Manchester Digital Twin
Creating an Environmental Evaluation Model for Urban Design

Callum Evans
27th April 2021

University of Exeter

Abstract

As the UN predicts the urban population will continue to grow despite the threat of climate change, sustainable development must become a fundamental part of reducing cities' emissions. This project, therefore, aims to demonstrate how digital twin technology could be applied to the reduction of greenhouse gas emissions from congestion. The final Manchester Digital Twin (MDT) system consists of a simulation of the traffic emissions levels in central Manchester, intending to help experts and non-experts alike gain insight into complex urban systems through the visualisation of data. A system such as this should help promote a better understanding of the ways cities function, whilst simultaneously helping urban planners make more considered choices in future developments. This report details the project's preliminary research into digital twins and traffic modelling as well as the development process of creating the final system. Specifically, this includes the network building and representation processes, the calculation of emissions with COPERT and the system's deployment as a web application. Research into OD-matrix estimation and the traffic assignment problem is also described, as previously planned features of the MDT. Evaluation of the final system through user feedback ultimately found the MDT performs well as a digital twin, despite the simulation itself only being able to accurately demonstrate some basic relationships.

I certify that all material in this dissertation that is not my own work has been identified.

Contents

1 Introduction

1.1 Project Overview

Cities play a fundamental role in the way our society functions and greatly impact the environment around them. For example, The United Nations (2018) estimates there will be 43 megacities of over 10 million people by 2030, and the world's total urban population is predicted to grow to 68% in the next 30 years. This urbanisation promises to increase productivity and efficiency as noted by X. Q. Zhang (2016), however, whilst acting as economic and technological centres, urban sprawl and poorly designed cities are also huge drains on energy and resources and contribute to higher levels of congestion and pollution. 70% of global CO_2 emissions are currently produced by cities according to C40 (2021), and as this urban population continues to grow, ensuring our future urban development is sustainable will become crucial in fighting climate change, especially alongside current emissions targets.

A large factor in the current impact of cities is transportation as, for example, a report by the European Environment Agency (2020) found the transport industry alone accounted for 27% of the EU's greenhouse gas emissions in 2017, with the majority of this being produced by road vehicles. However, being able to model and predict congestion in cities could dramatically reduce the impact of transportation on the environment. For example, a report by the California Air Resources Board Office of Strategic Planning (1989) found that vehicle hydrocarbon emissions increased from 1g per mile to 7g in congested traffic of 20mph when compared to a constant speed of 55mph. This is because, as noted by K. Zhang and Batterman (2013), the slower 'stop-and-start' driving conditions as seen in congested traffic causes higher emissions through increased braking, idling and acceleration. The need to predict and prevent congestion has ultimately lead to many traffic modelling methods being developed since the first was proposed by James Lighthill and Beresford Whitham in 1955 and are now being used more in larger-scale urban planning, such as by Castro, D. Zhang and S. Li (2012). This application of traffic modelling to predict congestion and plan accordingly could greatly impact the sustainability of our urban environments by allowing for considered choices in city design.

Alongside this increase in the prevalence of traffic modelling, digital twins, which are virtual representations of real-world systems, have seen an even greater rise in recent decades. They can allow for very complex data to be condensed and understood through visualisation and can be applied to an incredibly large range of applications. Amongst these, urban digital twins specifically have become much more popular as the amount of data surrounding our cities has dramatically increased. Their development comes with many challenges, for example, those described by A. Rasheed, O. San and T. Kvamsdal (2020), such as their computational cost, the data management necessary or the possible need to handle real-time communication with the physical system. However, their ability to help people gain understanding and insight into complex systems is a perfect solution to a problem such as congestion and climate change. Urban digital twins, such as the system developed by Dembski et al. (2020) of Herrenberg in Germany, highlight how they are able to encourage interaction and understanding even in people with no experience in urban science, whilst still acting as an excellent tool for experts and planners.

Considering the issue of climate change and congestion in cities along with the rise of digital twins, the ultimate goal of this project is to demonstrate how this technology could be applied to urban and environmental sciences. This will be done through the creation of an urban digital twin of Deansgate and Piccadilly in central Manchester, one of the most congested areas of the city. Manchester has also recently seen a huge rise in investment and development, even despite the COVID-19 pandemic as noted by Robson (2021), and so should make a perfect example. The twin itself, however, will focus on the relationship between congestion and emissions and will try to follow the principles of digital twins by using visualisation to help people gain a better understanding of these complex urban systems. The user should also be able to interact with and investigate these relationships by allowing them to see the impact of changes to the vehicle makeup and emissions calculations. Accessibility should also be an important aspect of the project by creating the twin as a publicly available web application, ensuring the system is as approachable as possible. Ultimately, giving tools that allow experts and non-experts alike to analyse and understand urban systems should allow the Manchester Digital Twin (MDT) to act as a useful evaluation tool for city design. This should then help to promote more considered and sustainable development alongside more interaction with our cities in an increasingly urban world.

1.2 Literature Review Summary

1.2.1 Urban Digital Twins

Digital twins were originally proposed by Michael Grieves at the University of Michigan in 2002, and were defined by Bolton et al. (2018) as "a dynamic virtual representation of a physical object or system across its lifecycle, using real-time data to enable understanding, learning and reasoning." The technology has been used in a wide variety of applications and can be incredibly efficient at helping to condense and visualise complex information. For example, Tuegel et al. (2011) used a digital twin with the US Air Force to evaluate the structural integrity of aircraft designs, or more recently, Corral-Acero et al. (2020) found digital twins could be very influential in diagnosing cardiovascular diseases through digital heart models.

Alongside these advancements, digital twins have also become more popular in the field of urban science as the amount of data surrounding our cities has increased dramatically in the past few decades. As noted by Kaur, Mishra and Maheshwari (2020), the rise of the Internet of Things and smart cities has lead to the ability to analyse and model urban environments in ways that were not previously possible. This is reflected in the prediction by ABI Research (2019) that there could be in excess of 500 urban digital twins deployed around the world by 2025, up from only a handful 2 years ago. Multiple cities have already created their own digital twins, such as the 'Virtual Singapore' project developed by the Singaporean National Research Foundation (2021), which uses 3D modelling and extensive data for research and planning throughout the city. Cityzenith (2021) is also helping to create the world's first city built in its entirety with the use of a digital twin with Amaravati, the new Indian state capital of Andhra Pradesh. On a much smaller scale, Dembski et al. (2020) developed a twin of the German town of Herrenberg for use in analysing, integrating and visualising data in urban planning. Their goal, which was ultimately successful, was to try and encourage citizen interaction with the urban environment around them, and to help even non-experts gain insight into how it functions.

This concept of helping experts and non-experts alike gain a better understanding of complex systems is incredibly important to all types of digital twins and does this primarily by visualising data. It is this visualisation that is able to highlight patterns or problems in real-world systems where static data and calculations fail, such as bottlenecks at road junctions or issues in manufacturing chains. Therefore, due to the complex and data-rich problems surrounding our cities and the environment, this application of digital twin technology to urban science and climate change is a perfect extension to the technology.

1.2.2 Traffic Flow Modelling

The first research into traffic flow modelling was done by Greenshields et al. in 1934, who studied the relationship between vehicle velocity and spacing and also introduced the concept of traffic flow and density. However, in terms of modern traffic flow modelling methods, there are three main approaches; microscopic, which simulates traffic through individual vehicles, macroscopic, which focuses on relationships throughout traffic such as flow or density, and mesoscopic that aims to combine elements of the two.

Whilst mesoscopic methods are not used very often, due to their complexity and the limited number of possible applications, the most common of the three is microscopic flow modelling. As noted by Bazghandi (2012), they have the advantage of being able to better represent emergent behaviour because they model individuals instead of relationships. This means that they can generally act as a more natural, and therefore realistic, simulation of real-world traffic. However, this does come with the limitation that this approach can quickly become very complex and computationally expensive, especially as the number of agents increases. Human drivers themselves can simply become difficult to predict and model, as behaviour can be irrational and vary dramatically with time and place.

On the other hand, macroscopic modelling can be much less complex and expensive due to its comparative simplicity. The earliest research into this type of modelling was done by James Lighthill and Beresford Whitham (1955), who proposed simulating traffic similarly to supersonic projectiles or flood movements in rivers, and with the addition of work by Richards (1956), created the Lighthill-Whitham and Richards model. Despite its original popularity, as noted by Chanut (2005), the model represents traffic as a continuous and homogeneous flow in an equilibrium state, effectively meaning that traffic consists of identical vehicles with infinite acceleration, all immediately moving at the same velocity with any change in equilibrium. In the decades since the Lighthill-Whitham and Richards model was first created, many different improvements have been proposed that aim to fix its homogeneity, such as the multi-lane, multi-class model by Daganzo (2002) that adds driver classes with their own individual behaviour.

1.2.3 Emissions Modelling

There are many models for calculating vehicle emissions, however, as noted by Knez and Baškovič (2013), they can broadly be split into three classifications; emission factor, average speed and modal emission models. Emission factor models use simple calculations without the need for large amounts of data, relying solely on vehicle type and a specific driving mode, which limits their use to regional or national emissions due to their inaccuracy on smaller scales. Average speed models base their calculations on speed-related emission functions and are often used in models on a road network scale. Lastly, modal emission models are the most complex, but also the most accurate, and calculate emissions as a function of speed as well as operational modes such as acceleration, deceleration etc. These can be specific enough to provide real-time values for a particular type of vehicle, but according to Esteves-Booth et al. (2002), can also be highly complex due to the amount of data needed.

The main model considered in this project is COPERT, which is the industry standard within the EU for emissions modelling. As seen in the technical report by Ntziachristos and Samaras (2000), it uses vehicle speed and type, along with other factors such as ambient temperature or road grade to calculate emissions. It can model major air pollutants such as CO, NH_3 or SO_2, as well as greenhouse gas emissions of CO_2, N_2O and CH_4, and has the benefit of being heavily parameterisable. This is all done without requiring a large amount of data like some modal emission models, meaning it can be applied to a wide variety of different scenarios. However, although the COPERT model methodology is peer-reviewed and validated by the United Nations Economic Commission for Europe, it has been accused of underestimating emissions for certain pollutants, particularly at lower speeds, as seen in the experiments done by Khreis and Tate (2017). Even though it is still suitable enough for European applications, this was especially the case in Australia where, according to Smit et al. (2017), COPERT underestimated values by between 7-37% leading to a separate COPERT Australia being developed.

