Bridging the Gap: Improving Data Accessibility and Comprehension in Smart Cities Through a User-Friendly Dashboard

Rufus Ewings

Principal Supervisor: Alma Cantu

February 29, 2024

Abstract

Rapid urbanisation and rising air pollution pose a significant threat to global health and climate. The availability of accessible data visualisations can provide key public insight into local pollution trends and hotspots. Although Newcastle Upon Tyne plays host to one of the countries largest collection of environmental data, the lack of an application dedicated to presenting this data to the non-technical public puts a barrier between the community and meaningful climate action. This paper presents a user-friendly dashboard comprising of meaningful and powerful visualisations and tools for local pollution analysis. By putting these tools in the hands of the public, institutions and policymakers are empowered to make informed decisions that contribute to sustainable urban living.

Index terms— Pollution, Visualisation, Smart City, Environment

Contents

1 Introduction

Rapid urbanisation has led to a significant increase in air pollution (Fecht et al., 2015): contributing greatly to climate change and posing a considerable threat to public health. The UK Government (2018) estimate that human-caused air pollution has an impact equivalent to 28,000 to 36,000 early deaths in the UK alone. 11.65% of deaths on the global stage are attributed to air pollution (Ritchie and Roser, 2017) and "91% of the world's population live in areas where air pollution exceeds safety limits" (Qu et al., 2007). By 2030 it is projected that 60% of the global population will live in cities (UN Population Division, 2018). As cities currently consume more than 78% of the world's energy and already produce 60% of greenhouse gases (UNC, 2018), as such making a considerable effort towards sustainable living is more critical than ever.

One potential new combatant against climate change, the concept of the "Smart City", utilises real-time environmental data (Cepero et al., 2022) to help build a "comprehensive vision" of a city. This vision can be used to inform policy-makers, promote responsible management of natural resources, and drive wider innovation by providing institutions with access (Trindade et al., 2017). Newcastle upon Tyne, considered a leading smart city(Bris and Lanvin, 2021) (Foresight, 2019), hosts a vast network of sensors. These work to collect a wide range of environmental data, including air quality, which can provide valuable insights into pollution levels. However, the potential of this data remains untapped until it is made accessible to all stakeholders. Without the implementation of a user-friendly interface and visualisations understandable by people of all disciplines, this resource cannot be fully utilised: a library of knowledge written in an unfamiliar language.

The problem with the current handling of data in Newcastle is the lack of an application specifically striving to present a clear overview of air pollution for public comprehension. Current solutions either focus on presenting sensors as a resource for access by those with a technical background, or lack enough detail for a level of sophisticated analysis. As a result the range of people who could put this data to use for the purpose of battling climate change is limited. A user-friendly application is required to address this issue- empowering the people to make informed decisions about their daily lives and facilitate engagement in the community to combat pollution with whatever means are available to them. As noted by Cepero et al. (2022), open data needs to be presented in citizen-accessible and citizen-intelligible form to truly function as "open information". There is an implication of duty to provide a front end to data and enable information to be used by "experts, decision-makers, and service providers alike", rather than left in the dark: "only to be seen and handled by staff".

The aim of this project, therefore, is to fulfill this duty by bridging the gap between data collection and comprehension, bringing this resource to a wider audience and increasing the opportunities for innovation. This

takes the form of a real-time air pollution dashboard that presents the sensor data in a coherent and organised manner, with meaningful visualisations and tools for flexibility. The Urban Observatory (henceforth referred to as the UO) (University, 2023) sensors are a valuable resource that provides real-time environmental data at a scale vastly unmatched across the United Kingdom - boasting considerably more sensors than both Birmingham and Manchester combined. As of December 2021, Newcastle's UO hosts 1178, with Birmingham's and Manchester's sensor total being only 193 and 492 respectively (James et al., 2021).

Serving as a front-end to the UO's extensive network of sensors, this UO dashboard seeks to bridge the gap between data collection and comprehension, granting access to the diverse range of stakeholders which would benefit from access to this data: from urban planners to the general public. By providing a user-friendly interface, the dashboard will bring a broader audience access to Newcastle's real-time environmental data. Information is available for both temporal and spatial analysis through a series of graphs, maps and charts. These are presented in a clear and user-friendly manner and grant the community new insight into environmental trends, patterns, and anomalies - seeking both to inform policy-making and boost institutional interaction.

Particular focus should be given to ensuring accessibility to users without a technical background. Depth and flexibility should be present to allow for meaningful analysis, but should not take precedence, only being implemented as far as it is not to the detriment of the ease of use and intuitiveness of the application. Despite being created on a local scale, the creation of clear, understandable visuals for Newcastle aims to act as a contribution toward global climate change mitigation efforts. For the ecological impact of urbanisation to be curbed each city must play its part in creating a more sustainable future. The creation of this dashboard aims to demonstrate the importance of making pollution data not only available, but accessible, with the fight for more sustainable living being one that everyone must be able to access. From this further applications for the purpose of pollution data accessibility could be created in Newcastle and beyond.

This paper is split into a series of sections, with the next section detailing some existing solutions, with examples in Newcastle, Bristol and beyond. Section 3 breaks down the methodology, detailing the selection of Jupyter and the various libraries used, before describing the visualisations and how they are formed together into a single dashboard. Sections 4 and 5 break down the results of the project, presenting the outcome: its successes, failures and any findings that were made along the way before finally concluding the work with section 6 - the conclusion.

2 Literature Review

2.1 Urban Observatory

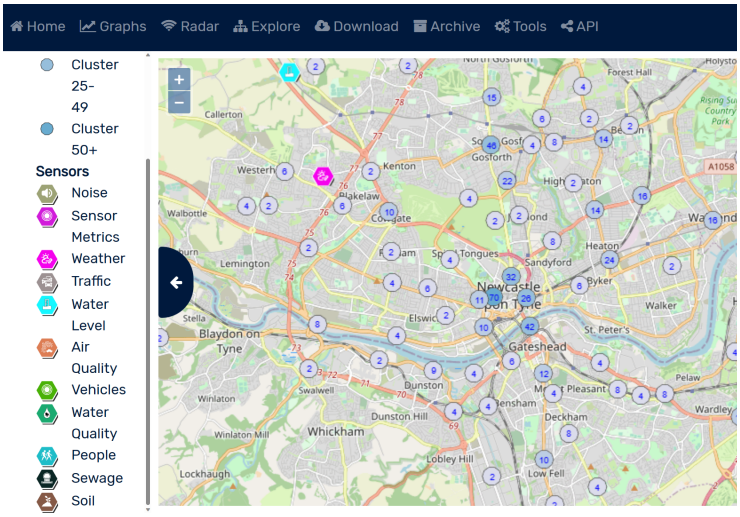

Figure 1: The Urban Observatory landing page.

The UO's (in Newcastle unless otherwise stated) website presents its vast number of sensors in various ways, with the primary landing page being a map of all the sensors, categorised by type. Clicking on a sensor brings up the data for that sensor for the last seven days, with buttons for 24-hour or monthly data generating the corresponding line graph. Buttons for all the available metrics (PM2.5, NO2) appear on the bottom right. This demonstrates the progress made towards creating an effective front-end, but there is much room for improvement when it comes to presenting air pollution data for easy visualisation.

A map of all the sensors is available but their data can only be viewed individually, preventing the user from analysing how pollution varies across the city. Though the 'graphs' page will generate a plotted graph for a fixed time frame, sophisticated analysis of patterns is limited by the lack of flexibility. Visualisations can be generated and compared for different data aggregation and even specific sensors (though many are not labelled with their locations), but the graph type is limited to a line graph. The time period is fixed to the past two months and although the ability to zoom in to the graph for closer inspection is present, the ability to adjust the time frame would prove easier to navigate and as such is applied in this project's implementation.

Although the themes page provides access to CSV files of a wide range of data, the lack of visualisations and analysis tools puts up a 'barrier of

entry' to proper interpretation. The API offers great flexibility and the ability to programmatically access vast amounts of data, but due to its technical nature, accessibility is limited to those with data science expertise: knowledge of REST API, data handling and visualisation generation. Many levels of access are available and a glimpse is given into just how much information is available, but the front-end could be improved upon to enable more meaningful analysis. Though the website acts very effectively as a sensor broker for those with a data science background, it is more focused on individual sensors than providing an overarching visualisation of Newcastle and isn't suitable for the interpretation of air pollution in the area. Providing an overview of the entire city would enable citywide analysis and provide a more solid basis for decision-making.

A front-end focused on visualisation would help greatly with real-world problem-solving and decision-making. Making this data available to more institutions would make the great expense and effort of setting up these sensors even more worthwhile as a contribution to the global efforts against climate change. This project takes on the task of creating a front-end 'dashboard': to give an immediate visualisation of pollution data for the entire city: collating all the sensors into meaningful graphs, charts and interactive maps for air pollution analysis; complete with widgets and options for customisation.

2.2 Manchester-i

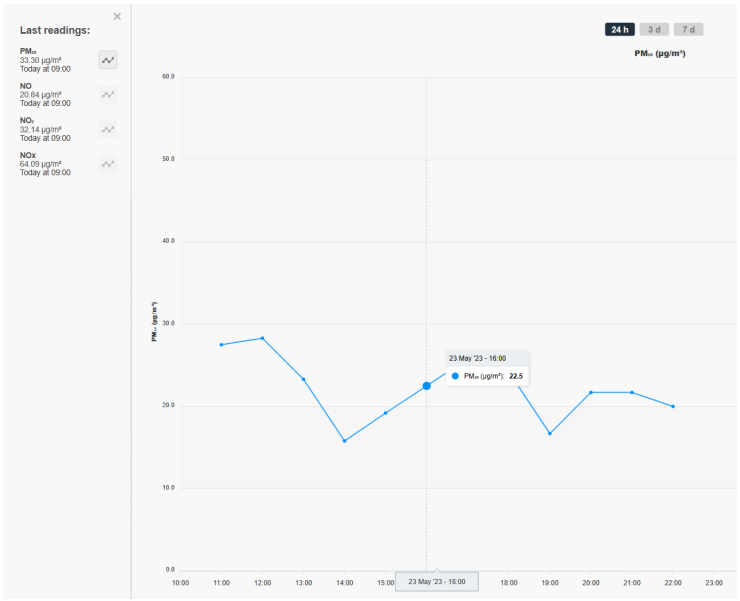

Figure 2: The Manchester-i landing page.

The Manchester Urban Observatory (Observatory, 2023a) (or Manchester-i) is a similar project set up in the City of Manchester, hosting 16 air quality sensors among other sensors such as traffic.

Similarly to the Newcastle UO website, selecting a specific sensor type requires unselecting all other types. This is slow as, taking the example of Newcastle's UO, there are 11 categories - meaning 10 must be switched off to focus on one. Taking inspiration from the snappiness of the other widgets on this site, switching variables is replaced by a drop down menu, allowing for quick and easy switching between datasets.

To look at a sensor's data the map must be zoomed all the way in until the widget to select that specific one appears - before this being represented by a bubble with a number denoting the amount of sensors in the area. Though this is likely done to avoid clutter, it is relatively slow, mainly due to the slow scrolling speed on the map which puts it behind the Newcastle UO in terms of perceived responsiveness. In response, this project presents all data points on a spike map, with the data points all visible at once and fast, responsive map movement on interactive maps.

Once a sensor is selected, the most recent readings of different variables can be viewed on a line graph instantaneously. The bright and vibrant blue colour used for the graph makes it easily readable and the widgets to select different variables feel snappy and responsive. Different time periods can be switched between with buttons at the top, such as 24 hours, 3 days and 7 days, with animations between each making the graph visually pleasing. Hovering over a data point brings up a vertical line and a tool tip of information about that particular reading.

Overall, although the Manchester website functions more as a sensor broker than a pollution dashboard it is a strong example of sensor data visualisation in terms of visuals and animation. It would benefit from an overview of averages, customisability and more graph types.

Inspiration has been taken from this in the project to help create a powerful yet accessible tool. The tool-tip feature is implemented on the interactive map, so that any particular sensor reading can be checked with ease, instantly showing up without the need to click on it or trawl through a site for that sensor's data. Switching between different time periods was implemented but opted for a slider widget over buttons to give the user a greater feeling of control over the time period. Further, the vibrant blue colouring inspired the graph colouring for many of the plots in this project, due to its vibrant yet professional look, creating 'easy-viewing' and clear visuals to improve user engagement.

