on these bridges is split up into four sections that each treat a different kind of issue.

The second part of the report focuses on the roads in Bangladesh. Here only two distinct issues have been found, to which two possible solutions will be explained.

Both parts have python scripts that solve (some of) the issues addressed. The bridge corrections can be found in the “Bridge correction” file. The roads corrections are split up into two files namely: “Road Corrections - Algorithm 1” and “Road Corrections - Algorithm 2”. The order in which the scripts should be run is: “Road Corrections - Algorithm 1” followed by “Road Corrections - Algorithm 2”, and then the “Bridge Correction”.

1. Bridges

As the end result makes use of the BMMS_overview.xlsx file, this file has been used to discover and repair data issues. Multiple data issues have been identified, which are divided into the following categories:

1. coordinates
2. bridge names
3. duplicated bridges
4. other issues

For each of these categories, some examples will be given and an explanation will be given of what should, or has been done to fix the issues.

To get a better overview of the size of the possible problems, in table 1, the most common issues their category and the percentage of data with this issue that is identified.

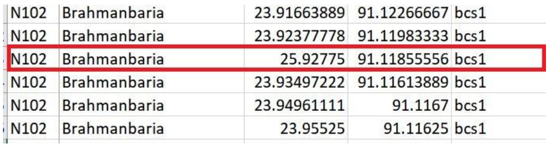
Table 1. Most common identified data issues for bridges with type and percentage of data with this issues

category	data issue	type	%
1	inaccurate coordinates	semantic inaccurate	<1
2	inconsistent names	semantic mapping/inconsistent	47
	missing names	semantic incomplete	1.6
3	duplicated bridges	semantic mapping inconsistent pragmatic timeliness	14
4	missing length	semantic incomplete pragmatic incomplete	<0.1
	missing span, width, constructionYear	semantic incomplete pragmatic incomplete	15

1.1 Coordinates

1.1.1 Misplaced bridges (semantic accuracy)

In some cases, either the longitude, latitude or both have unrealistic values, which sometimes are even outside of Bangladesh, see figure 1. These are most likely human errors that occurred during the creation of the data. Other similar errors can be identified which aren't always easy to find, nor fix. To remedy this a generic script is used to clean up the bridge coordinates. This is done on the premise that bridges are linked to a specific road LRP. The script updates all bridge coordinates in the BMMS_overview.xlsx file to their matching LRP coordinates in the Roads_InfoAboutEachLRP.csv.



N102	Brahmanbaria	23.91663889	91.12266667	bcs1
N102	Brahmanbaria	23.92377778	91.11983333	bcs1
N102	Brahmanbaria	25.92775	91.11855556	bcs1
N102	Brahmanbaria	23.93497222	91.11613889	bcs1
N102	Brahmanbaria	23.94961111	91.1167	bcs1
N102	Brahmanbaria	23.95525	91.11625	bcs1

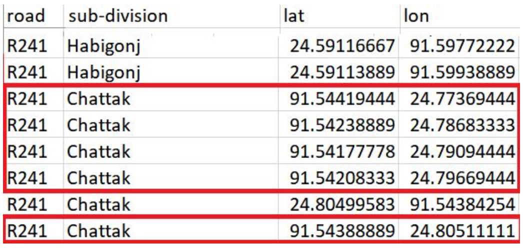
Figure 1. data issue example: inaccurate bridge coordinates

The approach remedies a lot of issues, such as the inconsistency between the placement of the bridges and the roads. The algorithm put the bridges on the location of their accordingh road, which means the success is highly reliant on the coordinates in the LRP file being correct. When this is not the case, it could even create new problems. As such, the road coordinates should be cleaned up before application of this script. This generic script should also be applied first as it is likely to overwrite any work done by other bridge related scripts. More on this can be found in the part on roads in this report.

Further optimisations to the script require a better understanding of the data so that more conditions can be applied to the code. These could be conditions such as further interaction when differences in coordinates are found instead of just changing the coordinates. Currently, LRPs in the bridge file that are not found in the road file are ignored. Depending on the frequency of this occurring, these cases should be handled differently.