1.2.4 Original Specification

The original specification for this project stated that the system should consist of an installable program that allows users to run a simulation of traffic in central Manchester, whilst allowing for small modifications to the network, such as closing roads or changing speed limits. The network would be made from OpenStreetMap data, whilst traffic flow would be calculated using a combination of real-world data and the Daganzo (2002) traffic flow modelling. This would ensure a more accurate macroscopic and homogeneous flow could be used, with average speed values for all driver classifications that could be used in the emissions calculations, which would be done with COPERT.

The final system would then primarily be evaluated by comparing the final emissions and flow rate values to real-world data and trends, which would be street-level emissions measurements and flow data respectively. Aside from this, the system can also be evaluated on how well it is able to help experts and non-experts alike gain insight into how urban systems function through user feedback.

2 Project Specification

A majority of the goals specified in the original specification remain the same, particularly including the primary goal of creating a digital twin for use as an environmental evaluation tool for city design. The network is still constructed from OpenStreetMap data, and emissions are still being calculated using COPERT. However, 2 relatively major changes were made to the original specification:

- The system will be deployed as a web application instead of an installable program. This was ultimately a good decision as deploying the system publicly online made it much more accessible, more closely following the principles of digital twins. This also allowed for much more flexibility in the design of the user interace and helped to ensure that the system was able to effectively communicate the data.
- The user will no longer be able to make small modifications to the network layout as this was found to be much too complex and computationally expensive for this application. Nevertheless, research was still carried out alongside development into possible implementations of this feature, with the two main methods detailed in section 3.6.

3 Design & Development

3.1 Design Overview

The design & development of the project was primarily separated into stages focusing on 3 fundamental elements; networks, emissions calculations and the web application. Firstly, the networks stage mainly consisted of creating a way for networks to be built from the available datasets and rendered in folium. The following emissions calculations stage focused on implementing the final calculations using parameters from real-world datasets and adding the ability to change these parameters. The final web app stage then focused on creating a user interface in Django and deploying the final system onto a web server. As briefly mentioned, alongside the first two stages, research was also carried out into methods for allowing the user to modify the road network. However, this was ultimately deemed to not be feasible for use in the final system, as described in section 3.6.

3.2 Data Sources

Choosing datasets plays a significantly important role in the development process of a digital twin as they determine how the network is represented and how accurate the final simulation will be to the real-world system. In the case of the MDT, the data used in this project comes from three sources:

- **OpenStreetMap (OSM)**: This is one of the most widely used open-source mapping datasets, and is used as the base map for the simulation, including all network geometry. When the network is being built, the OSM map data is queried with 'Overpass API' and its Python wrapper 'overpy.'
- **TOMTOM Traffic Index**: TOMTOM provides historic and real-time traffic data, collected from 600 million drivers across 57 countries, including flow rates, average speeds and Origin-Destination matrices. The historic data from April 2019 could be used for free and was downloaded in a JSON format. Each road has its own set of flow data values, divided into hour-long

segments, which includes average speed, median speed and sample size, i.e. the number of recorded vehicles. As these values represent one hour in the day, the sample size can be taken as the flow rate or '*vehicles per hour.*'

- **UK Department for Transport (UKDT)**: The UKDT supplies historic traffic count data from points across the UK including the proportions of different vehicle types. The dataset has over 3700 count points in Greater Manchester, which can be used in the emissions calculations.

3.3 Networks

3.3.1 Network Representation & Construction

All networks used in the project, the OSM, TOMTOM and final MDT networks, consist of a standard graph representation of nodes and segments. A *Network* object is created for each with a dictionary of segments and nodes, enabling fast lookup when querying elements with their key. The only addition to the datasets for this representation was that node keys had to be generated for the TOMTOM network, which is done by multiplying their longitude and latitude coordinate values together and removing the decimal point. This is because the original network format only uses segments, and defines points solely within segment geometry coordinates.

The main issue that arose when constructing the networks was the execution time, specifically when building the OSM network. The TOMTOM network can be built in seconds, as this data has been downloaded in a JSON format, which is very quickly processed by the system. However, as previously mentioned the OSM data is queried using the 'Overpass API,' and although this can finish in a reasonable time for smaller networks, this is not the case for the MDT with around 1150 segments and 4000 nodes. The main query used to fetch all nodes and segments for the MDT can be seen in appendix A and usually takes around 6 minutes. However, this is usually much longer as the server can time out or become overloaded, especially as the number of segments being fetched increases. This is partially because, once all the segments and nodes are fetched from the dataset, each of these still has to be queried again in order to fetch attributes or coordinates.

Evidently, this means that the MDT network cannot be constructed from scratch each time a user loads the page, so the solution to this is that all networks are built once and stored on the server in object files. A *NetworkCreator* class has been created that can query the 'Overpass API' to build the initial OSM network and build the TOMTOM network from the JSON file. It is then able to merge these to create the final MDT network, as detailed below.

3.3.2 Merging Networks

Once the separate networks representing the TOMTOM and OSM data have been built, a new MDT network can then be created by combining the two. This is done whilst maintaining the network geometry and characteristics from OSM, such as a segment's speed limit, road type and the number of lanes, whilst also adding TOMTOM's flow data to the corresponding segment. The OSM data is chosen as the base for the MDT network as the 'Overpass API' allows for very specific queries to be made to the database, as well as the fact that the resulting network will ultimately be displayed over an OSM map.

However, in trying to merge the TOMTOM and OSM datasets, the differences between the two become very clear. The network geometry itself is slightly distorted when comparing the two, and TOMTOM stores a large number of segments that are not needed for the simulation. As can be seen in appendix B, the TOMTOM data includes car parks and pedestrianised areas that should not be in the final MDT network, and road types cannot be filtered as they can with OSM's API. Despite this, the larger issue with combining the two network comes from the slight, random difference between almost all coordinates in the datasets. This becomes more problematic at points with multiple parallel road segments or complex junctions where lanes are separated as seen in appendix C. The TOMTOM network also uses much smaller segments, so in order to make the merging algorithm easier to perform, the network is simplified by removing degree 2 nodes.

Therefore, I created algorithm 1 in order to solve this problem, which works by first creating a K-D tree for the TOMTOM network's nodes. This is a binary tree representation of all network coordinates that can be very quickly queried to find the closest K neighbours in D dimensional data. The algorithm can then iterate through all points in an OSM segment's geometry, finding the 3 closest points in the TOMTOM network with the K-D tree. For each of these TOMTOM points, all their attached segments are collected and added to a *closest segment* list. Once this has been done for all points in the OSM segment, the most common TOMTOM segment in the *closest segment* list is then matched with the OSM segment. The algorithm also uses the constraint that both segments must have the same street name, and initially, the TOMTOM segment flow data must have a sample size above 0, meaning vehicles were observed on the segment, and there is flow rate data attached. This means the algorithm favours segments that have valid non-zero data attached to them whenever this is possible.

Algorithm 1 does assume that corresponding segments in the two networks are of similar length and shape, essentially indicating they run parallel. The simplification of the TOMTOM network does mean that this holds more true than using the raw data, but there are still OSM segments that are shorter than their TOMTOM counterpart. The problem that the correct segment could effectively 'get lost' if the two begin to diverge could arise, but the constraint of segments needing the same name ensures this does not happen.

Once this algorithm has finished executing, the resulting network can be saved to the server in an object file, and later read by the map renderer.

Algorithm 1 Network Merging Algorithm

```
for OSM segment in network do
  closest segments ← []
  for point in OSM segment do
    closest points ← 3 closest TOMTOM points
    for TOMTOM point in closest points do
      attached ← segments attached to TOMTOM point
      Append attached to closest segments
    end for
  end for
  Sort closest segments by frequency
  for TOMTOM segment in closest segments do
    if TOMTOM segment street name equals OSM segment street name then
      if TOMTOM segment data sample size > 0 then
        Match OSM segment with TOMTOM segment flow data
        Continue to next OSM segment
      end if
    end if
  end for
  Add Empty Flow Data to OSM segment
end for
```

3.3.3 Network Rendering

The final rendering for the user interface is done using the Python package 'folium,' which allows for data to be manipulated in Python before being visualised on a Leaflet map. It has the benefit of being easily customisable, in terms of map styles and overlays, and the final map can also be saved in a HTML format and easily integrated into a web interface. However, one of the limitations of folium is that it is mainly designed to show static data that does not change regularly, so there are not very many available interactions. This is not impossible to overcome, as I was able to, but folium was still found to be the best solution regardless. I found other options such as 'ipyleaflet,' which has inbuilt event handling for interaction from the user, were very difficult to use due to a lack of documentation or applicable examples.

The customisability of folium is certainly one of its largest benefits, with a large variety of free map styles available on GitHub, and the ease of adding custom markers and polylines, the last of which is fundamental in displaying this kind of data. Road segments can be drawn with polylines using defined coordinates, width and colour as well as tooltips and popups that were incredibly useful in finding an alternative to an event-based user interface, as detailed in section 3.5.2. Also, both the emissions and flow rate maps are able to use different labelless map styles from CartoDB, a GIS and mapping company, which do well in differentiating the two metrics.

Ultimately, a *FoliumMap* object was created for use with the *Network* object that can render the network on a folium map according to a specified metric with much greater flexibility. This approach using these objects dramatically reduces the time needed to draw complex road networks when compared with querying the datasets with each execution. As can be seen in appendix D, whilst Deansgate and Piccadilly only have around 1150 segments, networks containing upwards of 10,000 segments can easily be rendered in a fairly reasonable time, with the only limitation being the time needed to create the initial object files. There is still some overhead associated with folium rendering the final network that cannot be overcome as each segment has to be drawn 13 times, once for each hour of the simulation, and 1 for an outline. However, as detailed in section 3.5.2 and 3.5.3, there have been steps taken to reduce its impact.

3.4 Calculating Emissions

3.4.1 Applying the COPERT Model

As mentioned previously in section 1.2.3, COPERT is an industry standard within the EU and was therefore chosen as the best method of calculating emissions. Along with this, there is a publicly available Python implementation of COPERT developed by Chen and Mallet (2016) called 'Pollemission,' which provides all necessary calculations for each vehicle type and emissions standard. There are currently 6 classifications, from the most polluting Euro 1 to the least polluting Euro 6, with all of them being used to define the limits for exhaust emissions of new vehicles sold in the European Union and EEA. Pre-Euro can also be used to refer to vehicles sold before the standards were introduced in 1992. To ensure that the final emissions values are calculated correctly, the classification distributions for each vehicle type are applied to the final calculations.