2.3 Open Data Bristol

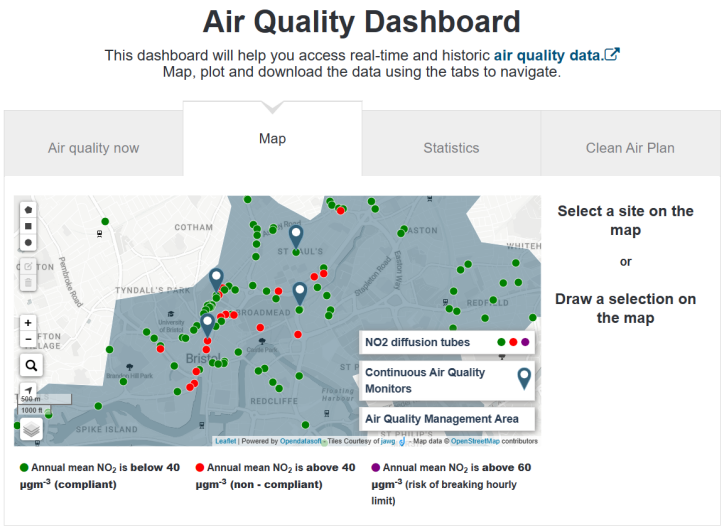

Figure 3: Bristol Air map tool

Open Data Bristol (Bristol, 2023) host an air pollution dashboard for visualising air quality in Bristol, formatted with tabs to switch between different screens. These consist of "Air Pollution Now", "Map", "Statistics" and "Clean Air Plan". Air pollution now contains the 7 air pollution sensors with a straight bar gauge marked with the potential levels: low, medium, high, very high as according to Gov UK recommendations (Air, 2023). The bar fills up to indicate what level the pollution is at and a tool-tip appears on hovering over to provide more information. This is a very effective and immediate indicator of air pollution across the city at a glance. Only the continuous analysers were selected, five NO2 and two PM10, to be displayed. These represent the most sophisticated sensors and provide updates every 15 minutes. Clicking on a sensor provides extensive information about it including a photo, description and statistics such as height off the ground, distance from the kerb and hourly average.

The map presents many data points, predominantly NO2 diffusion tubes and is colour-coordinated. Dots are either green, red or purple to denote compliant, non-compliant or 'at risk of breaking hourly limit'. One issue with the use of green and red is the lack of accessibility to those with red-green colourblindness, providing a barrier to access of information for those people. Besides this issue, the use of just three colours to display the safety levels of each location comes with pros and cons. On one side the specific levels can't be observed and data points could be just under or over the levels to fit in a category - which would not be visualised. Further, it is more difficult to compare the different levels across the map

than if a gradient had been implemented, however, in fulfilling the goal of simplicity the map succeeds - highlighting locations which exceed the recommended or safe levels.

Statistics more or less combines the previous two tabs, offering a table and a map of concentration summaries. The number of sites breaking compliance is noted, providing a clear ratio of how much of the city is within safe levels. Finally, the 'clean air plan' tab provides links to a number of websites related to the city's clean air strategy and locations where the user can find out more about air pollution. Outside of the dashboard, the website contains tools to browse tables of specific historical data and add them to visualisations if the user knows what they're looking for. While the ability to create visualisations is useful, the need to add specific datasets to have a complete understanding of the data prevents easy use as compared to the pre-built visuals.

Overall, the Open Bristol dashboard provides a clear overview for ease of access, however, for any greater detail of flexibility, one must delve into the process of looking through historical data and building their own visuals. A middle ground, between the basic, preset visuals and an ocean of data should be struck. This would improve the users' ability to analyse air pollution data without requiring a solid foundation in data science, but still giving them enough information to draw complex conclusions. This dashboard is an example of the sensor data being put to use to inform and educate the public, and despite hosting far fewer sensors has achieved a more effective front end than that of the Urban Observatory.

Taking note of the effectiveness of simplicity, this project includes scatter plots and choropleth maps to provide easy-to-read visuals, but with the removal of lines for clarity of the much greater number of sensors and with the introduction of gradients to provide more data whilst retaining the colour coded visualisation. Open Bristol's effective use of tabs to separate graph types is also taken on board for this project, for clear division and straightforward navigation of varying graph types.

2.4 UK Air Forecasting

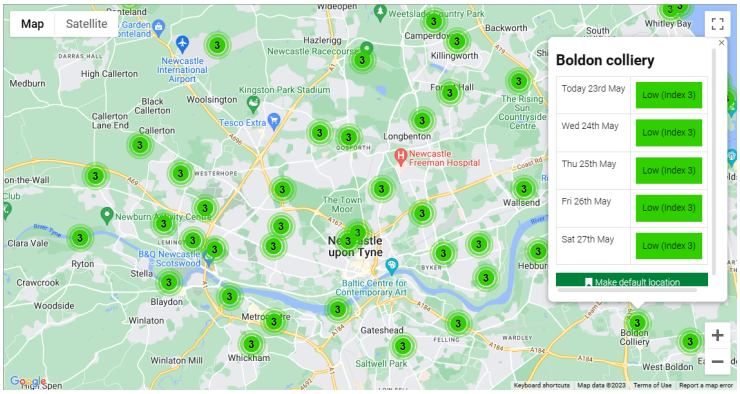

Figure 4: GOV UK Air pollution forecasting dashboard.

The UK Government Department for Environment Food & Rural Affairs hosts a tool for air pollution forecasting as part of their Air Information Resource. This tool seeks location input from the user before bringing up a map focused on that area. In Newcastle, a collection of over 20 predictions for Met Office sensors are available, with each containing a number representing the air quality based on five pollutants: ozone, nitrogen dioxide, sulphur dioxide, PM2.5 and PM10. The air quality is rated on a scale of 1 to 10 from low to very high, and predictions for the next 5 days can be looked at. The data points are colour coded on a scale of green through red to purple. Once again, this leads to the potential issue for those with colour blindness, demonstrating the lack of accessibility to effective data visualisation to this portion of the population and highlighting the necessity for consideration of this when selecting colour schemes.

The 5-day forecasting is calculated through a new model called AQUM and acts as a warning sign in the event that pollution moves beyond safe levels. Despite improvements having been suggested (Neal et al., 2014) and some biases, the model is overall useful for prediction (Savage et al., 2013) and its governmental provenance grants it credibility. Presenting the data points as a series of dots coloured along a gradient is effective for immediate data comprehension. This acts as part of the inspiration for implementing a choropleth map in the UO dashboard, with data points similarly marked with colour-mapped dots according to a gradient. Although the simplicity of presenting pollution as a 1 to 10 scale aids accessibility and is in line with government standards, this was replaced in favour of having the gradient represent real values. This ensures accuracy and grants access to more real data, with the colour gradient presented here having the potential issue of obscuring the problem - Newcastle, Port Talbot and the Scottish Highlands are all currently marked with a 3 and as such the usefulness of the graph is extremely limited for analysis.

While the tool presents real-time data, it is not possible to view the figures beyond the index. This and historical data is unavailable and the user must navigate to the data tab and specify a date range, monitoring site and pollutant - limiting user accessibility drastically. As a forecasting tool, it is suitable for the purpose of identifying at-risk areas (Birmingham is marked 4) and encouraging active action in those areas but does not fulfil the need for accessible air pollution visualisation for analysis.

3 Methodology

3.1 Overview

The dashboard aims to provide high-level access to real-time environmental data in a way that bridges the gap between complex sensor data and straightforward visualisations for comprehension by non-technical users. It should be detailed enough for a meaningful depth of understanding, but visually simplistic enough to tell this story 'at a glance'. Detail should not hinder accessibility, and as such the data quantity and visual appearance must be carefully considered to strike a balance - more data points may increase accuracy but at the drawback of overcrowded graphs and overwhelming visualisation. Pre-processing is critical to visualising data effectively, especially due to the presence of extreme outliers and high-frequency intervals. This section examines the strategies used to tackle these challenges and to present information for straightforward comprehension, aiming to be clear, flexible and fast.

3.2 Tool Justification: Python, Jupyter and Libraries

Python houses a vast number of libraries for data visualisation and is generally considered a good choice (Cao et al., 2021) for clear and concise data visualisation, with Jupyter especially being the *de facto* standard (Perkel, 2018). Although software such as Power BI could be used for generating graphs from the '*RESTful*' API (as described by (IBM, 2023): "REST APIs communicate via HTTP requests to perform standard database functions like creating, reading, updating, and deleting records within a resource") without programming expertise, it is here that some limitations begin to arise in using the software, with pull refreshes limited to 8 times per dataset per day (Microsoft, 2021).

Python allows for greater flexibility and speed of data processing. It grants access to a number of effective libraries for machine learning (learn developers, 2023)(Developers, 2023a) and analysis (Developers, 2023c)(Developers, 2023b) and provides a strong foundation for future increases in complexity. Power BI does allow for the implementation of Python scripts (Microsoft, 2023) and for a cleaner look could be considered for providing visualisation of results in the future to leverage the strengths of both pieces of

software.

The Jupyter Notebook breaks code up into 'cells' making for easy debugging and modifications. The combination of code, widgets and visualisations together on one page provides a platform for fast tweaking and fixing, allowing for the best combination of options and features to be found swiftly through agile testing and data exploration.

Libraries The *matplotlib*, *seaborn* and *plotly* libraries are used for different visualisations in the dashboard with each bringing its own strengths and weaknesses. All integrate well with *Pandas* (Bisong, 2019) (Inc., 2023), which forms the basis for all the data manipulation and analysis in this dashboard, and as such all are effective choices for straightforward implementation of visualisations.

Pandas *Pandas* brings the ability to create 'DataFrames': "two-dimensional, size-mutable, potentially heterogeneous tabular data" (Developers, 2023c). These essentially allow us to represent a database in Python. Axes and columns can be labelled and *Pandas* gives a vast amount of tools for data handling, manipulation and analysis.

Matplotlib The *matplotlib* library offers great amounts of customisability over plots, with colours, line types, titles and annotations being altered with ease. *Matplotlib* makes for effective future development due to its straightforward implementation and clarity of code. One drawback is that the generated visualisations could be deemed less aesthetically pleasing when compared to other software available, however, the simplicity of the visuals was deemed visually effective in the context of a Notebook application. The dashboard is cohesive and puts information clarity above flashy visuals. This library is utilised for many of the graphs in the dashboard and the IPython (Jupyter) *magic command* '%inline matplotlib%' is used to display the graphs in the notebook. Real-time updates are supported using the animation module (Toker and Kuhn, 2019). Although this isn't implemented in the dashboard at present, it could be considered a limitation for future exploration.

Seaborn *Seaborn* is a higher-level library built on top of *matplotlib* and it focuses on creating more descriptive and attractive visualisations. Fit with built-in themes and colour schemes, including those suitable for the colourblind (Waskom, 2012-2022). It is in use for the graphs visualising distribution: Box Plots, Kernel Density Estimation (KDE) Plots and Violin Plots. *Seaborn* was selected for these as it provides much better visuals for these graphs than its underlying library.

Plotly *Plotly* is another visualisation library complete with interactive graphs and customisability. Arguably more complex than the previous two libraries, it is put to use in the dashboard to create Gauge widgets.

3.3 Fetching Key Data in Real-Time

To fetch data from the API, HTML requests are formatted with parameters and pull a CSV file from the server. Examples on the website(Observatory, 2023b) demonstrate the basics of preparing parameters and storing them in a Pandas' Dataframe. To best represent sensor data, the median aggregate data is in use for its less sensitive response to outliers than that of the mean. With an aggregate period of 15 minutes, matching the time period on the UO website, the frequency is enough to capture the complexity of the data while not containing an unreasonable amount of data for fetching and processing.

Other parameters include the start time, end time, variable and sensor type. The key variables for a sensor dashboard focused on air pollution are PM2.5 (Feng et al., 2019), PM10(Department for Environmental Food and Rural Affairs, 2023), and Nitrogen Dioxide (NO2)(Anenberg et al., 2022)(for Environmental Food and Affairs, 2023), as these are considered to be among the pollutants possessing the greatest risk to public health (UK Government, 2018).

Key Variables: PM2.5, PM10, and NO2 PM2.5 refers to particulate matter with diameter 2.5 micrometres and below, whilst PM10 refers to 10 micrometres and below. Due to their size, these toxins can enter the bloodstream and cause health issues. As such, "the Air Quality Standards Regulations 2010 require that concentrations of PM in the UK must not exceed: an annual average of 40 $\mu g/m^3$) (micrograms per cubic metre of air) for PM10; An annual average of 20 $\mu g/m^3$) for PM2.5", these limits should be represented in the dashboard to help visualise the current levels. NO2 also proves an example of the impact on daily life pollution can have, with *Anenberg et al* estimating that 1·85 million cases of paediatric asthma can be attributed to it globally in 2019 alone.

Real-Time Implementation and Data Availability The dashboard aims to be real-time to ensure data is as current as possible, to enable real-time trend analysis, efficient hot spot identification and effective intervention. One drawback of relying on the UO for real-time updates was identified immediately during agile testing. Frequent downtimes and maintenance acted as a consistent barrier to development as access to data was wavering and inconsistent. In response, upon being fetched the data is saved and stored in a local file, so that it may be utilised if more recent data isn't available. The file is named based on the variable and number of days being fetched, as such, when more recent data is successfully fetched the file will be overwritten as appropriate.