1.1.2 Longitude, latitude switch (semantics accuracy)

A specific issue with the coordinates is the accidental switch between the longitude and the latitude, see figure 2. These cases can easily be found by looking for rows that have a latitude greater than 30 and a longitude lower than 80. In order to fix this issue, a new column has been added that copies the lon and the lat, which is necessary to switch them. After the new columns have been created, the lat is set to the copy of the lon and the lon is set to the copy of the lat. After this, the copies are deleted again to end with a data frame that has not been changed in properties.



road	sub-division	lat	lon
R241	Habigonj	24.59116667	91.59772222
R241	Habigonj	24.59113889	91.59938889
R241	Chattak	91.54419444	24.77369444
R241	Chattak	91.54238889	24.78683333
R241	Chattak	91.54177778	24.79094444
R241	Chattak	91.54208333	24.79669444
R241	Chattak	24.80499583	91.54384254
R241	Chattak	91.54388889	24.80511111

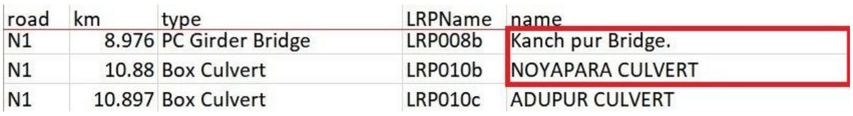
Figure 2. data issue example: latitude and longitude switched

1.2 Names

In the name column, data issues with the type semantic mapping inconsistency can be identified, since some names are written in only capital letters, some are with both capital and small letter and some are written as e.g. ‘.’, ‘0’, ‘--’, see figure 3 and figure 4. Next to this there is also semantic incompleteness identified, figure 4 shows that some name values are missing.

The bridge name data has a low priority, because bridge names are not necessary for the identification of the bridges. Bridges can be identified by the combination of the road and the LRPName. Since bridges can be identified without the name, this data does not contribute to assessing the criticality and vulnerability of Bangladesh's transport infrastructure.

No changes have been made to the bridge name data, because it has a low priority and at the moment it is not critical data. The name data has been kept for future purpose. If the data becomes more critical in the future changes can still be made.



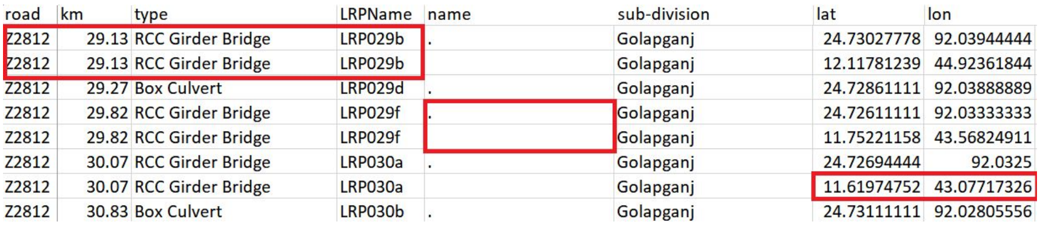
road	km	type	LRPName	name
N1	8.976	PC Girder Bridge	LRP008b	Kanch pur Bridge.
N1	10.88	Box Culvert	LRP010b	NOYAPARA CULVERT
N1	10.897	Box Culvert	LRP010c	ADUPUR CULVERT

Figure 3. data issue example: inconsistent names

1.3 Duplicated bridges

Some of the bridges in Bangladesh have more than one row in the data. These can be identified as bridges with the same road, LRP and chainage combination, see figure 4. This duplication is caused by the inserting new rows in the dataset with already existing bridges that have new data. These new entries might update the quality of a bridge, which is why they are valuable. However, these new bridge rows often have missing data and are semantic incomplete.

By sorting the bridges on their StructureNr, the oldest bridge entries are first in the dataframe. In order to pinpoint the duplicate bridges a new variable was added existing of the sum of the strings “road”, “lrp” and “chainage”, which creates for example the following entry “Z218LRP029b1.234”. After identifying the duplicates, only the first entry will be kept, as visual inspection has shown that the oldest entry always at least has the coordinates which are crucial to the system. This is, however, not the best solution. In order to keep the timeliness of the data, in future the bridges should be merged into one, by using the information of the newest entry as long as possible. This way, updates on for example the quality of a bridge are used as well.