In the MDT, emissions are calculated over 5 vehicle types; two-wheeled vehicles, passenger vehicles and taxis, buses and coaches, large goods vehicles (LGVs) and lastly heavy goods vehicles (HGVs). The proportions of these for each segment are found from the UKDT count points data mentioned previously and can be queried using a K-D tree. The distributions of European emissions standards for the different vehicle types were found from a variety of sources, for example, the distribution for cars was found from a report by Morton (2017) looking at cars registered in Scotland's main cities. Then, the distribution for LGVs and HGVs are projected values for 2020, from a Y. Li, Pearson and Murrells (2009) report for use in the UK's national transport model, with equal distribution between petrol and diesel engines. Lastly, the distribution used for buses and coaches was found slightly differently, being taken from the Stagecoach Manchester annual report for 2019-2020. Stagecoach is the company that provides the majority of bus services within Greater Manchester and, according to their report, have a total of 791 buses. This includes 32 electric buses, 173 in Euro 6 and 138 in Euro 4 & 5, with the rest assumed to be Euro 3 for this project. Also for the purposes of this project, the 138 Euro 4 & 5 buses have to be represented by the same number of Euro 4 buses. The final distributions used in the calculations can be seen in table 1, except for two-wheeled vehicles as emissions can only be calculated for Euro 3.

Vehicle Type	European Emissions Standard						
	Pre-Euro	Euro 1	Euro 2	Euro 3	Euro 4	Euro 5	Euro 6
Petrol Engine cars	0	0.002	0.017	0.200	0.374	0.371	0.034
Diesel Engine cars	0.013	0.008	0.057	0.292	0.324	0.269	0.037
Buses and Coaches	0	0	0	0.566	0.174	0	0.219
Large Goods Vehicles (LGVs)	0	0	0.002	0.019	0.150	0.313	0.516
Heavy Goods Vehicles (HGVs)	0	0	0	0.008	0.027	0.203	0.761

Table 1: The final distributions used for each European emissions standards classification, separated by vehicle type. These are used as base values before any modifications are made.

3.4.2 Calculation Modifications

The user is also able to make changes to the way emissions are calculated through a network editor panel in the interface. This includes a set of multipliers for each of the 5 vehicle types, a slider for the percent of petrol vs diesel engines and an input for the temperature. These were chosen as modifiers as not only are they easy to understand, they can also have a very clear impact on the amount of emissions without affecting other parts of the simulation, specifically the flow rate. Whilst changing speeds or capacities of roads throughout the network would be a good addition, this would have too much of an effect on the flow rate and would require something similar to the traffic assignment problem. Considering this, the vehicle type multipliers only changes the proportion of each type, ensuring that the amount of vehicles throughout the network remains the same. This is also done in relation to the vehicle type distribution of each segment that comes from the UKDT dataset to remain more true to life. Each type proportion is multiplied by the user's input, ranging from 0 to 10, and new distributions are calculated as a percent of the sum of the new values.

The effect of these modifications can be seen in figure 1, which shows how it is easy to understand how each change affects the network when the results are compared to the original network in figure 1 (a). The results of these modifications are also reflected in the graphs and data in the user interface as well as the values that are downloaded when the network is exported. This should allow users who want to analyse emissions results more closely to do so.

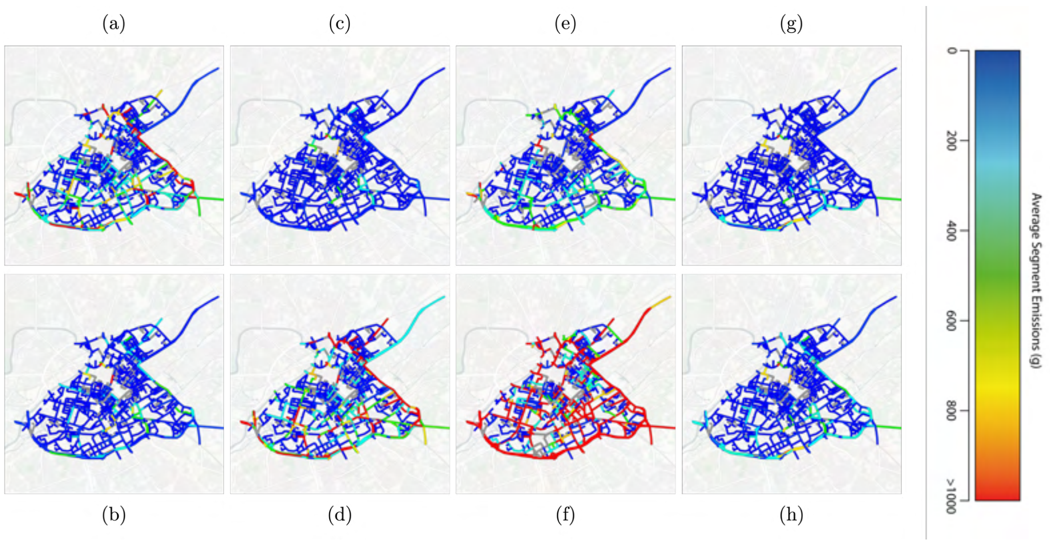


Figure 1: Resulting network emissions with (a) no modifications. (b) no HGVs, more cars, increased diesel. (c) no petrol engines. (d) no diesel engines. (e) only HGVs. (f) only mopeds & motorcycles. (g) ambient temperature of 40℃. (h) ambient temperature of -20℃. The legend to the right shows the emissions values.

3.5 Web Application Development

A web app was chosen as the best implementation of the project as, in aiming to stay true to the ultimate goal of a digital twin, this ensures the project is as accessible as possible. It was then decided that Django, a popular open-source and Python-based web framework, would be used due to its ease of use and wide range of functionalities.

Django handles requests and generates web pages through 'view functions,' mostly called 'views,' by first searching through a list of available URLs and directing the request to a view function if a match is found. It is this function that is then able to process the necessary data and return a web page along with a 'context' dictionary storing page elements. These pages are designed using HTML templates that can format the context values with the use of template tags that effectively allow for Python code to be executed within HTML. Django's template language does also provide a set of built-in tags which allow for basic logic such as for loops or if statements, but custom tags can be made for more complex tasks. For example, the inspector panel as detailed below extensively uses custom template tags to dynamically generate HTML elements depending on the segment, such as the graphs and tables. This all allows for a great deal of flexibility in the way that web pages are rendered and helps towards the goal of creating a more general tool that can easily be customised and changed to fit different purposes.

3.5.1 Web pages

There were 4 main templates created for use in the MDT; the index or landing page, the map page which can either display flow rate or emissions values, an inspector page as detailed in the user interface section and a create network page only accessible by administrators. When the user goes to the website, they are first directed to the landing page as seen in appendix E, which first only shows the green portion with the title, logo and links to the map views taking up the whole window. There is also a small downwards arrow that disappears once the user scrolls, where there is then a section giving users more context to the project by giving a very brief explanation on digital twins and talking about the project's motivation.

The same design is used for both the emissions and flow rate maps and relies on Django template tags and parameters in the FoliumMap class to change how the web page is rendered. This is done to ensure they remain consistent, whilst still looking visually distinct enough to be able to differentiate the two, mainly through the colour schemes. This is done by having the flow and emissions URLs direct to their own view function, which process and prioritises the most relevant data to the specified metric, which is then returned to the same '*map.php*' template.

The page for creating networks is much simpler as it is hidden to regular users and only shows a form with checkboxes for 'OSM,' 'TOMTOM' and 'MDT.' These simply create the checked networks in this order whilst displaying its progress on the console if required, which can be found by connecting to the server with SSH. Django does provide a user and authentication system, which is not used elsewhere in the MDT project, but it is used here to deny access to unauthorised users, so they are not able to create new networks. Administrators or superusers can log in through an admin login page, and although this project does not need to use the admin page itself, logging in does give the user access to the create networks page. Links can be found at the bottom of the footer on the landing page, and at the bottom of the MDT instructions window in the map view.

3.5.2 User Interface

Once a user has chosen one of the map views, the two main aspects of the user interface available to them are the sidebar and the inspector. First, the sidebar shows by default and includes the network editor options, the graphs displaying hourly emissions or flow depending on the view and the raw hourly data and averages. The graphs are drawn using Plotly during the initial page creation, whilst the table data is simply formatted into arrays that Django can easily print. Here, the system follows the goal of being a more general tool by aiming to be very easily expandable and customisable. The context dictionary contains sets of arrays for both graphs and datasets, which can be easily iterated through by Django as long as they are formatted correctly. These arrays can then be added to or shortened depending on the system requirements. It is also worth noting that each panel in the sidebar can also be shown or hidden by clicking the arrows next to the panel name, which aims to simplify the interface by not constantly showing unwanted information.

An example of the sidebar in the emissions map view can be seen in figure 2, showing the network editor. The emissions calculation modifiers as detailed in section 3.4.2 are the first elements within the network editor, however, there are also some options for displaying the map itself. These are the value upper and lower boundaries, and a 'draw zero values' checkbox, which can be used to help users analyse the network or just to improve performance. The draw zero values checkbox denotes whether or not segments with no observed vehicles throughout the day are drawn on the map by checking their average value. Using their average value ensures that segments with at least one non-zero value at a point in the day can still be analysed, but segments are not drawn if no information can be gained from them in any case. The value boundaries limit the segments drawn by ignoring those with average values outside the bounds, which are calculated as a percent of the highest average emissions throughout the network. For example, if the highest average emissions value is 1000g/hour and the boundaries are set at 25% and 50%, only segments with average emissions values between 250 and 500g/hour are drawn. Isolating the best or worst parts of the network should help users analyse and understand the data by making it easier to isolate parts of the data. It also helps reduce loading times when drawing the network, as it can dramatically limit the number of segments drawn.

Figure 2: An example of the sidebar layout & design.