Presenting graphs with real-time data ensures they are as current as possible. While this may be in line with the UO website, it contrasts previous accessibility of sensor data in which the annual report (Council, 2022) would have been the primary method for public access to Newcastle's data - providing far from current information about the state of the

environment and restricting institutions from real-time trend analysis, efficient hotspot (areas of high pollution events) identification and effective intervention. Current pollution data can now be accessed on the *UK Air Defra* site (Air, 2023) as a CSV or as a graph presenting weekly data and although the site's implementation of *Highcharts* (Highcharts, 2023) is effective, the graph only demonstrates one sensor's data at a time - reducing accuracy, prohibiting spatial analysis and limiting the real-time analysis to a single area.

Time Period Flexibility The *AggData* class contains functionality for fetching data and handling attributes and is instantiated easily with a variable name and the number of days. The application strives to be flexible and detailed with the ability to alter the time period implemented. This is, however, restricted to 1, 3, 7 and 30 days to prevent compromising on speed and to fit with the necessity for saved CSV files. That is to say, giving unlimited freedom on the number of days would lead to saved files for every number of days and each would remain cached, slowing down the program and potentially leading to memory errors. Besides these technical limitations, there is also the consideration that too much flexibility would be a detriment to the application's usability, through the introduction of unnecessary complexity. An overview can instead be gained from the given time period choices, with the potential for updates in the future pending user feedback.

Units Another attribute in the *AggData* class is *units* for access by graphs so that axes can be easily labelled with variable and units, $\mu g/m^3$) in the case of PM10. This is vital for ensuring graphs are informative, especially to those who may not have previous experience with the elements being visualised. To take this further, an information tab could be implemented with descriptions of units, variables, and their importance - further increasing accessibility and inclusion.

3.4 Trend Visualisation and Analysing Distribution

The first implementation of visualisation utilises the *matplotlib* library to create line graphs to display the data frame. This graph is selected to be an effective method of visualising time series data (Wang et al., 2018) for trend analysis of pollution data over time. The colour is set to blue, so as to be easily visible to a colourblind audience as well as improving visibility. A more pastel shade of medium-cyan blue is chosen to produce a more sophisticated, less garish visualisation Stone (2006). The decision to opt for a more *moderate chroma* (intensity) than pastel makes the graph more visible while maintaining the subtle refined appearance of pastel colouring. A high saturation colour could potentially be paired with this on the graph to highlight outliers or other selected data in future.

When the line graph is tested with a period of 1 day and the variable PM2.5, it is revealed the number of outliers causes the graph to be illegible for identifying trends. This poses a dilemma, as the outliers are likely accurate and merely indicative of hotspots. Removing these outliers to improve the visibility of trends could be problematic. Important information could be lost which would make it more difficult to identify patterns. The number of sensors displayed also makes trend spotting difficult with so lines obscuring the overall 'shape' of the data.

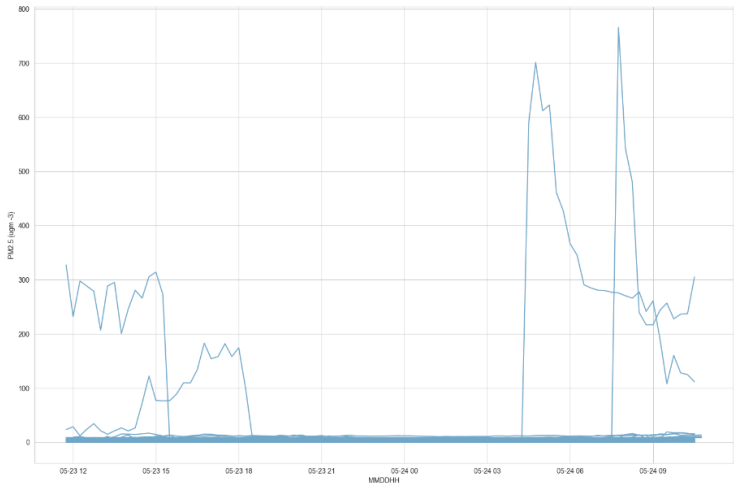

Figure 5: Extreme outliers presented on a line graph.

The lines are omitted to create a scatter plot, presenting trends more clearly as outliers, variance and noise aren't creating "disturbing lines" (Wang et al., 2018). Sensors are unlabelled due to little information being ascertained from sensor titles and the purpose of these plots being for time series analysis, rather than spatial analysis.

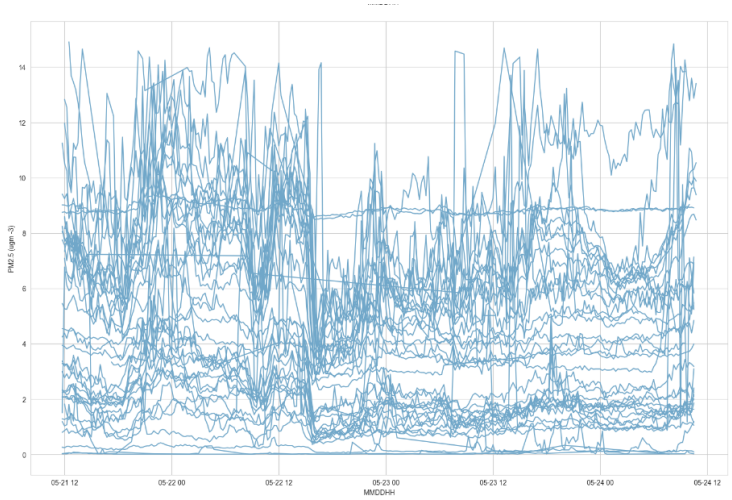

Figure 6: An overcrowded line graph of the past 3 days, outliers-removed.

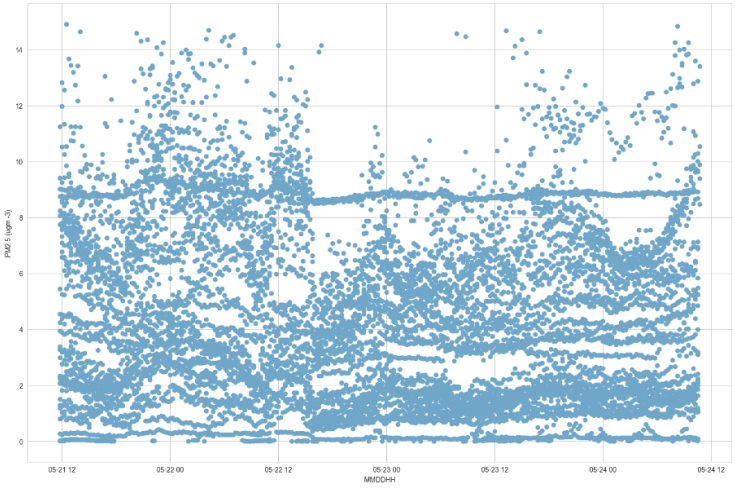

Figure 7: The same overcrowded graph, but with a scatter plot.

The *seaborn* library is utilised to create a number of graphs for distribution analysis.

Box Plot Generated to present the dataset with its outliers, this type of plot displays the distribution through the interquartile range, characterising the data with a median. The number of outliers is so large, however, that the plot is illegible. The interquartile range box and whiskers are

visible only as a line near the zero mark and the 'outliers' ranged from 15 to 1200. This variance is likely indicative of the spatial distribution and demonstrates the importance of spatial analysis when dealing with pollution data.

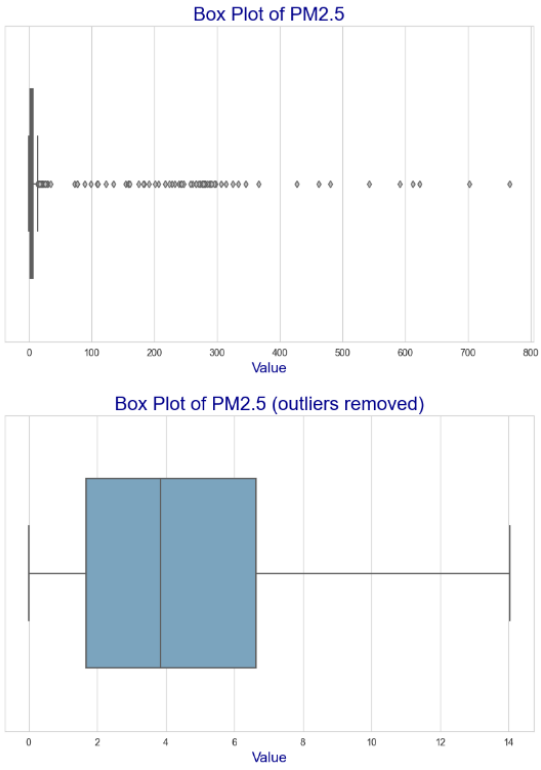

Figure 8: Box plots, with and without outliers.

Histogram Visualising the distribution of values through basic bars impresses simplicity and is an example of one of the more straightforward non-technical graphs on the dashboard. With clarity, the number of attributes with each given value is displayed with light blue colouring for easy viewing. While outliers are retained the graph appears unimodal, with only one peak in the data, but with outliers removed a bimodal pattern emerges. Making the assumption that the removal of the highest outliers eliminates those specific sensors entirely, it is likely this pattern denotes variation in pollution across different times, as is supported by the scatter plot. Histograms are likely the best-understood distribution plot (Scott, 2015) for the non-technical audience, but hosts drawbacks in the form of arbitrary offset due to the 'bin edge', or the location of the histogram block splits, rather than a continuous smooth curve of estimated density (Kelley, 2021).

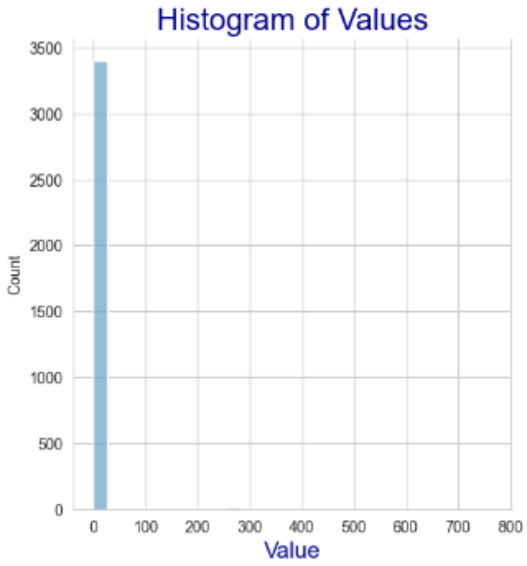

Figure 9: A histogram of the data distribution with outliers. See Results for an outliers-removed plot.

KDE Plot To analyse the distribution of the dataset further, seaborn is utilised to create a *Kernel Density Estimate* (KDE) plot. This applies a *kernel function* to each data point, the specificities of which are unimportant for the functioning of this dashboard, suffice to say it generates a smooth, continuous curve of the dataset distribution. As this removes the 'bin edge' offset it utilises the exact locations of each data point(Kelley, 2021) and provides a more accurate insight into the shape and peaks of the distribution as compared to the histogram.

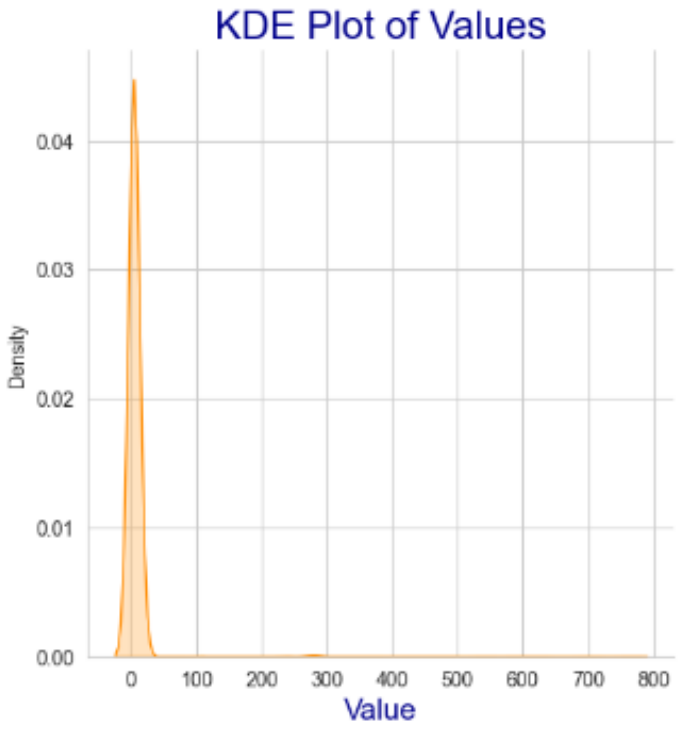

Figure 10: A KDE plot of the data distribution with outliers. See Results for an outliers-removed plot.

Violin Plot Bringing together the benefits of both Box Plots and KDE graphs, violin plots display both the interquartile range (as strings) and the probability density (as the body) of each value, even incorporating a white dot to denote the median. This is an effective display of the distribution as it visualises the 'breadth' of the distribution of each value, although it lacks the specific frequency data shown in the histogram.