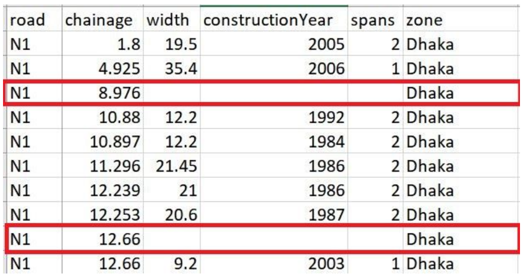
road	km	type	LRPName	name	sub-division	lat	lon
Z2812	29.13	RCC Girder Bridge	LRP029b	.	Golapganj	24.73027778	92.03944444
Z2812	29.13	RCC Girder Bridge	LRP029b		Golapganj	12.11781239	44.92361844
Z2812	29.27	Box Culvert	LRP029d	.	Golapganj	24.72861111	92.03888889
Z2812	29.82	RCC Girder Bridge	LRP029f	.	Golapganj	24.72611111	92.03333333
Z2812	29.82	RCC Girder Bridge	LRP029f		Golapganj	11.75221158	43.56824911
Z2812	30.07	RCC Girder Bridge	LRP030a	.	Golapganj	24.72694444	92.0325
Z2812	30.07	RCC Girder Bridge	LRP030a		Golapganj	11.61974752	43.07717326
Z2812	30.83	Box Culvert	LRP030b	.	Golapganj	24.73111111	92.02805556

Figure 4. several data issue examples: duplicated bridges, missing names and inaccurate coordinates

1.4 Other issues

1.4.1 Missing data

Other missing data has been identified in the width, constructionYear, span and length columns, see figure 5. This missing data is semantic incomplete as well as pragmatic incomplete, because it can have a critical contribution to purpose of the data use, the assessment of the criticality and vulnerability of Bangladesh's transport infrastructure. Data on width and length give information about the number of vehicles the bridge can handle and the data on constructionYear and span gives information about the vulnerability of a bridge. Unfortunately this issue has not been fixed yet, since it is difficult and time-consuming to find the data that is missing. Dropping the bridges is also not an option, since 15% of the bridges have to be dropped to fix the issue, see table 1. The issue could be fixed by extracting the missing data from the raw data files on bridges, this is a future step in the data preparation process.

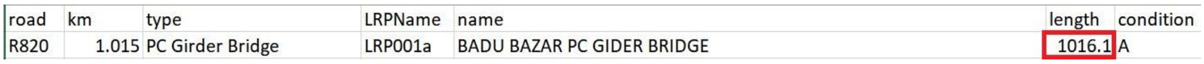


road	chainage	width	constructionYear	spans	zone
N1	1.8	19.5	2005	2	Dhaka
N1	4.925	35.4	2006	1	Dhaka
N1	8.976				Dhaka
N1	10.88	12.2	1992	2	Dhaka
N1	10.897	12.2	1984	2	Dhaka
N1	11.296	21.45	1986	2	Dhaka
N1	12.239	21	1986	2	Dhaka
N1	12.253	20.6	1987	2	Dhaka
N1	12.66				Dhaka
N1	12.66	9.2	2003	1	Dhaka

Figure 5. data issue example: missing data on width, construction year and span

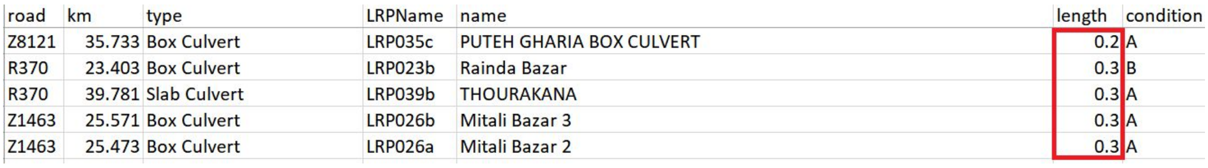
1.4.2 Inaccurate data

Data issues of the type semantic inaccuracy have been identified in the length, width and span data. In these data the values can be really large or small, example can be seen in figure 6, 7 and 8. It is not clear if this data issue occurs often, since it is difficult and time-consuming to find these issues. In figure 6 an example is given of this data issue. The data says that the specific bridge has a length of 1016.1 meter, but in reality this bridge has a length of around 220 meter[1]. In figure 7 and 8 other dat can be found that is probably inaccurate. A first way to solve this issue is to point out a maximum width, length and spans, which can be based on the largest bridge in Bangladesh. This way high or low outliers could be removed easily. To solve the less obvious issues, a view at the individual bridges should take place, which is relatively time consuming.