The other two functionalities within the sidebar are for finding the worst segments in terms of average speed performance index (SPI) and emissions, and the ability to export the current network. The worst emitter is simply the segment with the highest average emissions throughout the day, but the SPI is calculated for each segment to find the most congested. SPI is an evaluation indicator of urban traffic state, defined by He et al. (2016) as the ratio of the observed to maximum road speed, effectively showing how much slower the traffic is compared to the speed limit. If traffic is moving at a speed 0-25% of the speed limit, this is considered heavy congestion, whilst 75-100% is considered to be very smooth with a good road traffic state. The only constraints placed on finding the lowest SPI in the MDT network is that the sample size has to be above 100, and there has to be a known speed limit. This ensures that roads with very few vehicles observed over the course of a day are not included as they are unlikely to be the most congested. Lastly, using the export network feature will download a zip file containing CSV files for its segments and nodes, as well as the flow and emissions data. This should allow anyone to analyse their results in more depth than the interface allows, or potentially even rebuild the network.

The inspector is an overlayed panel, that once a segment has been selected by the user, displays its basic characteristics as well as its hourly flow rate, emissions and vehicle type distribution. This works by using folium's popups, which allow for basic HTML, and a button that calls a JavaScript function. This *inspect* function takes in a segment's ID and sends this in a URL to a separate inspector view that can process any modifiers and format the segment data. This was found to be the only method for this because firstly, as previously mentioned, folium is designed for static data and has no callback or event capabilities. This meant that calling a JavaScript function from inside the map's iframe was the best method for modifying the page's Document Object Model (DOM). Secondly, the only way to query the MDT network is through Django, which has to be run on the server-side, as opposed to JavaScript which is run on the client-side. This can easily be done through a separate inspector view that can load, query and process the network data, which is then loaded into the inspector's div on the map page using jQuery.

An example of the actual contents of the inspector can be seen in figure 3, showing the layout of the data. The inspector first includes a minimap with a marker for the segment's location and uses a different map style to the network map. This uses a map by 'JawgMaps,' a company that provides maps and geocoding for websites and mobile apps, and was chosen as it gives more context to the road by displaying some surrounding street names and landmarks. The segment's attributes are then formatted for the table below the minimap, which mainly includes forcing the correct case, making the data more readable by rounding decimals or adding units etc. The flow rate and emissions graphs are created identically to those for the network averages, and the vehicle distribution is taken from the attributes dictionary and drawn in a pie chart. Lastly, the segment specific raw data values can be viewed at the bottom of the panel.

Figure 3: An example inspector panel for 'Cobourg Street,' a small residential street close to Piccadilly station.

There is also a navigation bar on the map and create network views, which allow for the user to navigate to either the map view or back to the landing page. The link at the end overlays a window containing instructions on how to use the MDT system, mainly giving information on how the modifiers affect the network and how to use the inspect function.

3.5.3 Loading Screen

As mentioned in section 3.3.3, there is still some overhead time that comes from trying to render the final map due to the large number of segments. The issue with this render time, other than the fact there is no real way to reduce it, is the point at which the rendering is executed. When the user goes to the URL for either map view, the *FoliumMap* class immediately starts to draw the segments, which does not take that long to complete. The map can then be rendered, however, this has to be done before the view function returns the template and context, meaning that the web page appears broken. It seems this way as the current URL does not change until the rendering has completed, meaning the current window freezes whilst the next page loads with no indication of progress. An example of this can be seen by going directly to one of the map views without going through one of the links on the landing page.

To try and reduce the impact of this and assure the user that the system or link is functioning correctly, a loading screen div has been created in both the landing and map templates. For example, if the user is on the landing screen and clicks on the link to the flow map, a div as seen in appendix F fades in with a loading bar. This, which is the first and longest loading screen, then waits until the folium map is rendered and the view function returns the map template. Once the user is redirected to the map template, there has to be another loading screen because the map takes a few seconds to open the final HTML file, however this never more than a few seconds. A template tag is then used at the end of the page that calls a JavaScript function once the map can be viewed, removing the final loading screen.

Although the need for a loading screen that takes this amount of time is not ideal, I do feel this is a sufficient enough compromise when there is no feasible way to reduce the render time itself. Without any indication of progress, the site would certainly seem like it is not working correctly, and the wait would most likely cause users to leave before the page is able to load. However, as mentioned in section 3.5.2, the user can reduce the number of segments drawn which does reduce the loading times. This should be a useful feature when trying to analyse the data, especially when repeatedly reloading the page to make small modifications.

3.5.4 Web App Deployment

Deploying the final system onto a web server was the last stage of the development process, and mainly consisted of configuring the virtual private server (VPS) and web server. This meant that a VPS first had to be rented from 'webdock.io,' which was primarily chosen for its cost but also its scalability and performance. Then, Django's web interface could be set up to allow for communication with the web server, and to allow for Python code to be executed. Currently, 2 interfaces are supported by Django, which are WSGI and ASGI, with WSGI being the main synchronous standard for Python-web server interfacing and ASGI being its newer, asynchronous equivalent. However, the primary deployment platform used by Django is WSGI and this project does not require asynchronous code, so the MDT is deployed using WSGI.

Therefore, the deployment is done firstly through 'Gunicorn,' a Python WSGI HTTP server for UNIX, which can directly run Django and communicate to the client through a proxy server. The proxy server in turn is configured with 'Nginx,' as recommended by Gunicorn, and directly handles client requests. The last stage of this process was then to link 'Nginx' to a public domain, *http://manchester-dt.co.uk*, aiming to make it more accessible to the public. The only change made to the system itself to be able to deploy the application was to limit the length of the simulation displayed on the map to 4 hours. This, unfortunately, had to be done as Nginx returns a 502 error when trying to load the whole simulation as it will always timeout before it is finished.

3.6 Network Manipulation

As mentioned in section 1.2.4, the original plan for the project was to allow the user to modify the road network, which would have been through small changes such as adding or removing lanes, changing speed limits or closing roads entirely. This was planned with the goal of helping people gain an understanding of how urban systems work by allowing them to see the impact of decisions that real-world planners have to make. However, compared to the modifiers that are available in the final system, these changes would require flow values to be calculated for segments in the network. Throughout the project's development, a few methods were considered for this process but were ultimately deemed to not be feasible.

3.6.1 Origin-Destination Matrix Estimation

Origin-destination (OD) matrices are used to represent travel demand between origin and destination nodes in a network and are an incredibly important aspect in transport planning and analysis, for example, with the design of rail services as proposed by Albrecht, Howlett and Coleman (2009). However, OD matrices themselves can be very difficult to produce, and so, these values must be estimated in many cases. This became problematic during development as I was not originally able to find an appropriate OD matrix dataset. This would have meant that when a user would make a change to the network, a new matrix would have to be estimated from vehicle count data and modified before flow values could be recalculated.

There are many proposed methods for OD matrix estimation but they ultimately aim to generate a matrix $g = g_i$, that consists of the demand for all OD-pairs $i \in I$. This demand then has to be assigned to links within the network with an assignment proportion matrix $P = p_{ai}$, with p_{ai} being the proportion of the demand g_i that uses link a. The assignment proportion matrix P could be set as a constant, however, this assumes that the assignment, which effectively represents the route choice, is independent of the load throughout the network and this is not realistic on congested networks. This is then solely applicable when there is only one practical route for each OD-pair, for example modelling between exits on a stretch of motorway. This is much easier as in this case, the shortest path between for the OD-pair is predetermined and independent of travel time, so demand is simpler to calculate from the observed flows.

However, in more realistic cases on congested networks, the complexity comes from the fact that an observed flow rate or vehicle count value on a segment could be produced by any matrix from an incredibly large set of possible matrices. This is of course exacerbated as the number of segments increases, especially with the 1150 links in the MDT network. The main methods researched for this project were in work by Peterson (2007) and Lundgren and Peterson (2008), who essentially frame the estimation as a gradient-based descent problem by aiming to minimise the function:

$$\min_{g} F(g) = \gamma_1 F_1(g, \tilde{g}) + \gamma_2 F_2(v(g), \tilde{v})$$
$$\text{s.t } g \in \Omega$$

Here, F_1 is a function that returns the distance between the estimated matrix g and a previously observed target matrix $\tilde{g}$, whilst F_2 does the same for the observed link flows $\tilde{v}$ and flows calculated by the implicit function $v(g)$. The variables γ_1 and γ_2 are also important, as these represent weights for how much confidence is placed in either the target matrix or calculated flow respectively. For example, the target matrix is simply reproduced if $\gamma_1 = 1$, but these weightings can be used by planners to update previously created matrices by adjusting confidence in their previous data. This target matrix also has a large impact on the final resulting matrix and is usually chosen to be the initial solution for the gradient descent. However, the weighting does mean that it is possible to estimate a matrix with a zero value matrix as the initial solution, as would originally have had to have been the case in this project.

The first step to estimate the matrix is to set an initial solution g^0, and let v^0 be the resulting flow values obtained from $v(g^0)$. The descent direction r for the current iteration then has to be found, such that the value of $F(g + r)$ is lower than $F(g)$, and this can be modified slightly to ensure that all link flow values are positive. The next step for each iteration is to then find the search limit, which is the largest possible step where all OD demands are feasible and better than the current solution. Then lastly, the step length is set, which is the maximum possible distance to move the current solution that fulfils this limit and is either the limit itself or 0.

However, as may be evident, a large problem with this method is that $F(g)$ is not an explicit function, as an equilibrium traffic assignment problem has to be solved with each iteration. Lundgren and Peterson specifically state that in their tests, over 95% of computational time is spent in solving traffic assignment, and although their paper found that this method does work for large networks, this would be too expensive for the MDT system to repeatedly process with each user modification. Another possible issue is in the computation of the search direction, as an approximate Jacobian matrix, $J = \{\frac{\delta v_a}{\delta g_i}\}$, has to be calculated for each OD-pair, again adding even more complexity.

Ultimately, all OD matrix estimation methods were found to be too complex and computationally expensive for the purposes of this project, especially when this process potentially has to be done regularly as someone is using the system. Regardless of this, implementing this type of problem would have been far out of the scope of the project and no suitable implementations could be found online.

3.6.2 Traffic Assignment

Traffic assignment, as briefly mentioned in the previous section, aims to determine traffic flow over a whole network given the travel demands, and has a range of applications past simulating networks. Once I was able to find a suitable OD matrix for central Manchester from another TOMTOM dataset, solely performing traffic assignment on an already available matrix would have been much more feasible. As can be seen in appendix G, reports could be requested from TOMTOM with custom-defined areas as the origin and destination nodes. This allowed for the simulation area to be separated into around 160 different nodes, including all entrance and exit points to the simulation, generally with positive results.