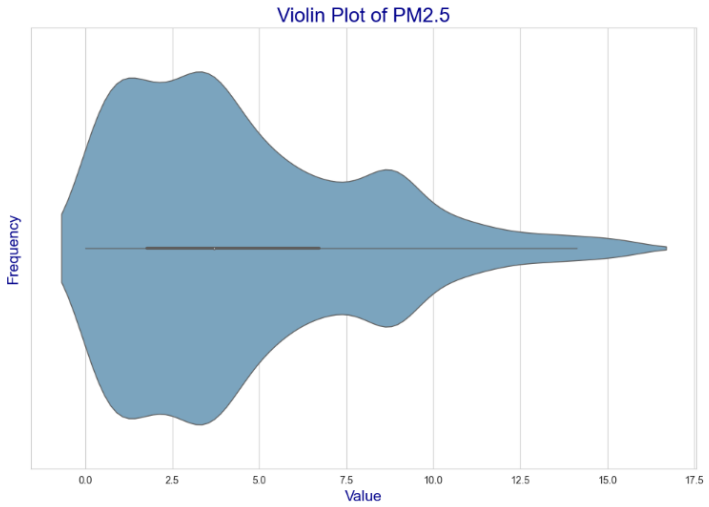

Figure 11: A violin plot of the data distribution, outliers-removed.

3.5 Handling Outliers

The generated distribution plots made it clear that a large number of values were zero or below and a small number of values were extreme upper outliers. As such these outliers need to be handled to ensure that the dashboard presents clear and accurate visualisations.

Remove Suspects One attribute in the dataset, 'Flagged as Suspect', proves useful during data sanitation as values which are likely to be false readings have already been identified by the sensor, enabling straightforward removal when preprocessing. In addition to this, values below zero were dropped as part of the 'Remove Suspect' function. This problem has been previously noted by van Zoest et al. (2018): 'Negative concentration values occurred when the concentrations were below the limit of detection and were removed from the dataset (1.5%). Zeroes in the data indicated a sensor failure and were removed from the dataset (1%)'.

Outliers Toggle An 'Outliers Removed' toggle box gives flexibility to the user on which data is visualised for analysis, enabling adjustment of visualisations as best suits them rather than limiting their options to only the processed data. Data accuracy is preserved, but the user can remove the outliers to provide a more 'zoomed in' view, to focus only on the core distribution of the data. Information should be provided on the impact of removing outliers to keep the dashboard accessible - both the method used and the impact this has had should be displayed.

Interquartile Range The 'Remove Outliers' function uses the interquartile range to cut out extreme 'outliers', removing data points which fall outside the first (0.25) and third quartile (0.75): or outside of the middle distribution (Vinutha et al., 2018). With this the visualisations become much more legible and it is likely many mistake values are also removed, as extreme outliers which may have represented mistakes have been eliminated. The plots present a clear representation of the majority of the data, functioning to display the trends, though it is true that the missing outliers could obscure data patterns beyond the middle distribution as well as hiding any drastic changes from sensors which usually record lower values: for example, if an anomalous yet non-erroneous spike in pollution were to take place.

Z-score An alternative method of removing outliers is z-score, "a standardized score given to a continuous variable expressed in terms of its relation to the mean" (Mowbray et al., 2019). The z-score has been said to be more effective at removing outliers(Chikodili et al., 2021), however, as the z-score uses the Mean and Standard Deviation to calculate outliers, the number of extreme outliers in these datasets it is likely that this would skew the results. The interquartile range could indeed lead to outliers in the middle distribution being maintained, however, the primary cause for outlier removal in this case is to improve graph readability.

3.6 Spatial Analysis: Visualising Hotspots

The ability to easily analyse spatial patterns is an example of something the UO website is lacking. To properly visualise these trends the dashboard incorporates a series of maps to plot upon the values for the corresponding sensors. The longitude and latitude properties are columns in each data frame and can be easily plotted upon an OpenStreetMap tile with the *smopy* library(Lars Yencken, Accessed 2023). This visual generates a map of sensor locations but no way of comparing the values of each location, short of perhaps implementing the websites' solution of each point generating its corresponding graph when clicked upon. To visualise and compare each location a new type of graph must be implemented.

Static Spike Map In an effort to expose hotspots or areas of significant pollution readings, the dashboard contains a Spike Map, in which 'spikes' of varying heights are plotted to denote intensity. This map aggregates the data by grouping elements with shared longitude and latitude, before taking the max value of that location and forming a *Geodataframe*. This is an object type from the *Geopandas* library (Geopandas Development Team, Year) formed of an attribute and a location. The maximum value is chosen as it demonstrates the 'worst case' for each location, focusing on visualising areas with the most significant pollution-causing 'events'. Although aggregation to take the max is unnecessary, with the same visualisation being created if spikes are stacked atop one another, computing unnecessary rows is avoided for efficiency.

The map visualisation is created and customised with *Pyplot* and the base maps are added through the *Contextily* library (Daniel Bastidas, Accessed 2023). This is set up with coordinates to focus on Newcastle and each row has its location plotted on the map with a 'spike' pointing upward - its length equal to the value scaled up by a given factor. The relatively 'high-chroma' colour choice of dark orange is chosen to represent the hotspots as it is suitable for users with colourblindness (Brewer, 2023), and its hue and saturation convey a sense of danger: orange is a 'hot' colour, considered to be "arousing" (Bartram et al., 2017). The visibility of the spikes upon their background is clear, and the distinctive colour ensures that the spikes stand out prominently to ensure ease of analysis. A minimum distance between spikes is enforced to prevent the graph from appearing cluttered

The biggest hotspots cause a problem as they can be vastly larger than all others, causing the map to zoom out and obscure the other data points. Displaying the hotspots is the purpose of the map, but scaling them down so that only they are visible is problematic as there is no data for comparative visualisation. Removing the highest values from the spike map would hide the most significant hotspots, however, which defeats the purpose of creating a map of max values. One option is 'winsorizing'(Duan, 1998), in which the most extreme outliers are replaced by smaller values to prevent the outliers from dominating the map.

Winsorization In this method, the interquartile range is calculated to identify outliers and all variables above or below the given percentile are substituted with that percentiles' value (Dash et al., 2023), preventing the outliers from holding too great an influence on the spikes in the graph. This may be effective with the real-time data in this application, as the values could change to much higher values than previously expected, but it somewhat pacifies the greatest hotspots, the presentation of which is the primary purpose of this graph. Although capped relative to distribution, an increase in value from one location wouldn't appear as substantial in the visualisation.

Logarithmic Scale The use of a logarithmic scale for the spike map allows for better visualisation of the lower end of values as upper values are scaled down. The overall range of values is compressed but similar data points become more discernible as the spread becomes more evenly split. A limitation of this method, however, is it can give an inaccurate impression of uniformity. This provides complications for interpreting values compared to a linear scale, it is noted that "if a logarithmic scale is used, it may be difficult for the user to interpret [and] the user should be sophisticated and familiar with the data" (Benson, 1997). This was a particular concern during the pandemic, as such scales have been noted to have caused "people [to] have a less accurate understanding of how the pandemic has developed" with the change of scale altering policy prefer-

ences (Romano et al., 2020). As the intended audience of this dashboard is those with a non-technical background it is crucial that accessibility be considered when making design choices for these visualisations and as a result, the logarithmic scale alone is not appropriate for maintaining dashboard simplicity.

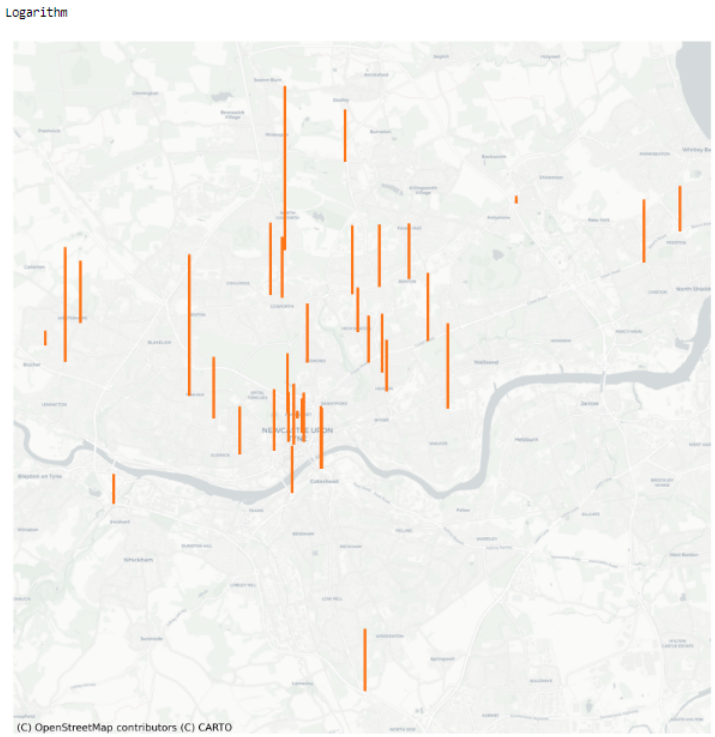

Figure 12: A static spike map of max PM2.5, 1 day, including outliers, logarithm scale.

Choropleth Map Another map type is included in the dashboard, in which sensors are plotted on the map with varying hues to denote intensity. The chosen colour gradient shifts from purple to yellow as the value increases, enabling straightforward comparison between different regions. This colour scheme, *plasma*, is considered "a strong candidate for accessible visualization" (Reda and Szafir, 2021) and presents the variance in a clear manner. The variance between each sensor is clear and patterns are easy to comprehend, but this graph type is not as effective as the spike map at visualising hotspots. While it is difficult to gauge the exact values for each sensor a rough estimate can be gauged by looking at the colour gradient guide on the side of the map.

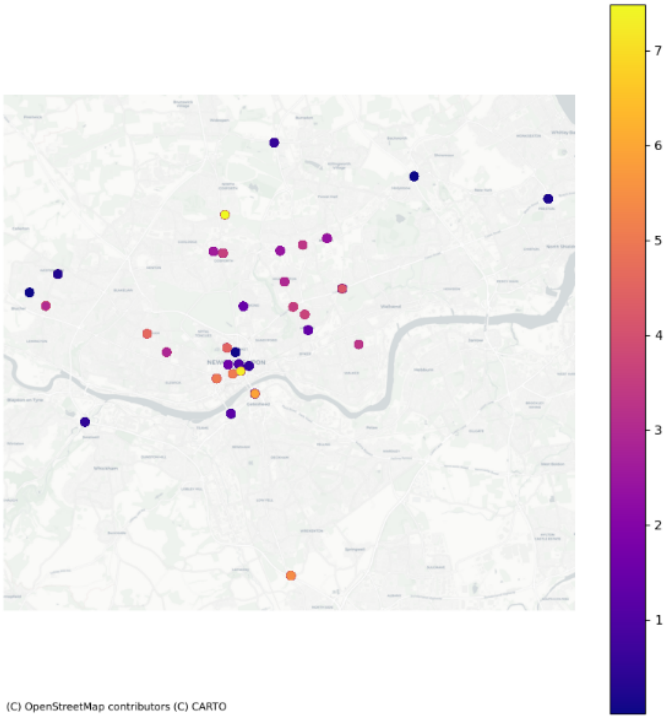

Figure 13: An choropleth map of PM2.5, 1 day, excluding outliers

Interactive Spike Map The *Folium* library allows data to be displayed on an interactive map, with the common example being for a choropleth map (Folium Development Team, 2013). The spike map is recreated with this tool, contrasting the static spike map, so as to allow the user to navigate around the sensors, changing view and zooming as suits their use. Interactivity provides the ability to drill down into a particular region or zoom out to give a more overall view. Another result of this is it's likely to lead to an increase in engagement (HUNDHAUSEN et al., 2002)(Kamburjan and Grätz, 2021) due to interaction with the visualisation. To improve load times when generating and refreshing the Folium map the *joblib* is implemented to support spike creation in parallel, utilising multiple CPU cores (joblib developers, 2023).

Colour has been utilised in this map similarly to the choropleth map, with spikes changing in brightness based on their values. This colour gradient shifts from light to dark orange to retain the sense of danger while still being colourblind-friendly. Changing just brightness, rather than hue, is effective here as the colour change is only simply to emphasise the strength of outliers, rather than being the sole measure of magnitude. One limitation in the static map is that it is not possible to read the specific values presented by each spike, as such for the interactive map a tool tip has been implemented allowing the user to hover over and read

the max value of each location.

Both maps are included in the dashboard, with the static map presenting the logarithmically transformed values and the interactive map the winsorized values (the most extreme 0.5% of values is altered to fit better). This is effective as, despite the high-value hotspots requiring a larger frame to visualise, the ability to adjust zoom enables the user to shift the focus of the visualisation toward specific values if they so desire, the hotspots no longer obscure the rest of the map. Further, any reduction in emphasis caused by winsorization on the interactive map is salvaged by the colour gradient emphasising hotspots. Meanwhile, the static map is able to fit all the logarithmic scale spikes in an appropriately sized frame, with all spikes easily visible. Maps are appropriately labelled and the incorporation of both prevents any misunderstanding caused by the logarithmic scale.