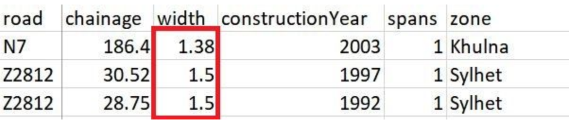
road	km	type	LRPName	name	length	condition
R820	1.015	PC Girder Bridge	LRP001a	BADU BAZAR PC GIDER BRIDGE	1016.1	A

Figure 6. data issue example: unrealistic long bridge



road	km	type	LRPName	name	length	condition
Z8121	35.733	Box Culvert	LRP035c	PUTEH GHARIA BOX CULVERT	0.2	A
R370	23.403	Box Culvert	LRP023b	Rainda Bazar	0.3	B
R370	39.781	Slab Culvert	LRP039b	THOURAKANA	0.3	A
Z1463	25.571	Box Culvert	LRP026b	Mitali Bazar 3	0.3	A
Z1463	25.473	Box Culvert	LRP026a	Mitali Bazar 2	0.3	A

Figure 7. data issue example: unrealistic short bridges



road	chainage	width	constructionYear	spans	zone
N7	186.4	1.38	2003	1	Khulna
Z2812	30.52	1.5	1997	1	Sylhet
Z2812	28.75	1.5	1992	1	Sylhet

Figure 8. data issue example: unrealistic small width

1.4.3 Semantic definition issue

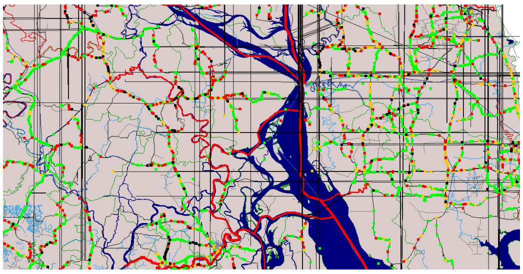
When the term spans is used in engineering, this mostly means the length of the beam between two supports of a bridge. However, in the data file it is used to represent the number of arcs on a bridge which can be misleading when it is not known what the column means. As none of the given data quality categories could be linked to this issue, it has been decided to create a new category for this data issue, which has to do with the semantic definition of a data type.

[1] Retrieved on Feb. 20 2019 from https://www.google.nl/maps/dir/23.7097932,90.4023439/23.7080532,90.4012398/@23.707854,90.4010066,17.21z/data=!4m2!4m1!3e2

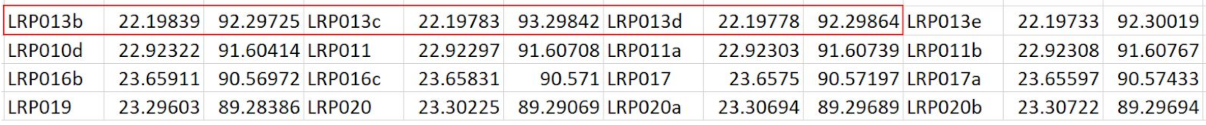
2. Roads

As the model makes use of the _roads.tcv file, this file has been used to discover and repair data issues. This file is used to draw the N-, R-, and Z-roads with the LRPs and the associated longitude and latitude. Multiple data issues have been identified,which can all be lead back to a problem with the coordinates of some of the LRP's:

2.1 Coordinates



The problem type is of semantic accuracy, the coordinates of the LRPs are not equivalent to the real-world data. In the example it is shown that there is a big jump between the LRPs. An example is a road that moves from the 92.297 to 93.298 in one jump. This is most probably an error and should be 92.298. These huge jumps are the reason behind the big horizontal and vertical jumps in the map shown to the right, creating a grid system look in the model. To clean these errors two approaches will be discussed next.