The traditional method for solving the traffic assignment problem is called the Four Stage Model (FSM), which was originally developed during the 1950s and 1960s, particularly with the work done by Mitchell and Rapkin (1954) on the relationship between traffic flow and land use. In the decades since, the FSM has become the standard for traffic flow calculation in macroscopic modelling, and although some may not always be applicable, the 4 stages as described by Peterson (2007) consist of:

1. **Trip Generation**: The number of trips originating and finishing at all zones in the network is generated.
2. **Trip Distribution**: OD demands are calculated for the number of trips between all zones to create an OD matrix.
3. **Modal Split**: The travel demand for each OD-pair is divided into separate travel modes.
4. **Traffic Assignment**: Estimated trips between OD-pairs are loaded onto the network under the assumption travellers want to minimise their travel time.

Evidently, the first two stages would not have been necessary for the MDT project, but generally involve the collection and processing of data. A popular method for generating the OD matrix, once trip data is collected, is the gravity model and was proposed by **jr˙law˙1955**. This model was inspired by Newton's law of gravitation and assumes that the travel demand in an OD-pair is directly proportional to the number of trips generated from the origin node. This increases the travel demand similar to how the demand between two cities grows proportionally with the size of the origin city and is analogous to the gravitational force between two objects increasing as one grows in mass. This has been found to be a reliable model for trip distribution, however, it does have some limitations, such as that noted by Peterson (2007) in the way the model struggles with a variation of demand over the course of a day.

The simplest case of the modal split stage could separate demand solely into private cars and public transport, however, this could be expanded to fulfil further requirements. Public transport could be split into different modes, such as walking, buses or trams, or the purpose of journey could be added to distinguish people travelling to or from work etc. Each of these transport modes then has to be assigned a utility function that reflects the decisions made by real passengers, such as those noted by Sjöstrand (2001) on the comfort, travel time etc. These utility functions are used to assign probabilities to each mode, denoting how likely someone is to use a certain travel mode to go from an origin to a destination.

The last stage of the FSM is the traffic assignment, where the demand is loaded onto the network to calculate its flow values, which is done using the network data generated in the previous steps. This can be done either deterministically, which assumes that all travellers take the shortest available path, but can also be done stochastically as proposed by Daganzo and Sheffi (1977), which adds probability to drivers' routes. Either way, both of these methods also introduce link cost functions for each road in the network, which are functions that use characteristics of the road, such as its length, speed limit or uncongested travel time, and returns the 'cost' of travelling along with the link. If the link cost functions of roads throughout the network are not independent of each other, roads are allowed to become congested once they are filled to capacity.

This whole process is then performed iteratively, with the final values obtained from the traffic assignment stage being used as feedback for the trip generation or distribution stages. The model should then be able to converge towards a suitable solution, often one using the minimal possible transportation cost for all OD-pairs. However, the issue with applying this type of model to the MDT system was then found to be the execution time needed, even with a pre-generated OD matrix. For example, Florian, Mahut and Tremblay (2005) performed traffic assignment on a relatively similar sized problem of Stockholm with 2100 links, and the model converged in 40 iterations. In many ways, the same problems of complexity and execution time were found in the traffic assignment models as in OD estimation models, and it is also worth noting that traffic assignment is a subproblem within the OD estimation procedure. Alongside this, and again similarly to OD estimation, implementing traffic assignment was deemed as out of the scope of this project, and no suitable implementations could be found elsewhere.

4 Testing & Evaluation

The system was mainly tested with bottom-up integration testing, specifically to ensure that all of the network, emissions calculations and web application elements functioned correctly together. This allowed for less time to wasted later in development, as it would then be clear that all lower-level modules in the system, such as the *Network* class that was used as the network representation, had no major errors. Unit tests were also performed throughout the entire development process to make certain that smaller sections of the code also returned the correct outputs. For example, a fair amount of time was spent early in development in ensuring the algorithm for merging the OpenStreetMap and TOMTOM datasets performed correctly.

Meanwhile, the evaluation of this project was divided into 2 separate parts, each focusing on one aspect of a digital twin. The first part, as detailed in section 4.1 evaluates the MDT as a simulation by comparing the calculated emissions against real-world values. As the flow rate is not calculated by the system and solely use the TOMTOM dataset, only the emissions calculations need to be evaluated in this way. Section 4.2 evaluates the MDT as a tool for helping people gain a better understanding of the way complex urban systems function and does this through a user feedback survey. As previously mentioned, this is a key aspect of digital twins, and so became an important part of the evaluation process.

This user feedback was also used as a form of user acceptance testing, mainly to ensure the final deployed system worked as a whole. Each participant was asked which part of the system they used, as well as if they had any technical issues occur during the testing, which could then be fixed. Ultimately, no major issues were found during this stage.

4.1 Emissions Modelling Evaluation

Although there were no suitable datasets online for street-level CO_2 emissions measurements, which would have been ideal for measuring the accuracy of the simulation, there is more general data available. Specifically, the Highways Forecasting and Analytical Services (HFAS) published a 2010 emissions inventory update for Greater Manchester which shows the various pollutant emissions broken down by road type, vehicle type and borough. It is also worth noting that it is assumed if the MDT is able to correctly model emissions without any modifications to the vehicle distributions, then the model can be taken as an accurate simulation with modifications. This should especially be the case as it is solely the general trends and relationships between measures being evaluated, not absolute values.

Speed is an important factor in measuring emissions levels, and so this relationship was an important aspect to evaluate within the MDT model. The graph on the right in figure 4 shows the average CO_2 emissions per vehicle for motorway and trunk segments in the MDT against the segment's average speed, whilst the graph to the right by Shahid et al. (2014) shows the expected values of speed against emissions per mile. As can be seen, the expected trend is followed relatively closely by the MDT in that there is a large peak in emissions at very low speeds and a smaller peak at much higher speeds. This becomes much clearer once disaggregated by road type as the vehicle types and other factors are removed. Overall, I believe this result is promising for the MDT's ability to show the basic relationship between vehicle speed and emissions when the network disaggregated into different road types.

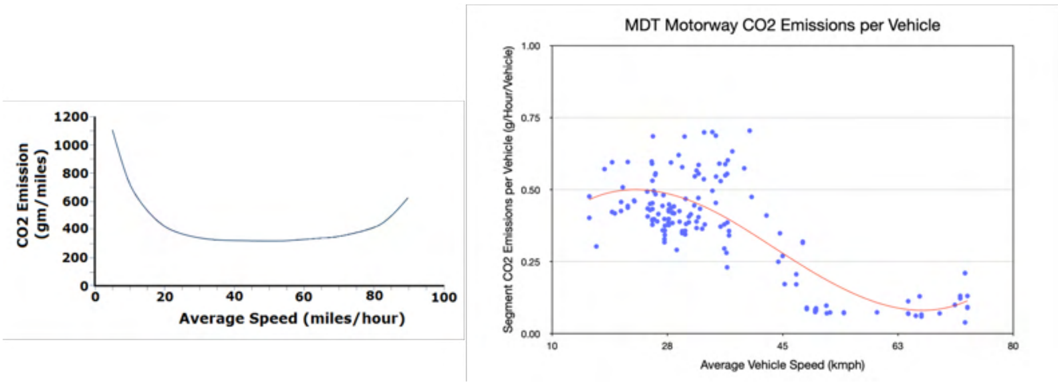


Figure 4: Left: The relationship between average speed and CO_2 emissions, by Shahid et al. (2014). Right: The average hourly CO2 emissions per vehicle on motorway or trunk segments in the MDT. Segments with no vehicles are ignored.

However, when further analysing emissions, the MDT was found to dramatically underestimate or overestimate levels for certain vehicle types, specifically for cars and HGVs respectively. This is noted in figure 5, which compares the percent of emissions generated by each vehicle type from the MDT system with the values reported by HFAS. Here, the total emissions for cars were overestimated by around 34%, whilst HGVs had their emissions underestimated by 36%. Although the emissions values for two-wheeled vehicles, buses and coaches and LGVs only varied from the HFAS values by around 1%, this does show a large difference between the two. However, in investigating this further, the MDT emissions values are very close to the average vehicle proportions, as seen in figure 6. Generally, the base values calculated by the system only deviated from their vehicle proportion by 1-3%, for all vehicle types, including HGVs and cars. These proportions should be accurate as these values are taken directly from the UKDT count points dataset, so a misjudgement of the impact for these two vehicle types by the emissions model appears to be the best explanation.

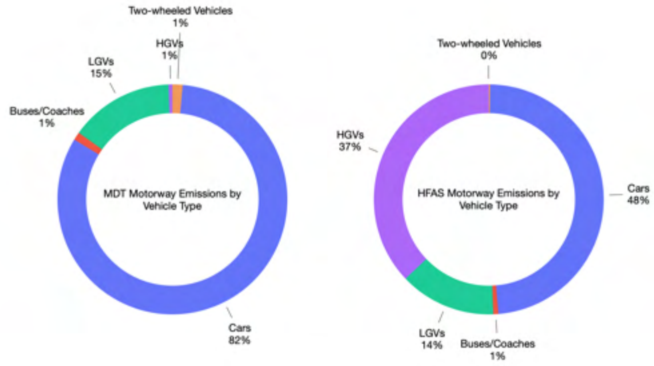


Figure 5: Total network emissions distributed by vehicle type for MDT's calculated values and HFAS reported values.

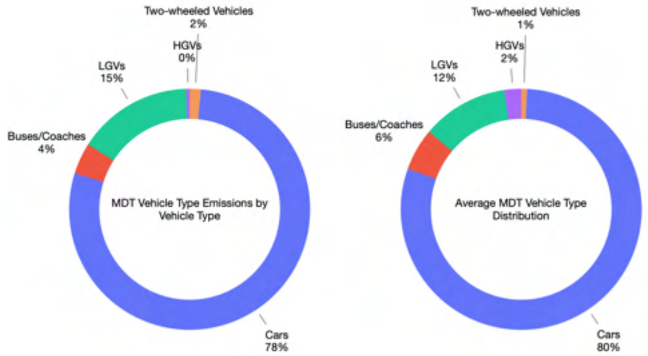


Figure 6: MDT total emissions vehicle by road type, against the average vehicle distributions for all segments in the network.