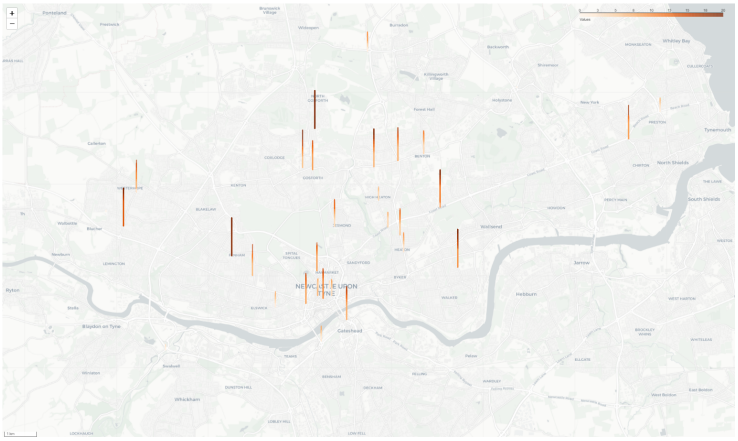

Figure 14: An interactive spike map of max PM2.5, 1 day, including outliers, winsorized.

3.7 Forecasting with Prophet

Forecasting future air pollution levels within the dashboard contributes to informing policymakers and helps act as an early warning to the event of pollution rising beyond safe levels, allowing for active measures to be taken where necessary to prevent this. Granting straightforward access to the non-technical community also helps to encourage innovation through public information being both accessible and comprehensible.

A regression algorithm is a form of supervised machine learning in which an attempt is made to predict a continuous value (Stulp and Sigaud, 2015). The algorithm is supervised when it has been trained on prior information in order to make predictions (Singh et al., 2016). The Prophet

library was created by (Facebook, 2023) and allows for straightforward prediction of a single variable over time (uni-variate time-series forecasting), fit with automatically generated visuals.

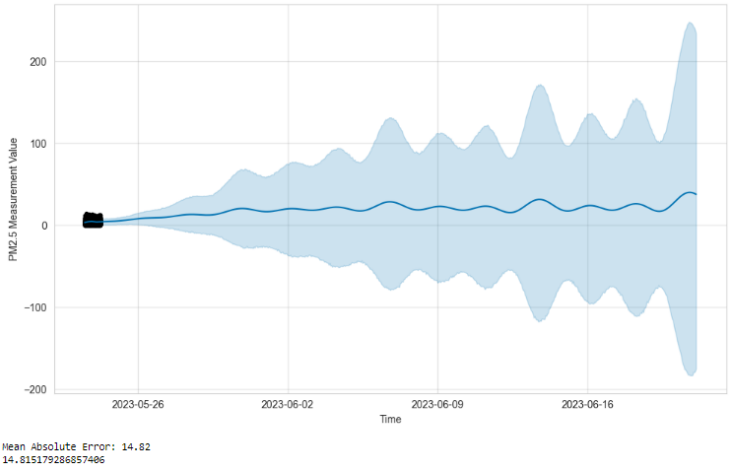

Figure 15: Prophet forecasting visuals, displaying general future trends.

This library was chosen for its 'out of the box' user-friendly visualisations and due to it being considered an effective choice for regression (Babu et al., 2022). In the context of this dashboard, it is demonstrating the potential for advanced machine learning techniques to be applied in real-time and presented as a high-level visualisation understandable to a non-technical user. Prophet has features for outliers, missing data and can also be extended to ignore holidays (Kumar Jha and Pande, 2021) - something which would be useful for air pollution as certain holidays, such as New Year's Eve, could be considered outliers due to the use of fireworks increasing variable measure considerably (Tanda et al., 2019). Hyperparameters are tuned in the implementation as follows. The 'seasonality-mode' is set to multiplicative to anticipate seasonal fluctuations due to holidays, and large changes in trend across the year are accommodated by setting the yearly seasonality order to 10. 'changepoint_prior_scale' was also set to a high value of 0.5 to give the model flexibility in the case of sudden significant changes due to industrial development or other large changes.

This is implemented to demonstrate the capabilities of the dashboard and datasets to include machine learning algorithms. The implementation takes one variable, trains the model with 75% of the values and generates a visualisation of predicted future values. The remaining 25% values are used to test the model's accuracy. The *mean absolute error* is generated with the *sklearn* library and displayed to describe the regression loss: how close or far the predictions are to the true values. The mean absolute error

was selected over the *Root Mean Square Error* (RMSE) due to the RMSE being more effective with normal distributions (Chai and Draxler, 2014) and air pollution, as demonstrated by the KDE graphs, does not follow a normal distribution with the graph showing many outliers. The inclusion of multiple variables may lead to more accurate forecasting and could potentially reveal new correlations between variables. An example of an effective secondary variable for forecasting could be weather data such as wind speed, as there is a strong correlation between wind speed and air pollution such as PM10(Cichowicz et al., 2020).

To extend the machine learning element of this dashboard in future, the implementation of Random Forests for regression could provide more robust and accurate predictions for air pollution variables. Random Forests uses a series of decision trees to train the model and "performs more effectively in general terms"(Papacharalampous and Tyralis, 2018). As it can utilise multiple features simultaneously it could make use of a plethora of correlated variables to seek out the best combination for prediction. Users would benefit from more accurate forecasting and previously unnoticed correlations could be uncovered.

3.8 Bringing it All Together: Dashboards and Widgets

To compile the graphs into a dashboard the *ipywidgets* (Project Jupyter, 2023) library is utilised with the '%matplotlib inline%' magic command forming the graphs and widgets into the notebook. The dashboard is navigated through tabs along the top (as inspired by the Bristol dashboard) to switch between graph types. One or more drop-down menus are available to switch which variables are being analysed. Some tabs, such as scatter plots, allow for two graphs to be visualised to allow for comparisons and correlations to be drawn between the different pollution types.

Gauges The *Numpy* library mathematics library is utilised to find the mean of an entire column and this is presented as a gauge graph (or dial) for the purpose of immediate and clear comprehension of the 24-hour average for the most critical variables. This provides an easy-to-interpret visualisation of a single statistic in real-time(Mohammed et al., 2022). It is a very recognisable and intuitive visualisation making it a suitable choice for a non-technical audience. Research conducted for car gauges (François et al., 2019) broke the types of gauge available down into shape, indicators and directions - with the findings being that circular, horizontal gauges with pointed indicators where necessary for reading were the most effective for vehicles to ensure safety.

Although this is a different field to the gauge chart, the shared aim is ease of interpretation, however, a dashboard focused on presenting air pollution data also has some other aims. For example, a car gauge is not required to be aesthetically pleasing, on the contrary, it is critical that the gauge not be distracting for the driver. As such its simplicity need not

be replicated in this setting. Plainness is unnecessary and a detriment to the breadth of data that can be conveyed through the use of colour. The gauges are circular and horizontal. Generated with the *Plotly* library, the graph is instead split into two colours, with a third darker colour moving along the outer circle denoting 'fullness'. Pointed indicators are replaced by this and the centre of the gauge displays the exact value for clarity.

The safe and unsafe values are set corresponding to the Defra figures for safe 24-hour average(for Environmental Food and Affairs, 2023)(Department for Environmental Food and Rural Affairs, 2023). Effective use of colour here is crucial. Green and red were considered to be used for intuitiveness, representing the safe and unsafe values respectively. What must be considered, however, is that colours used should also be effective for users with colourblindness (Midway, 2020). A medium-cyan blue and dark orange is employed as an appropriate substitute, with medium-cyan representing safe values and the more harsh yet vaguely complimentary dark orange intuitively representing danger. These two colours were also chosen to match the line and spike graphs respectively. Purple to represent the danger was avoided to prevent confusion when being compared to the choropleth map, in which purple represented lower values. The use of orange in the spike map representing hotspots also fits with the sense of danger for this gauge.

The location in the bar at which the dangerous value began was considered - based on the precedent of standard safety gauges long in place in various fields, 2/3 safe and 1/3 unsafe was deemed intuitive, however, this may prove limited if the reading were to go beyond the values plotted on the graph. Furthermore, the intention of the graph is to give the impression of danger as necessary, and having a smaller 'safe' segment conveys the small amount of pollution desired, rather than seeming as though there is space for lots more pollution before it becomes unsafe. The graphs are given a sense of imminence and have the desired impression of danger achieved.

Figure 16: Gauges demonstrating 24 hour averages of two key variables: PM2.5 and NO2.

Time Period and Down-sampling Down-sampling is a technique used to reduce the amount of data (granularity) so that complexity can be reduced (Burchi and Vielzeuf, 2021). This is useful for reducing the amount of data points being processed and visualised when dealing with the larger time periods of 7 days and 30 days. For 24 hours, the aggregate period of 15 minutes is deemed suitable, but on scaling the data visualised to 30 days the high granularity restricts the clarity of the graph. On top of this, the large number of data points being processed increases the load times of the dashboard drastically.

Downsampling also helps to reduce the number of outliers, as the most extreme values are smoothed out. Although the mean function is used within each sample, the initial median aggregation to 15 minutes meaning the smoothing process is already given a head start. As such care must be taken when considering which graphs utilise the downsampled data. Graphs presenting data for trend analysis, such as the scatter plots, are suitable as the reduction in high outlier values is not to a detriment of analysis, but for spatial analysis plots such as spike maps, downsampling the data could prove problematic. Due to the maps only presenting the max values for each location, it is not necessary to downsample as the

same amount of data is plotted regardless, and doing so would hinder the accuracy of the graphs.

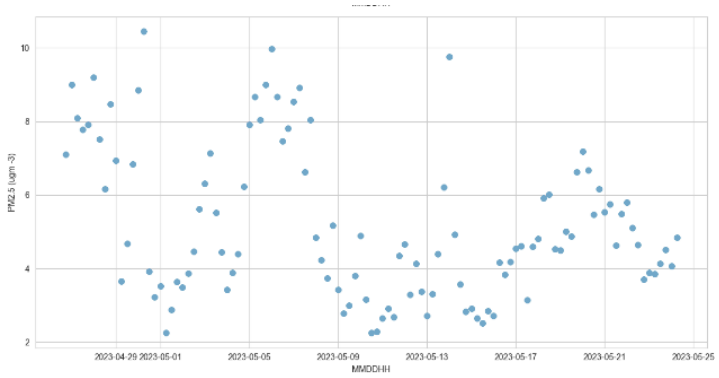

Figure 17: Scatter plot showing the previous month of PM2.5, downsampled to 6 hour resolution. Notable spikes occur on days where football matches occur, e.g. 7th, 18th and 22nd of May.

Scale Factor A 'scale factor' slider widget is implemented on spike map tabs to allow the user to adjust the scale multiplier of the spikes. This is a necessity due to the real-time nature of the application. What is an appropriate scale factor one day may be ineffective the next, with the shape and scale of the values potentially changing considerably. The scale factor selected is visible to allow the user to always use their preferred scale factor if they find an effective value and this will allow for better ease of day-to-day comparison. An alternative approach could be to alter the scale factor relative to the range of values being displayed on the map, but this flexibility grants greater power to the users and is relatively easy to understand. The slider widget was selected for this to give the user greater control over the value and to maintain the tidy dashboard look despite the number of widgets for adjusting graphs.

Spinner Visualising loading when the dashboard is updating as a response to user input acts as feedback and gives assurance that the application is working, with the graph not yet having been updated to its selected settings. Whenever a change is made to a widget this is observed and the loading widget will appear until the update function has finished running. HTML/CSS was utilised to create the loading spinner as it is more visually appealing than the Python loading bars and as the complexity of the different updates means that the progress of the functions is difficult to track linearly - that is to say, implementation of an accurate progress bar would be difficult, but 'lying' or estimating would be feasible. The implementation of a progress bar may make users more willing to wait, however, as the updates are relatively short a spinner is an appropriate choice - especially a fast-moving one which gives the perception of a shorter wait time (PIBERNIK et al., 2018). The spinner was selected

and placed appropriately to allow continued access to the graph and to ensure it is clear when refreshing is taking place.

Figure 18: Spinner, displayed when loading.

3.9 In Summary

In summary, after deciding upon the tools that the dashboard would use the graphs were generated and compiled into a dashboard with widgets to provide customisability and feedback. Graphs appear on the right and navigation is available through tabs and drop-down menus. The displayed graphs straddle the line between simplicity and complexity, presenting a variety of graphs with vast amounts of data, yet all while taking care not to be overcrowded or illegible. Preprocessing is utilised to ensure the graphs are presentable while maintaining accuracy, and that false readings and outliers are removed. This methodology describes the majority of the process, but without going into too great detail about the low-level features and specific inner workings. Large amounts of the agile process of testing and decision-making have been ignored for the sake of brevity and many of the algorithms and libraries utilised have been treated as 'black-box' as describing them beyond their use in the dashboard is beyond the scope of this paper.