LRP013b	22.19839	92.29725	LRP013c	22.19783	93.29842	LRP013d	22.19778	92.29864	LRP013e	22.19733	92.30019
LRP010d	22.92322	91.60414	LRP011	22.92297	91.60708	LRP011a	22.92303	91.60739	LRP011b	22.92308	91.60767
LRP016b	23.65911	90.56972	LRP016c	23.65831	90.571	LRP017	23.6575	90.57197	LRP017a	23.65597	90.57433
LRP019	23.29603	89.28386	LRP020	23.30225	89.29069	LRP020a	23.30694	89.29689	LRP020b	23.30722	89.29694

Figure 9.

2.1.1 Interpolation data cleaning - "Road Corrections - Algorithm 1"

The first approach used to clean up the outliers within road segments was by using interpolation. A python script was made that checks whether any given LRP coordinate falls between its previous and following LRP points for both the longitude and the latitude. If this is the case then nothing happens, but when the middle point isn't between its neighbors it is treated as an outlier and then corrected. Graphically, the correction is made by drawing a line between the neighboring points and placing the outlier in the middle. This approach works well when the outliers come in singles. When there are multiple outliers in a row, then this approach only straightens it out a little bit, depending on the number of outliers in a row. A different issue arises when the road's LRPs end up zigzagging a lot. This would cause the algorithm to incorrectly identify these LRPs as outliers. This doesn't seem like a common occurrence as roads are unlikely to follow this pattern unless the terrain demands it.

Two potential improvements were identified for this script. The first improvement would analyze the road LRPs over more than three LRP points after which it identifies and replaces outliers based on that. This would bypass the consecutive outlier problem that the current script has. A second improvement would be to make the script more iterative. The current end result has only been iterated five times which means that outlier groups are put closer to their original position 5 times. If the script doesn't work just on the central LRP's location to its neighbours, then the script could be used to slowly pull consecutive outliers back to where they should be.

2.1.2 Distance data cleaning - "Road Corrections2 - Algorithm 2"

The second approach has been performed on the 'Roads_InfoAboutEachLRP.xlsx' file as it contains information for the cleaning of bridges. This algorithm could also be used for the '_roads.tcv' file if it is changed to the same file structure or '_roads.tcv' could be adjusted based on this data. Meaning the dataframe that is now the end of the Road Corrections 2 file would be transformed into the format used in the tcv file. This will lead to a better roadmap in the java based map. The reason for fixing the 'Roads_InfoAboutEachLRP.xlsx' file is to be able to adjust the bridges positions according to the positions of the roads so consistency between the two structures can be achieved. The algorithm and explanation for this can be found in the Bridges section above.

The algorithm uses the absolute distances between latitudes and longitudes to correct for mistakes. The standard deviation of the distances is then used to create the maximum limit (1.96 * std.dev) between the latitude and longitude values of the next LRP. If the distance is bigger than the maximum limit the entire row will be removed. The decision of using 1.96 times the std.dev comes is based on a 95% confidence interval. In this decision the assumption has been made that the differences in distance between two point can be seen as a normal distributed variable.

Removal of the row that contains an outlier leads to loss in information, however an approximation of the correct latitude and longitude could not be completed in the timeframe. Potential solutions would be to use the moving average of the latitude and longitude to replace it or average between the coordinates. Furthermore, the chainage can be used to gauge the jm distances and see if the coordinates match the distances.

3. Conclusion and reflection

This report is a summary of preliminary work done on cleaning a dataset related to Bangladesh's infrastructure network. The datasets that were focused on contained summarized information on the roads and bridges in Bangladesh from numerous other files. A variety of errors were found, such as swapped latitudes and longitudes, missing values, duplicate entries with differing information or simple typos. These errors were attempted to be resolved using python scripts. Most of the scripts focused on fixing specific errors, whereas others used a more general approach to fix common issues.

Although numerous problems with the data were identified it is likely that others of them were missed completely. This is partially due to some issues being more obvious than others and occuring more often, but also due to a lack of time and perhaps even a tunnel vision. In a full scope project more effective troubleshooting is likely possible when it is fully clear how the data will be used. Not every problem in the data is relevant given a certain usage for the data. More time would also be important for a better and more thorough implementation of the used python scripts. Still the cleaning covered in this report made the first step in algorithmically cleaning the datasets, which means there is space for additional data without any significant additional cost in cleaning time.