Another evaluated aspect was the distribution of emissions among the different road types, particularly focusing on 7 main classifications, motorways, trunks, primary, secondary, tertiary, residential and unclassified roads. Here, trunk roads are defined by OSM as a country's most important roads after motorways, whilst unclassified means a road is less important than tertiary but is not used for access to properties. OSM does use additional road types alongside these, particularly those that link roads of different types, but these are not included for the purposes of this evaluation. On the other hand, HFAS uses a much more simplified system, with only 3 road types, motorways, major roads and minor roads. This is most likely because the HFAS report is not as detailed as the OSM dataset, mainly focusing solely on motorways, A and B roads, and any road used by a bus route. However, for these purposes, trunks are included as motorways, primary and secondary as major roads, and any lower classifications are in minor roads. Regardless, the results of this comparison can be seen in figure 7, along with the distribution of road types in the MDT network. Motorway emissions show the smallest difference between the two, with the MDT system only overestimating the HFAS values by around 10%, whilst the other two perform only slightly worse. Major and minor roads were underestimated by 30% and overestimated by 20%, although they are arguably more loosely defined. A similar investigation to the vehicle types was done by then looking at the distribution of road types throughout the network, which showed that motorways emit much more than their proportion of the network and the opposite for minor roads, as was expected. Therefore, the MDT did not perform too badly in this area even though it did overestimate the impact of minor roads by a fairly large margin. The level of detail in the simulation of the two models could have affected this difference, but the HFAS values can still act as a good benchmark for evaluating the MDT's calculations.

Figure 7: The MDT and HFAS total emissions values by road type along with the distribution of road types in the MDT network, calculated using their total length as a percent of the total road network length.

4.2 User Feedback

4.2.1 Questionnaire

All answers from participants were given anonymously and on a scale of 1-10, except for question 4 which allowed for longer-form answers. Although they were also asked more basic questions, primarily to find which part of the system they used or whether they saw any technical issues, the main questions are as follows:

1. Before using the MDT, how was your understanding of the impact of congestion on greenhouse gas emissions?
2. Did you find the system easy to use in general?
3. Was the data presented in a way that made it difficult to interpret or understand?
4. Did any aspect, in particular, make this more difficult, or is there a way this could be improved?
5. Would you say using the MDT has improved your understanding of the impact of congestion on greenhouse gas emissions?
6. Would you say using the MDT has increased your interest in urban sciences at all?

4.2.2 Feedback Results & Evaluation

Feedback on the final system was generally positive, with most participants finding the system easy to use and understand. Although no experts in urban science took part in the user feedback, question 1 was aimed to find participants' level of knowledge of urban systems. It was generally phrased with the assumption that people might be able to guess certain relationships, such as that congestion is bad for emissions, but did receive relatively spread out answers. Around a third of participants scored themselves an average of 8, and the others just under 4, and so there was a decent mix in the level of knowledge within the group.

In terms of questions on the system itself, both question 2, which focused on how easy it was to navigate the interface and understand the network modifiers, and question 3 had an average response of 8.2 out of 10. This shows that the MDT system performed very well in following the goal of digital twins to communicate and visualise data, especially with the mix of responses to question 1. The main criticisms that came from question 4 however, on aspects that made the data more difficult to interpret, came from the panel layout and network modifiers. One participant, in particular, noted how the panel drop-down arrows were difficult to spot, with another adding that the map legend could get lost once someone scrolled down. Most also noted how there could be a better explanation of what network modifiers do, even just with a tooltip, and although the information was in the instructions window, a large amount of text could be seen as rather tiring or unappealing. Some other suggestions included the addition of landmarks to the main map, which would be useful especially for people unfamiliar with the city, or a reduction in the size of the green 'home screen' portion of the landing page as people could easily miss the context of the project.

Participants also complimented how professional the final interface looked, saying they enjoyed exploring the system and thought there was good potential. One in particular also said how they could see something similar working for a council or company, and recognised from their experience in a business setting that being able to download the raw data alongside the visuals was a great addition. However, the responses to question 5 and 6 showed the most promising results for the MDT, with an average of 7.8 and 7.0 respectively. Ultimately, I think this shows that the project was a success in the way it is able to help people gain insight and understanding into such a complex system. I'm very happy that the system received such a high score from these two questions, particularly for question 6, which I believe excellently demonstrates the potential of the MDT system and digital twins as a whole.

5 Future Scope & Conclusions

5.1 Project Critique

During the evaluation and development stages of the project, 3 major critiques can be drawn; the removal of the network layout modifications, the accuracy of the simulation itself, and the albeit smaller problem of loading times.

Originally, the network layout modifications were intended to act as a way to much more clearly demonstrate the relationship between traffic and emissions, as well as helping towards the goal of allowing the system to act as an evaluation tool for city design. Users would have been able to make changes to the road layout, such as changing the speed limit, closing roads or adding lanes, all decisions that real-world planners have to make. Evidently though, as previously detailed, this was ultimately deemed to not be possible for this type of application and was removed from the final system. Adding this kind of feature could have had been a huge help towards the goals of this project and made the system a much better demonstration of a digital twin. Whilst the data is still able to be visualised and interacted with, this dynamic element of digital twins could have been explored much further and interaction increased dramatically.

The accuracy of the simulation is a large issue with the current system, as the emissions calculated by the MDT varied significantly from real-world values in certain measures. Although no suitable street-level emissions data could be found for evaluation, it is clear that some elements of the calculations performed by the MDT don't perfectly reflect the real world. This is especially the case with HGVs and passenger cars, which had their impact underestimated and overestimated respectively. Lastly, the time needed to load networks had been one of the larger problems faced throughout the development process, leading to the need to add loading screens as previously said. This issue certainly had the smallest impact on the final system of the 3 and has been reduced as much as possible but, especially when loading the full 12-hour simulation, is less than ideal. This also lead to having to limit the length of the simulation in the deployed version of the web application due to the timeout value set by the Nginx proxy server, which does potentially lessen the value in the system.

5.2 Future Scope

Along with some improvements to the user interface, as suggested in section 4.2.2, the main addition to the system would be some form of network modification. This would more closely follow the goals set at the beginning of the project and could include more of the traffic modelling aspects of the preliminary research. This could be possible as, despite the complexity and computational costs of traffic assignment solvers, there have been successful implementations in similar applications, albeit in more computationally expensive forms. This could therefore mean that the MDT system could be extended into a downloadable software that might be able to better perform these tasks without the limitations of Django or a web server. Potentially, this software could also be more aimed towards planners or developers, allowing it to focus more on some technical aspects of urban design, but this would mean that the aspect of the current system that appeals to non-experts would certainly be lost. Whilst I do still definitely believe that a web app is a much better choice for the implementation of the current system, if it was to be expanded with these features, this change could be worth considering as an alternative.

Another possible addition could be to apply the same system to other cities or urban areas. The MDT does aim to be a proof of concept for applying digital twins to urban and environmental science, so showing how the system works in multiple situations would certainly be beneficial to this. As has already been shown in appendix D, the current system can render much larger networks so potentially one of these cities could be tried, although maybe with a limit on the length of the simulation to one daily average value so segments do not have to be repeatedly redrawn.

5.3 Final Conclusions

Overall, despite the removal of some planned aspects of the project, I am confident that the current system fulfils many of the goals I set out to achieve. As mentioned previously, digital twins aim to help experts and non-experts alike gain an understanding of how complex real-world systems function and from the feedback results, the MDT has performed excellently in this measure. Participants noted how the system looked professional and was able to visualise the data in an easy to interpret and interesting way. I was particularly proud of the response to the final question asking if the MDT increased participant's interest in urban science at all, which received an average of 7.0/10. This is alongside the responses to question 5, asking if participants had an improved understanding of the impact of congestion on emissions after trying the system, which received an even higher average of 7.8/10. Therefore, this feedback certainly shows the potential of even basic digital twins, and their ability to visualise these complex systems.

This is of course despite the issues highlighted above surrounding the model's accuracy and the network layout manipulation. However, whilst the accuracy would require a fair amount of attention if the system was to be further expanded, I believe that the current system can still be considered a success even without the network manipulation. Although it would have been an excellent feature, the system still is able to follow the principles of digital twins, and can still be used as an evaluation tool for city design, albeit current designs. As well, in place of this feature, the emissions modifiers perform well at encouraging interaction with the urban environment in an approachable manner. Ultimately though, I do believe that this system is successful in showing how digital twins can be applied to urban and environmental sciences and demonstrates how this technology if properly implemented, could have a large impact on the future sustainability of our cities.