4 Results

4.1 Dashboard Application

Here the resulting application is described in detail, alongside some of the more apparent findings presented by the graphs for the sake of evaluating the success of the dashboard as a tool for visualisation and basic analysis by non-technical users.

The application is split into two, with the left side presenting the graph and the right hosting the two gauges showing the 24-hour averages for PM2.5 and NO2. These switch from medium cyan to dark orange, representing safe and unsafe values respectively with the bar filling up with black to signify where the current value is on that scale. In the centre of the gauge, the current value is displayed in large characters for clarity

On initialisation, a loading widget is briefly displayed as the tabs are cycled through to generate each graph. Along the top of the main screen are tabs containing each graph type: scatter graphs, distribution analysis, static and interactive spike maps, choropleth maps and Prophet forecasting. These are organised in a logical way to allow for sensible navigation vaguely from simple to complex, with trend analysis, distribution analysis, spatial analysis and forecasting grouped together.

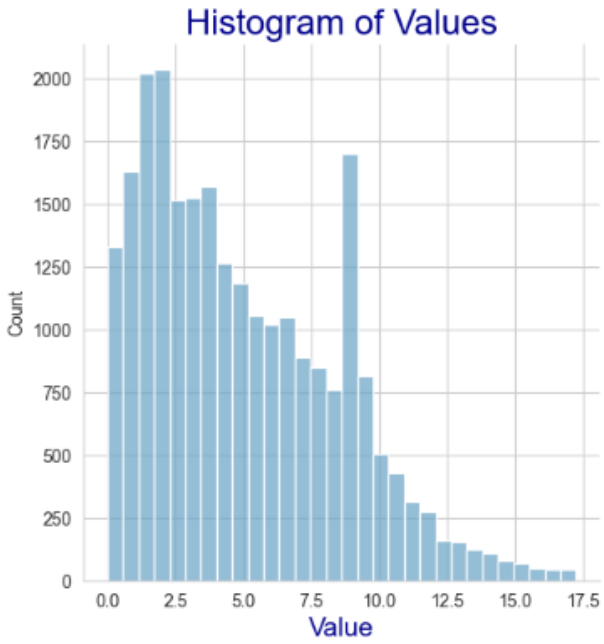

Figure 19: Histogram Distribution, outliers removed.

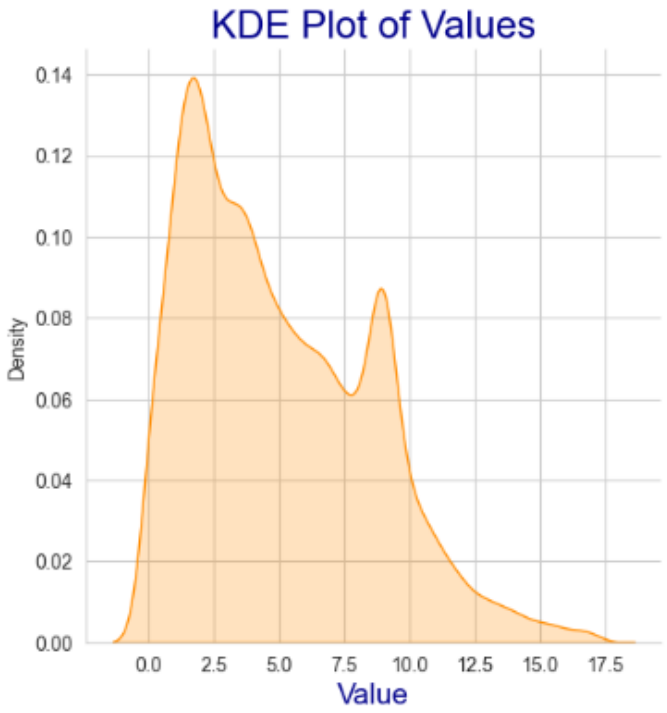

Figure 20: KDE Distribution, outliers removed.

Once on a tab, drop-down menus are available for variable selection. Information is presented as to what is currently being shown on the graph, for example, "Outliers Removed with Interquartile Range" and "Down-sampled to 2-hour resolution." Further features such as a checkbox widget for removing outlier values and a slider to alter the time period give the user extensive control over the data being visualised, a level of customisability which allows for more effective exploration and analysis.

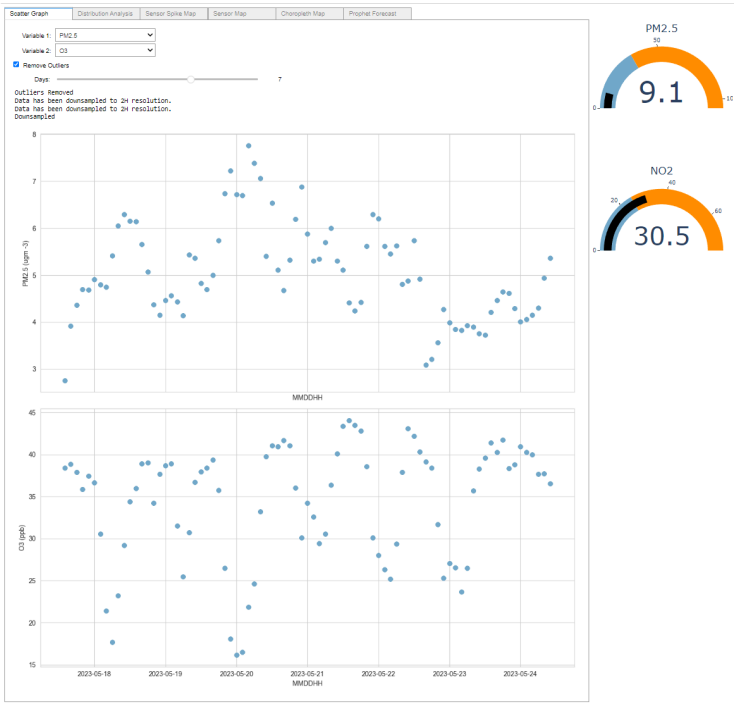

Figure 21: Landing page of the dashboard.

When changes are made to the visualised graph, and on initialisation, the loading spinner widget is displayed and text is output describing what dataset is being fetched. If a dataset is fetched from local storage, due to the UO API being down for maintenance, a message describing this is shown. Changes to any of the widgets results in an immediate update of the graph and once complete the loading widget and text is cleared.

The widgets present for graph adjustment include the 'days slider', which can be set to 1, 3, 7 or 28 days as desired and each alters the downsampling on the data to either 15 minutes, 1 hour, 2 hours or 6 hours respectively. The 'Remove Outliers' box is checked by default, outputting clearly that outliers have been removed. This is to emphasise the readability of data and reduce outliers, although the more downsampling that takes place the less influence outliers have over the data due.

The scatter plot graph displays two plots for straightforward comparison. The axes are labelled: with the X axis displaying the dates in 'YYYY-MM-DD' (or MM-DD-HH in the case of fewer days) fashion to meet global standards and the Y axis concatenating the variable name and the appropriate unit as fetched from the dataset. This could be made more clear by altering the time scale to better fit real-world standards, for

example, having the time in HH: MM format for the 1-day time span. The distribution tab compiles a box plot, a histogram, a KDE plot and a violin plot. The presence or lack of outliers is clearly listed in the emboldened blue title and the axes are clearly displayed in a similar typeface.

On spike map tabs, a scale slider is available to alter the scaling multiplier on the spike size. This is useful for identifying hotspots and seeing in greater clarity the variation between different data points. Where colour is used on the interactive spike map and choropleth map, a map of the gradient is displayed with numbers at regular intervals for ease of reading.

4.2 Findings

Some examples of findings upon brief inspection of the visualisations.

One might presume the city centre to be host to greater readings of PM2.5 and NO2 due to the amount of traffic, especially with "33% of Nitrogen Oxides (NOX) emissions and 14% of Particulate Matter (PM2.5) emissions" coming from transport (Department for Transport, 2022). Although while idling, "gasoline vehicles emit a minimum amount of nitrogen oxides (NOx) and negligible particulate matter (PM)" but produce large amounts of "carbon monoxide (CO)", however, "NOx and comparatively larger PM are produced by diesel vehicles during idling"(Shancita et al., 2014). As the city centre is a 'Clean Air Zone' (Newcastle City Council, 2023), older Taxis, buses and HGVs now have to pay to enter - the likely outcome of which being a reduction in traffic or high emissions vehicles. Traffic is, of course, not the only cause of pollution, and notably, road traffic is not the only source of vehicle emissions in Newcastle. 12.4 Kilometres Northwest of the city centre lies the largest airport in the North East: Newcastle International Airport.

Research by Schlenker and Walker (2015) notes that "although the aircraft engine is often idling during taxi-out, the per minute CO and NO x emissions factors are higher than at any other stage of a flight" and that "Los Angeles International airport is estimated to be the largest point source of CO emissions in the state of California, the second largest of NO x." The study uses 10 km as the radius for what they consider to be 'close' to an airport and with the Gosforth sensors sitting around 5 km away (University, 2023) (Google, 2005) it is likely this is the cause of the significant increase in NO2 measurements at this location. Another potential cause could be the abundance of industrial estates in the area.

Notable peaks are visible when analysing weekly and monthly data with considerable increases in pollutants occurring during game days. This is likely due to the influx of traffic in the city due to people visiting the city. As the stadium seats 52,000 (Grounds, 2023) this leads to a large increase in population and traffic on these days.

5 Discussion

5.1 Limitations

Deployment The primary limitation of this app currently is that it only runs in Jupyter inline and as such is not currently suitable for deployment. To break beyond this limitation an HTML document could be generated, containing the dashboard and widgets. Alternatively, a framework such as Flask could be used to create a web application containing the matplotlib graphs for deployment. *React-JS* (Meta, Accessed: 2023) is a JavaScript library designed for web interface creation and is capable of creating effective one-screen dashboards. This would make it a suitable choice for future development of this application, benefiting from more up-to-date and flashy visuals, and improving user experience.

D3js (Bostock et al., Accessed: 2023) is a popular JavaScript library for the creation of interactive visualisations. This could be an effective choice for making graphs more dynamic and interactive - grating the user the power to hover over data points for more information and even for straightforward implementation of animations and transitions. Touch events would be a great improvement as they would improve user engagement, keeping user attention more than that of a static plot. The *mpld3* library combines *matplotlib* and *d3js* to create "a simple API for exporting your matplotlib graphics to HTML code which can be used within the browser, within standard web pages or the IPython (Jupyter) notebook" (Vanderplas, Accessed: 2023). This would allow for the *matplotlib* dashboard to be deployed with an appropriate web framework for public use.

Aesthetics Many of the plots created are created in *matplotlib* and haven't been expanded upon greatly beyond the basic implementation of graph visuals. Though this is done to prevent detracting from the data itself and to ensure the visuals don't threaten the simplicity, it is also true that this simplistic effect could be retained while improving the aesthetics. One solution could be the use of Power Bi. Touched on in the methodology section, Power Bi is capable of generating powerful visuals as well as fetching data from API. Though the limitations of fetching data would need to be overcome, and although it is subjective and would require user feedback, it is also true that the visuals might be considered more pleasing than the basic ones generated by *matplotlib*. Outside of this, a sample of users from the target audience could be polled for their thoughts on the visuals generated by React-JS with the *d3js* library or the *chartsjs* (Contributors, Accessed: 2023) library.

Outliers The challenge of finding an appropriate method of presenting data with outliers remains. For the purpose of this dashboard, the 'Remove Outliers' option has been implemented, but for future development,

alternative solutions could be explored. One option could be the implementation of 'Grouped Box Plots' with 'zones' defined to separate the city centre from the outskirts. This would provide summaries of each area but would rely on strong correlations between zone and values. Analysing the spike maps it becomes apparent that this could be problematic, due to the outliers not being grouped together as they might be on a larger scale which might include rural areas readings. Grouping together the zones which usually present high values could make it more clear when a spike occurs elsewhere but this could risk obscuring patterns within those areas especially if anomalies occur.

Interactivity Many of the charts in this dashboard lack interactivity. Although some graphs contain tooltips and one spike map is interactive, the plots are generally limited in their responsiveness in terms of user interaction. More tooltips would enable greater quantities of information to be displayed without clutter, such as hovering over lines to display the sensor names or hovering over a data point to present its value similarly to in the spike map implementation.

Real-time Graph Updates Real-time graph updates as supported by *matplotlib* would allow for graphs to display new data as soon as it becomes available from the API, providing more up-to-date insight to users without the need for reloading the graph. The need for this is questionable and could be open to further investigation by seeking feedback on how users are using the application - this could be in the form of a feedback survey. One use for real-time updates might be if the dashboard was desired to be left running for constant information - such as in a widget. In this case, running an update every 15 minutes would suffice for the purpose of air pollution data.

Interquartile Range This method of identifying outliers removes extreme outsiders but doesn't consider those which might stray from the general trend. However, as the distribution of air pollution data doesn't follow a normal distribution and can be quite random, these outliers that prove difficult. The term 'Remove Outliers' for the widget could be considered inaccurate, as it implies all outliers are removed, when in fact only extreme values outside the IQR are identified and removed. This limitation could be improved upon, given more time, through the development of a more robust outlier identification scheme, one example would be median absolute deviation, using absolute deviation around the median to detect outliers. This has been shown to be effective at removing outliers in asymmetric datasets due to the presence of extreme outliers (Leys et al., 2013), similar to those presented in this dataset.