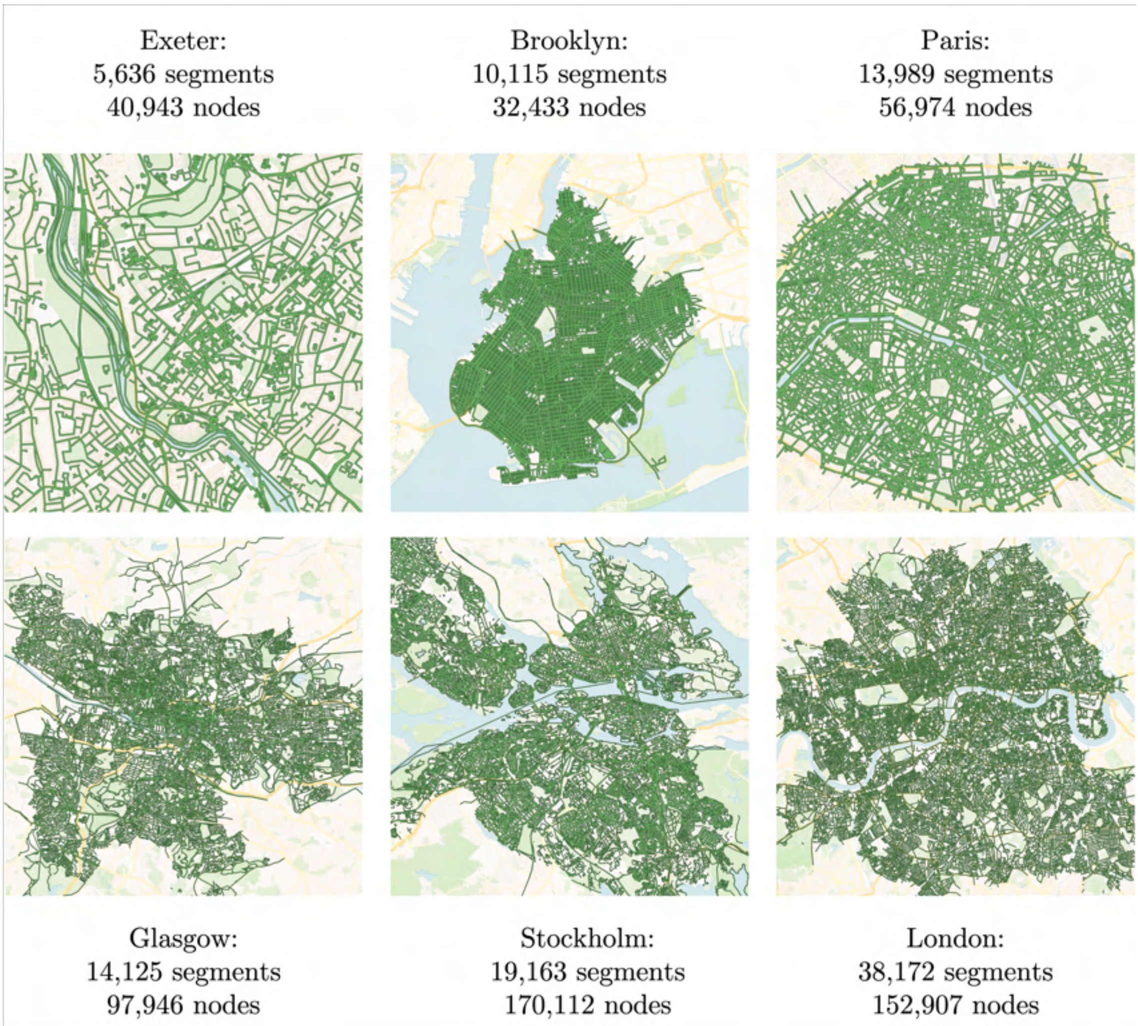
Appendices

```
query.txt
[timeout:900][out:json];
(area(3609070835);area(3609070834);)->.a;
    (
        way(area.a)
            ['name']
            ['highway']
            ['highway' !~ 'path']
            ['highway' !~ 'pedestrian']
            ['highway' !~ 'steps']
            ['highway' !~ 'raceway']
            ['highway' !~ 'bridleway']
            ['highway' !~ 'proposed']
            ['highway' !~ 'construction']
            ['highway' !~ 'elevator']
            ['highway' !~ 'bus_guideway']
            ['highway' !~ 'footway']
            ['highway' !~ 'cycleway']
            ['highway' !~ 'crossing']
            ['foot' !~ 'yes']
            ['access' !~ 'private']
            ['access' !~ 'no']
            ['area' !~ 'yes'];
        node(w)(area.a);
    );
out;
```

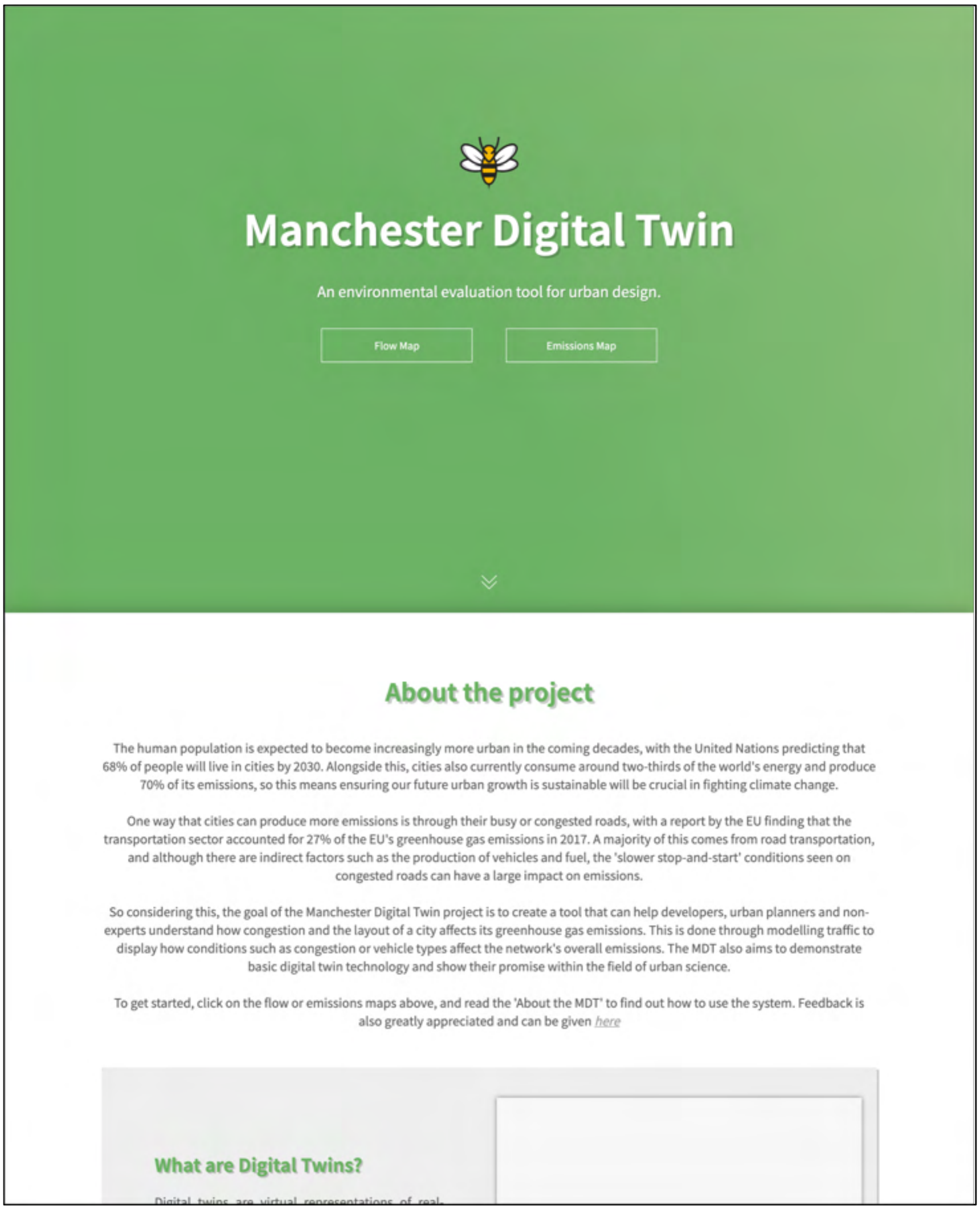
Appendix A: The query fetches area 9070835 (Deansgate) and 9070834 (Picadilly), and returns ways with a name and highway tag, as long as they fulfill the other requirements.

Appendix B: The OSM network is shown in green, whilst the TOMTOM network is in red. Left: The Manchester Arndale car park is highlighted, and is represented by TOMTOM, but not OSM. Right: The TOMTOM network includes ‘The Left Bank,’ a recent pedestrianised development.

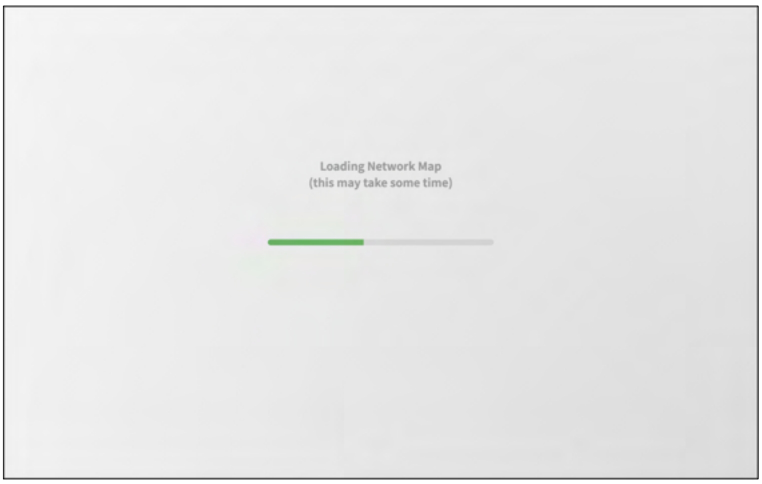
Appendix C: The OSM network is shown in green, whilst the TOMTOM network is in red. Left: The area around Deansgate train station, showing the difference in coordinates between datasets. Right: The A6/Mancunian Way junction and the difficultly of matching parallel roads.



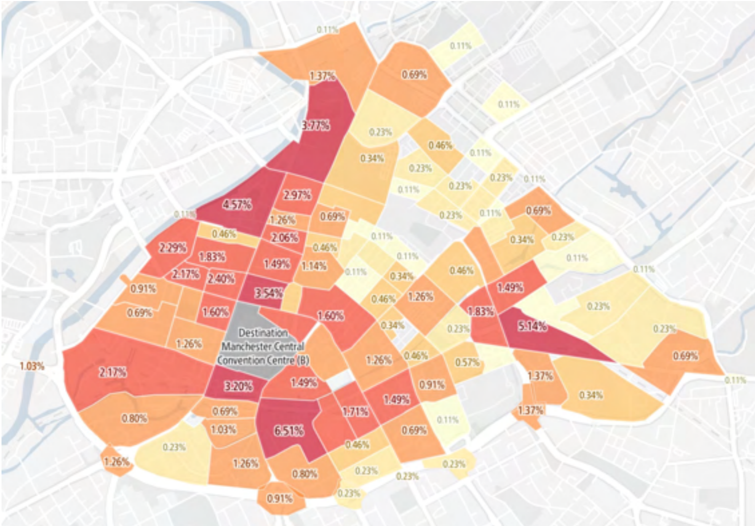
Appendix D: This shows the capability of the network representation and map renderer, being able to render large and complex networks in reasonable time.



Appendix E: The website's landing page, with part of the 'About the project' section.



Appendix F: The loading screen displayed when moving between views.



Appendix G: The OD matrix from the TOMTOM dataset using custom origin and destination nodes. The map is showing the origin distribution for journies that ended at the Manchester Central Convention Centre.

References

A. Rasheed, O. San and T. Kvamsdal (2020). 'Digital Twin: Values, Challenges and Enablers From a Modeling Perspective'. In: *IEEE Access* 8, pp. 21980–22012. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2970143.

ABI Research (Sept. 2019). *Transformative Urban Digital Twin and City Modeling Deployments to exceed 500 by 2025*. URL: https://www.abiresearch.com/press/transformative-urban-digital-twin-and-city-modeling-deployments-exceed-500-2025/ (visited on 18/04/2021).

Albrecht, A R, P G Howlett and D Coleman (2009). 'Application of Origin-Destination Matrices to the Design of Train Services'. In: *Australasian Journal of Engineering Education* 15.2, pp. 95–104. DOI: 10.1080/22054952.2009.11464021.

Bazghandi, Ali (Jan. 2012). 'Techniques, Advantages and Problems of Agent Based Modeling for Traffic Simulation'. In: *International Journal of Computer Science Issues* 9.1, pp. 115–119.

Bolton, Ruth N. et al. (Oct. 2018). 'Customer Experience Challenges: Bringing Together Digital, Physical and Social Realms'. In: *Journal of Service Management* 29.5, pp. 776–808. DOI: 10.1108/JOSM-04-2018-0113.

C40 (2021). *Why Cities?* URL: https://www.c40.org/why_cities (visited on 30/03/2021).

California Air Resources Board Office of Strategic Planning (1989). *The Air Pollution-transportation Linkage: How Vehicle Travel on Our Roads and Highways Affects the Air We Breathe : a Report from the State of California, Air Resources Board, Office of Strategic Planning*. Air Resources Board, Office of Strategic Planning.

Castro, Pablo Samuel, Daqing Zhang and Shijian Li (2012). 'Urban Traffic Modelling and Prediction Using Large Scale Taxi GPS Traces'. In: *Pervasive Computing*. Ed. by Judy Kay et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 57–72. ISBN: 978-3-642-31205-2.

Chanut, Stephane (2005). 'A First-Order Macroscopic Traffic Flow Model for Mixed Traffic Including Moving Bottleneck Effects'. In: College Park Maryland, United States, pp. 323–343. ISBN: 978-0-08-044680-6.

Chen, Ruiwei and Vivien Mallet (Jan. 2016). *pollemission*. URL: https://github.com/pollemission/pollemission (visited on 01/04/2021).

Cityzenith (2021). *Amaravati Smart City - Cityzenith - The World's Most Advanced Digital Twin Software Solution*. URL: https://cityzenith.com/customers/amaravati-smart-city (visited on 18/04/2021).

Corral-Acero, Jorge et al. (Dec. 2020). 'The 'Digital Twin' to enable the vision of precision cardiology'. In: *European Heart Journal* 41.48, pp. 4556–4564. DOI: 10.1093/eurheartj/ehaa159.

Daganzo, Carlos F (Feb. 2002). 'A Behavioral Theory of Multi-lane Traffic Flow. Part I: Long Homogeneous Freeway Sections'. In: *Transportation Research Part B: Methodological* 36.2, pp. 131–158. DOI: 10.1016/S0191-2615(00)00042-4.

Daganzo, Carlos F and Yosef Sheffi (1977). 'On Stochastic Models of Traffic Assignment'. In: *Transportation Science* 11.3, pp. 253–274. DOI: 10.1287/trsc.11.3.253.

Dembski, Fabian et al. (Mar. 2020). 'Urban Digital Twins for Smart Cities and Citizens: The Case Study of Herrenberg, Germany'. In: *Sustainability* 12.6, p. 2307. DOI: 10.3390/su12062307.

Esteves-Booth, A et al. (Aug. 2002). 'A review of vehicular emission models and driving cycles'. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 216.8, pp. 777–797. DOI: 10.1243/09544060260171429.

European Environment Agency (Dec. 2020). *Greenhouse Gas Emissions From Transport in Europe*. Indicator Assessment. URL: https://www.eea.europa.eu/data-and-maps/indicators/transport-emissions-of-greenhouse-gases-7/assessment (visited on 27/04/2021).

Florian, Michael, Michael Mahut and Nicolas Tremblay (2005). 'Application of a Simulation-Based Dynamic Traffic Assignment Model'. In: *Simulation Approaches in Transportation Analysis: Recent Advances and Challenges*. Ed. by Ryuichi Kitamura and Maso Kuwahara. Boston, MA: Springer US, pp. 1–22. ISBN: 978-0-387-24109-8. DOI: 10.1007/0-387-24109-4_1.

Greenshields, B. D. et al. (1934). 'THE PHOTOGRAPHIC METHOD OF STUDYING TRAFFIC BEHAVIOR'. In: *Proceedings of the Thirteenth Annual Meeting of the Highway Research Board*.

Vol. 13. Washington DC, USA: Highway Research Board, pp. 382–399. URL: https://trid.trb.org/view/120821 (visited on 27/04/2021).

Grieves, Michael (Jan. 2019). 'Virtually Intelligent Product Systems: Digital and Physical Twins'. In: *Complex Systems Engineering: Theory and Practice*. Ed. by Shannon Flumerfelt et al. American Institute of Aeronautics and Astronautics, pp. 175–200. ISBN: 978-1-62410-564-7.

He, Feifei et al. (2016). 'A Traffic Congestion Assessment Method for Urban Road Networks Based on Speed Performance Index'. In: *Procedia Engineering*, pp. 425–433. DOI: 10.1016/j.proeng.2016.01.277.

Highways Forecasting and Analytical Services (HFAS) (June 2013). *The Greater Manchester Emissions Inventory 2010 Update*. Tech. rep. 1750. Manchester: Transport for Greater Manchester, p. 113. URL: https://www.gmtu.gov.uk/reports/emigma/HFASReport1750v1.0.pdf (visited on 22/04/2021).

James Lighthill, Michael and Gerald Beresford Whitham (May 1955). 'On Kinematic Waves II. A Theory of Traffic Flow on Long Crowded Roads'. In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229.1178, pp. 317–345. DOI: 10.1098/rspa.1955.0089.

Kaur, Maninder Jeet, Ved P. Mishra and Piyush Maheshwari (2020). 'The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action'. In: *Digital Twin Technologies and Smart Cities*. Ed. by Maryam Farsi et al. Internet of Things. Cham: Springer International Publishing, pp. 3–17. ISBN: 978-3-030-18731-6.

Khreis, Haneen and James Tate (June 2017). 'Alternative Methods for Vehicle Emission Modelling and Its Impact on Local Road Transport Emission Inventories in Bradford, UK'. In: *Journal of Transport & Health* 5, S50. DOI: https://doi.org/10.1016/j.jth.2017.05.341. URL: https://www.sciencedirect.com/science/article/pii/S2214140517305005.

Knez, Matjaž and Katja Baškovič (June 2013). 'A review of vehicular emission models'. In: *Preconference proceedings of the 10th International Conference on Logistics & Sustainable Transport 2013*. URL: https://www.researchgate.net/publication/257207886 (visited on 19/04/2021).

Li, Yvonne, Ben Pearson and Tim Murrells (Dec. 2009). *Updated Vehicle Emission Curves for Use in the National Transport Model*. Tech. rep. Issue 2. ITEA Division, The Department for Transport, p. 32. (Visited on 07/04/2021).

Lundgren, Jan T. and Anders Peterson (May 2008). 'A heuristic for the bilevel origin–destination-matrix estimation problem'. In: *Transportation Research Part B: Methodological* 42.4, pp. 339–354. DOI: 10.1016/j.trb.2007.09.005.

Mitchell, Robert B and Chester Rapkin (1954). *Urban Traffic: A Function of Land Use*. Publications of the Institute for Urban Land Use and Housing Studies, Columbia University 1. New York, NY: Columbia University Press. ISBN: 978-0-231-89909-3.

Morton, Craig (Feb. 2017). *The European Emission Standards of the Car Fleet Registered across Scotland's Main Cities*. Tech. rep. Edinburgh: ClimateXChange, p. 8. (Visited on 07/04/2021).

National Research Foundation (Feb. 2021). *Virtual Singapore*. URL: https://www.nrf.gov.sg/programmes/virtual-singapore (visited on 27/04/2021).

Ntziachristos, Leonidas and Zissis Samaras (Nov. 2000). *COPERT III: Computer Programme to Calculate Emissions from Road Transport: Methodology and Emission Factors (Version 2.1)*. Technical Report 49. Copenhagen, Denmark: European Environmental Agency.

Peterson, Anders (2007). *The Origin Destination Matrix Estimation Problem Analysis and Computations*. Linköping Studies in Science and Technology Dissertations 1102. Linköping: Institutionen för teknik och naturvetenskap. ISBN: 978-91-85831-95-1.

Richards, Paul I. (Feb. 1956). 'Shock Waves on the Highway'. In: *Operations Research* 4.1, pp. 42–51. DOI: 10.1287/opre.4.1.42.

Robson, Steve (Feb. 2021). 'More homes built in Manchester city centre in 2020 than ever before'. In: *Manchester Evening News*. Section: Greater Manchester News. URL: https://www.manchestereveningnews.co.uk/news/greater-manchester-news/more-homes-built-manchester-city-19748186 (visited on 27/04/2021).

Shahid, Shamsuddin et al. (Sept. 2014). ‘Figure 13, Assessment of Greenhouse Gas Emission Reduction Measures in Transportation Sector of Malaysia’. In: *Jurnal Teknologi* 704, pp. 2180–3722. DOI: 704(4):2180-3722.

Sjöstrand, Helena (2001). ‘Passenger assessments of quality in local public transport- measurement, variability and planning implications’. PhD thesis. Lund: Lund University, Lund Institute of Technology, Department of Technology and Society, Traffic Planning.

Smit, R. et al. (2017). ‘A tunnel study to validate motor vehicle emission prediction software in Australia’. In: *Atmospheric Environment* 151, pp. 188–199. DOI: https://doi.org/10.1016/j.atmosenv.2016.12.014.

Stagecoach Manchester (Apr. 2020). *Stagecoach Manchester Annual Report 2019-2020*. Tech. rep. Stagecoach. URL: https://www.stagecoachbus.com/about/manchester (visited on 27/04/2021).

Tuegel, Eric J. et al. (2011). ‘Reengineering Aircraft Structural Life Prediction Using a Digital Twin’. In: *International Journal of Aerospace Engineering*, pp. 1–14. DOI: 10.1155/2011/154798.

United Nations (May 2018). *68% of the World Population Projected to Live in Urban Areas by 2050, Says UN*. URL: https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html (visited on 27/04/2021).

Zhang, Kai and Stuart Batterman (Apr. 2013). ‘Air Pollution and Health Risks Due to Vehicle Traffic’. In: *Science of The Total Environment* 450-451, pp. 307–316. DOI: 10.1016/j.scitotenv.2013.01.074.

Zhang, Xing Quan (May 2016). ‘The trends, promises and challenges of urbanisation in the world’. In: *Habitat International* 54, pp. 241–252. DOI: 10.1016/j.habitatint.2015.11.018.