Loading Widget The loading widget could be improved upon by using a progressive loading bar for longer loading times, such as initialisation. In the implementation of this dashboard, the progress is somewhat shown by the dashboard cycling through tabs and loading each page, however, the application may benefit from a linear 'progress bar' style loading widget for this - as users are more contented to wait when these are displayed.

Memory Errors The potential for memory errors is a current limitation in the system. When fetching data, the current instances of the AggData class are checked to ensure duplicates aren't created, but all new fetched instances are retained in this array, which could potentially lead to a memory error if enough data were accessed. Python's *'Garbage Collection'* won't clear up the instances (Devguide, 2023), as the *instances* list attribute is referencing them. This limitation could be solved in future by accessing the CSVs in local memory as needed at the expense of speed, or by only retaining the most recently cached data frames.

Information and Links The application would benefit greatly from an 'Information' tab, not only to explain each of the graphs so that they may be better understood by the non-technical user but also so that the public can be more informed on air pollution and its impact. For the general public to understand the gravity of the information presented and to make the best use of this tool they need to be informed on the variables that the graphs are presenting and what that means in real terms. For future development, terms used in the dashboard should be clearly defined with accompanying links to websites where the user can find out more. Besides this, an '*i*' widget should be added to each graph with a description of what it shows and a guide on how to read it.

5.2 Design Approach

The approach taken focused on ensuring that the application would be suitable to its target of a non-technical user, either one who without the graph couldn't access all the data presented or one who doesn't want to. The application needs to be *intuitive* and not assume a prior level of understanding. As such, the graphs should be relatively straightforward with few advanced graphs. A user should be able to understand the visualisations at a glance, with widgets describing metrics such as colour gradients and how they relate to the values.

Visualisations took a simplistic design to ensure readability and appropriate colours to suit any users with colourblindness, again to increasing accessibility. The *Colorbrew* tool was used to examine appropriate colour maps and great consideration was given to the clarity of visualisations in

terms of colour saturation and hue. The background colour was considered to complement this, with map colours being adjusted to ensure data points stood out.

This approach could be improved, however, with regular user testing added to bolster agile testing, taking the guesswork out of decision-making and backing up choices made during development - or if need be, altering decisions that were made. By highlighting issues in development with real-world feedback from the target market identifying what is suitable for a non-technical audience would be made considerably easier and solutions may present themselves in the form of user feedback. This kind of feedback loop would allow for iterative improvements to the design, leading to improvements in accessibility and user engagement.

Though the security of the design was considered, given the nature of this application it is of little concern. One issue might be the increase in traffic to the API, however, given the data is already publicly accessible this is of little consequence. The dashboard was also designed with scalability in mind. More tabs can be added with ease to implement more graphs. One issue with this would be the increase of load times, however, as all tabs are loaded at login. Despite this oversight, load time was considered strongly during development, with aggregation where possible to reduce data points (as with the spike map) and implementation of parallel computing for particularly computationally expensive processes where possible, for example with *joblib*.

5.3 Application Evaluation

This dashboard improves upon the existing UO website visualisations by providing a more comprehensive overview and succeeds in presenting straightforward 'at a glance' visuals. The use of spike maps and choropleth maps is particularly effective as it allows for comparisons to be drawn between different locations with ease. The addition of basic forecasting also draws upon the successes of the UK Defra Air website to enable early warning signs of dangerous pollution levels with some degree of accuracy. Taking the success of intuitive navigation tools such as tabs from the Bristol Dashboard, this dashboard adds an extra level of depth: not only in the form of additional graphs but with the many extra sensors that the UO gives access to.

Overall, the dashboard is effective as a demonstration of how a dashboard could be created to extend the UO website's accessibility and widen both the audience and the information readily available. Despite limitations in the visuals the application is functional and displays the impressive amount of information that can be accessed with some programming knowledge and the aim to bring these tools to non-technical disciplines is both noble and achievable. Given that this application has yet to be deployed it should be evaluated as a prototype, and under by this metric it is a successful implementation of the API into an informative and

straightforward application, pending the addition of some interactivity and more up-to-date visuals, would make for an effective web app.

The application holds great potential for future development and provides successful examples of how machine learning could be implemented to further expand upon the complexity and accuracy of forecasting. Due to the focus on scalability, more variables can easily be added to this end, with the UO's weather sensors easily serving to improve predictions, particularly if another API for future weather data were implemented. Great control over the data presented is granted, with the many widgets and quick reload times leading to effective customisation of the graphs. With the implementation of tooltips and animations, the user experience would be improved but this doesn't detract from the basic foundations being well established for a user-friendly air pollution dashboard.

Urban Observatory Sensors One limitation inherent to this application is the reliance on UO sensors. As demonstrated in methodology, there are issues with the API being down regularly for maintenance. These downtimes seemed random and sometimes lasted several days. Although this was overcome by updating locally stored CSVs, this clashes with the real-time goal of this dashboard. Further, as noted by van Zoest et al. (2018), low-quality urban sensors are more prone to erroneous data than conventional monitors. Though the Bristol Dashboard mentioned earlier only hosted a limited number of sensors, these were high-quality continuous monitors. Though each sensor can be independently checked on the UO website, this shows only the 'pod's' history. Some pods are noted to be due to be removed due to issues with reliability, implying quality control, but it is not clear what type of sensor each is for the purpose of assessing quality, though this could be due to the air quality tools page returning 'not found'.

For the purpose of this project, the sensors are assumed to be accurate. Though this reliance on these sensors should be noted, this application is merely a front end for the hard work, cost and effort of the UO and without this incredible wealth of public and free resources this project would not have been possible, nor would it have been inspired if not for the work put into making the website and API an effective and (mostly) painless data broker.

6 Conclusion

This project's aim was to bridge the gap between data collection and comprehension by creating a user-friendly dashboard for presenting Urban Observatory environmental sensors with a focus on air pollution data. This was achieved to some extent, in that the created application acts as an effective prototype for an air pollution dashboard to be deployed on the web and demonstrates how this could be implemented. This is a deviation from the initial aim but nonetheless serves as an important

step for the creation of such a tool. A variety of static and interactive graphs such as scatter plots, violin plots, choropleth maps and spike maps present data for trend, distribution and spatial analysis, with a prophet tab also present for basic forecasting - signposting how to further machine learning could be implemented in future. Although these visualisations are functional, they could be improved with an increase in interactivity and attractive visuals led by user feedback.

The primary objective was to create visuals which can be understood by non-technical users for the purpose of making sensor data more accessible. The generated graphs are of varying complexity, though not all would be understood by those unfamiliar, they predominantly can be interpreted with relative ease - especially with the future implementation of an information tab on how to read graphs to widen the availability of information further. In terms of technical analysis, the implementation of the sensor resource into meaningful data for analysis demonstrates what can be achieved and the potential for innovation would be made even greater by future development into a publicly accessible web application. Balancing complexity with usability was a key element of development and should serve as the foundation of future design development. Developing this app in Jupyter not only served to improve programming skills but also to develop a strong understanding of what it means to create meaningful visualisations: from processing data to balancing design choices and fine-tuning graphs and forecasting for accuracy.

To conclude, serving both as a valuable learning experience and a strong example of the potential of the sensors as a public tool, the development of the UO Dashboard demonstrates the need for effective use of data. Mitigation of air pollution requires the entirety of the population to be on board, and by keeping the general public 'in the dark' behind barriers of complex data and technical prerequisites, the potential for collaborative innovation is cut down. Vast amounts of environmental data are available, but the public needs the right tools to make use of it.

References

Cities: Major contributors to climate change. United Nations, 2018. Retrieved from `https://www.un.org/en/climatechange/climate-solutions/cities-pollution`.

U. Air. Defra uk air information resource. `https://uk-air.defra.gov.uk/`, 2023. Accessed: 09 May 2023.

S. C. Anenberg, A. Mohegh, D. L. Goldberg, G. H. Kerr, M. Brauer, K. Burkart, et al. Long-term trends in urban no2 concentrations and associated paediatric asthma incidence: estimates from global datasets. *The Lancet Planetary Health*, 6(1):e49–e58, 2022. doi: https://doi.org/10.1016/S2542-5196(21)00255-2.

M. A. Babu, M. M. Ahmmed, M. A. Helal, and M. A. Hoque. The fbprophet forecasting model to evaluate the spread of covid-19 pandemic: A machine learning approach. *Journal of Interdisciplinary Mathematics*, 25(7):2073–2082, 2022. doi: 10.1080/09720502.2022.2133234. URL https://doi.org/10.1080/09720502.2022.2133234.

L. Bartram, A. Patra, and M. Stone. Affective color in visualization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 1364–1374, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346559. doi: 10.1145/3025453.3026041. URL https://doi.org/10.1145/3025453.3026041.

J. Benson. Visualizing data for environmental analysis. *Technical Report*, 4 1997. URL https://www.osti.gov/biblio/463608.

E. Bisong. *Matplotlib and Seaborn*, pages 151–165. Apress, Berkeley, CA, 2019. ISBN 978-1-4842-4470-8. doi: 10.1007/978-1-4842-4470-8_12. URL https://doi.org/10.1007/978-1-4842-4470-8_12.

M. Bostock, V. Ogievetsky, and J. Heer. D3.js: Data-driven documents. https://d3js.org/, Accessed: 2023.

C. A. Brewer. Colorbrewer 2.0: color advice for maps, 2023. URL https://colorbrewer2.org.

A. Bris and B. Lanvin. By ranking - IMD Business School, IMD. Smart City Observatory, 2021. URL https://www.imd.org/globalassets/wcc/docs/smart_city/smart_city_ranking_2021.pdf.

O. D. Bristol. Open data bristol, 2023. URL https://opendata.bristol.gov.uk/. Accessed: 2023-05-22.

M. Burchi and V. Vielzeuf. Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 8–15, 2021. doi: 10.1109/ASRU51503.2021.9687874.

S. Cao, Y. Zeng, S. Yang, and S. Cao. Research on python data visualization technology. *Journal of Physics: Conference Series*, 1757: 012122, feb 2021. doi: 10.1088/1742-6596/1757/1/012122. URL https://doi.org/10.1088\%2F1742-6596\%2F1757\%2F1\%2F012122.

T. Cepero, L. G. Montané-Jiménez, and G. P. Maestre-Góngora. Data visualization guide for smart city technologies. In F. Ortiz-Rodríguez, S. Tiwari, M.-A. Sicilia, and A. Nikiforova, editors, *Electronic Governance with Emerging Technologies*, pages 176–191, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-22950-3.

T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7(3):1247–1250, 2014.

doi: 10.5194/gmd-7-1247-2014. URL https://gmd.copernicus.org/articles/7/1247/2014/.

N. B. Chikodili, M. D. Abdulmalik, O. A. Abisoye, and S. A. Bashir. Outlier detection in multivariate time series data using a fusion of k-medoid, standardized euclidean distance and z-score. In S. Misra and B. Muhammad-Bello, editors, *Information and Communication Technology and Applications*, pages 259–271, Cham, 2021. Springer International Publishing. ISBN 978-3-030-69143-1.

R. Cichowicz, G. Wielgosiński, and W. Fetter. Effect of wind speed on the level of particulate matter pm10 concentration in atmospheric air during winter season in vicinity of large combustion plant. *Journal of Atmospheric Chemistry*, 77(1–2):35–48, 2020. doi: 10.1007/s10874-020-09401-w.

C. Contributors. Chart.js: Simple yet flexible javascript charting library. https://www.chartjs.org/, Accessed: 2023.

N. C. Council. Air quality annual status report (asr), 2022. Accessed: 09 May 2023.

Daniel Bastidas. contextily. https://pypi.org/project/contextily/, Accessed 2023. Python Package Index.

C. S. K. Dash, A. K. Behera, S. Dehuri, and A. Ghosh. An outliers detection and elimination framework in classification task of data mining. *Decision Analytics Journal*, 6:100164, 2023. ISSN 2772-6622. doi: https://doi.org/10.1016/j.dajour.2023.100164. URL https://www.sciencedirect.com/science/article/pii/S2772662223000048.

Department for Environmental Food and Rural Affairs. Particulate matter (pm10/pm2.5). https://www.gov.uk/government/statistics/air-quality-statistics/concentrations-of-particulate-matter-pm10-and-pm25, 2023. [Accessed: 09-May-2023].

Department for Transport. Transport and Environment Statistics 2022, 2022. URL {https://www.gov.uk/government/statistics/transport-and-environment-statistics-2022/transport-and-environment-statistics-2022}.

K. Developers. Keras. https://keras.io/, 2023a. Accessed: May 5, 2023.

N. Developers. Numpy. https://numpy.org/, 2023b. Accessed: May 5, 2023.

P. Developers. Pandas: Powerful data analysis tools for python. https://pandas.pydata.org/, 2023c. Accessed: May 5, 2023.

P. Devguide. Garbage collection in python. https://devguide.python.org/internals/garbage-collector/, 2023. Accessed: 09 May 2023.

B. Duan. *The robustness of trimming and Winsorization when the population distribution is skewed.* Tulane University, 1998. ISBN 978-0-599-03929-2.

Facebook. Prophet. https://facebook.github.io/prophet/, 2023.

D. Fecht, P. Fischer, L. Fortunato, G. Hoek, K. de Hoogh, M. Marra, H. Kruize, D. Vienneau, R. Beelen, and A. Hansell. Associations between air pollution and socioeconomic characteristics, ethnicity and age profile of neighbourhoods in england and the netherlands. *Environmental Pollution*, 198:201–210, 2015. ISSN 0269-7491. doi: https://doi.org/10.1016/j.envpol.2014.12.014. URL https://www.sciencedirect.com/science/article/pii/S0269749114005144.

H. Feng, B. Zou, J. Wang, and X. Gu. Dominant variables of global air pollution-climate interaction: Geographic insight. *Ecological Indicators*, 99:251–260, 2019. ISSN 1470-160X. doi: https://doi.org/10.1016/j.ecolind.2018.12.038. URL https://www.sciencedirect.com/science/article/pii/S1470160X18309713.

Folium Development Team. Folium Documentation, 2013. URL {https://python-visualization.github.io/folium/}. Created using Sphinx 3.1.0. Theme by vkvn.

D. for Environmental Food and R. Affairs. Nitrogen dioxide (no2). https://www.gov.uk/government/statistics/air-quality-statistics/ntrogen-dioxide, 2023. Accessed: 09 May 2023.

U. Foresight. Newcastle named smart city of the year (2019). https://urbanforesight.org/latest/newcastle-named-smart-city-of-the-year/, 2019. Accessed: 08 May 2023.

M. François, A. Fort, P. Crave, F. Osiurak, and J. Navarro. Gauges design for a digital instrument cluster: Efficiency, visual capture, and satisfaction assessment for truck driving. *International Journal of Industrial Ergonomics*, 72:290–297, 2019. ISSN 0169-8141. doi: https://doi.org/10.1016/j.ergon.2019.06.004. URL https://www.sciencedirect.com/science/article/pii/S0169814119300216.

Geopandas Development Team. Geopandas Documentation - GeoDataFrame, Year. URL {https://geopandas.org/en/stable/docs/reference/api/geopandas.GeoDataFrame.html}.

Google. Google maps, 2005. URL https://www.google.com/maps.

A. Grounds. Newcastle united – st james' park. https://www.awaygrounds.com/newcastle-united-st-james-park/, 2023. Accessed: 2023-05-24.

Highcharts. Highcharts - interactive javascript charts for web and mobile. https://www.highcharts.com/, 2023. Accessed: 09 May 2023.

C. D. HUNDHAUSEN, S. A. DOUGLAS, and J. T. STASKO. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing*, 13(3):259–290, 2002. ISSN 1045-926X. doi: https://doi.org/10.1006/jvlc.2002.0237. URL https://www.sciencedirect.com/science/article/pii/S1045926X02902375.

IBM. What is a rest api? https://www.ibm.com/topics/rest-apis, 2023. Accessed: 09 May 2023.

P. T. Inc. Plotly Python with Pandas Backend, 2023. URL https://plotly.com/python/pandas-backend.

P. James, J. Jonczyk, D. Bell, L. Chapman, N. Cowell, J. Evans, D. Topping, T. Bannan, E. Murabito, and E. Tsoneva. Urban observatories sensor report, December 2021. Newcastle University, University of Birmingham, The University of Manchester.

joblib developers. joblib. https://joblib.readthedocs.io, 2023. Software package.

E. Kamburjan and L. Grätz. Increasing engagement with interactive visualization: Formal methods as serious games. In J. F. Ferreira, A. Mendes, and C. Menghi, editors, *Formal Methods Teaching*, pages 43–59, Cham, 2021. Springer International Publishing. ISBN 978-3-030-91550-6.

L. Z. Kelley. kalepy: a python package for kernel density estimation, sampling and plotting. *Journal of Open Source Software*, 6(57):2784, 2021. doi: 10.21105/joss.02784. URL https://doi.org/10.21105/joss.02784.

B. Kumar Jha and S. Pande. Time series forecasting model for supermarket sales using fb-prophet. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 547–554, 2021. doi: 10.1109/ICCMC51019.2021.9418033.

Lars Yencken. smopy. https://pypi.org/project/smopy/, Accessed 2023. Python Package Index.

S. learn developers. Scikit-learn: Machine learning in python. https://scikit-learn.org/stable/index.html, 2023. Accessed: May 5, 2023.

C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013. ISSN 0022-1031. doi: https://doi.org/10.1016/j.jesp.2013.03.013. URL https://www.sciencedirect.com/science/article/pii/S0022103113000668.

Meta. React: A javascript library for building user interfaces. https://reactjs.dev/, Accessed: 2023.

Microsoft. Refresh dataset. Microsoft Power BI REST APIs, 2021. URL https://docs.microsoft.com/en-us/rest/api/power-bi/datasets/refresh-dataset. Accessed: May 5, 2023.

Microsoft. Run python scripts in power bi desktop. Microsoft Power BI documentation, 2023. URL https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-python-scripts. Accessed: May 5, 2023.

S. R. Midway. Principles of effective data visualization. *Patterns*, 1(9): 100141, 2020. ISSN 2666-3899. doi: \url{https://doi.org/10.1016/j.patter.2020.100141}. URL \url{https://www.sciencedirect.com/science/article/pii/S2666389920301896}.

L. T. Mohammed, A. A. AlHabshy, and K. A. ElDahshan. Big data visualization: A survey. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–12, 2022. doi: 10.1109/HORA55278.2022.9799819.

F. I. Mowbray, S. M. Fox-Wasylyshyn, and M. M. El-Masri. Univariate outliers: A conceptual overview for the nurse researcher. *Canadian Journal of Nursing Research*, 51(1):31–37, 2019. doi: 10.1177/0844562118786647. URL https://doi.org/10.1177/0844562118786647. PMID: 29969044.

L. Neal, P. Agnew, S. Moseley, C. Ordóñez, N. Savage, and M. Tilbee. Application of a statistical post-processing technique to a gridded, operational, air quality forecast. *Atmospheric Environment*, 98: 385–393, 2014. ISSN 1352-2310. doi: https://doi.org/10.1016/j.atmosenv.2014.09.004. URL https://www.sciencedirect.com/science/article/pii/S1352231014006967.

Newcastle City Council. Clean air zone launches later this month, 2023. URL https://www.newcastle.gov.uk/citylife-news/transport/clean-air-zone-launches-later-month. Accessed: 05 January 2023.

M. U. Observatory. Manchester urban observatory, 2023a. URL https://manchester-i.com/home. Accessed: 2023-05-22.

N. U. Observatory. Urban observatory api examples. Newcastle Urban Observatory API documentation, 2023b. URL https://newcastle.urbanobservatory.ac.uk/api_docs/examples/example/2/. Accessed: May 5, 2023.

G. Papacharalampous and H. Tyralis. Evaluation of random forests and prophet for daily streamflow forecasting. *Advances in Geosciences*, 45: 201–208, 08 2018. doi: 10.5194/adgeo-45-201-2018.

J. M. Perkel. Why jupyter is data scientists' computational notebook of choice. *Nature*, 563(7729):145–146, 2018. doi: 10.1038/d41586-018-07196-1.

J. PIBERNIK, J. DOLIĆ, R. STANIĆ, and L. MANDIĆ. User experience of wait-animation progress indicators. In *8th CONFERENCE ON INFORMATION AND GRAPHIC ARTS TECHNOLOGY*, 2018. URL "https://www.bib.irb.hr/943014/download/943014.8_CIGT_Proceedings1.pdf#page=123".

Project Jupyter. ipywidgets. https://ipywidgets.readthedocs.io, 2023.

H. Qu, W.-Y. Chan, A. Xu, K.-L. Chung, K.-H. Lau, and P. Guo. Visual analysis of the air pollution problem in hong kong. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1408–1415, 2007. doi: 10.1109/TVCG.2007.70523.

K. Reda and D. A. Szafir. Rainbows revisited: Modeling effective colormap design for graphical inference. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1032–1042, 2021. doi: 10.1109/TVCG.2020.3030439.

H. Ritchie and M. Roser. Air pollution. *Our World in Data*, 2017. https://ourworldindata.org/air-pollution.

A. Romano, C. Sotis, G. Dominioni, and S. Guidi. The scale of covid-19 graphs affects understanding, attitudes, and policy preferences. *Health Economics*, 29(11):1482–1494, 2020. doi: https://doi.org/10.1002/hec.4143. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hec.4143.

N. H. Savage, P. Agnew, L. S. Davis, C. Ordóñez, R. Thorpe, C. E. Johnson, F. M. O'Connor, and M. Dalvi. Air quality modelling using the met office unified model (aqum os24-26): model description and initial evaluation. *Geoscientific Model Development*, 6(2):353–372, 2013. doi: 10.5194/gmd-6-353-2013. URL https://gmd.copernicus.org/articles/6/353/2013/.

W. Schlenker and W. R. Walker. Airports, Air Pollution, and Contemporaneous Health. *The Review of Economic Studies*, 83(2):768–809, 10 2015. ISSN 0034-6527. doi: 10.1093/restud/rdv043. URL https://doi.org/10.1093/restud/rdv043.

D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2015.

I. Shancita, H. Masjuki, M. Kalam, I. Rizwanul Fattah, M. Rashed, and H. Rashedul. A review on idling reduction strategies to improve fuel economy and reduce exhaust emissions of transport vehicles. *Energy Conversion and Management*, 88:794–807, 2014. ISSN 0196-8904. doi: https://doi.org/10.1016/j.enconman.2014.09.036. URL https://www.sciencedirect.com/science/article/pii/S0196890414008371.

A. Singh, N. Thakur, and A. Sharma. A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1310–1315, 2016.

M. Stone. Choosing colors for data visualization. *Business Intelligence Network*, 2, 2006.

F. Stulp and O. Sigaud. Many regression algorithms, one unified model: A review. *Neural Networks*, 69:60–79, 2015. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2015.05.005. URL https://www.sciencedirect.com/science/article/pii/S0893608015001185.

S. Tanda, R. Ličbinský, J. Hegrová, and W. Goessler. Impact of new year's eve fireworks on the size resolved element distributions in airborne particles. *Environment International*, 128:371–378, 2019. ISSN 0160-4120. doi: https://doi.org/10.1016/j.envint.2019.04.071. URL https://www.sciencedirect.com/science/article/pii/S0160412019304921.

O. Toker and B. Kuhn. A python based testbed for real-time testing and visualization using ti's 77 ghz automotive radars. In *2019 IEEE Vehicular Networking Conference (VNC)*, pages 1–4, 2019. doi: 10.1109/VNC48660.2019.9062830.

E. P. Trindade, M. P. F. Hinnig, E. M. da Costa, J. S. Marques, R. C. Bastos, and T. Yigitcanlar. Sustainable development of smart cities: a systematic review of the literature. *Journal of Open Innovation: Technology, Market, and Complexity*, 3(3):1–14, 2017. ISSN 2199-8531. doi: https://doi.org/10.1186/s40852-017-0063-2. URL https://www.sciencedirect.com/science/article/pii/S2199853122003316.

UK Government. Health matters: Air pollution, 2018. URL https://www.gov.uk/government/publications/health-matters-air-pollution/health-matters-air-pollution.

UN Population Division. The world's cities in 2018:: Data booklet. https://digitallibrary.un.org/record/3799524, 2018. Accessed: 20 February 2023.

N. University. Urban observatory, 2023. URL https://urbanobservatory.ac.uk/. Accessed: February 21, 2023.

V. M. van Zoest, A. Stein, and G. Hoek. Outlier detection in urban air quality sensor networks. *Water, Air, & Soil Pollution*, 229(4):111, 2018. ISSN 1573-2932. doi: 10.1007/s11270-018-3756-7. URL https://doi.org/10.1007/s11270-018-3756-7.

J. Vanderplas. mpld3: Bringing matplotlib to the browser. https://mpld3.github.io/, Accessed: 2023.

H. P. Vinutha, B. Poornima, and B. M. Sagar. Detection of outliers using interquartile range technique from intrusion dataset. In S. C. Satapathy, J. M. R. Tavares, V. Bhateja, and J. R. Mohanty, editors, *Information and Decision Sciences*, pages 511–518, Singapore, 2018. Springer Singapore. ISBN 978-981-10-7563-6.

Y. Wang, F. Han, L. Zhu, O. Deussen, and B. Chen. Line graph or scatter plot? automatic selection of methods for visualizing trends in time series. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1141–1154, 2018. doi: 10.1109/TVCG.2017.2653106.

M. Waskom. seaborn: statistical data visualization, 2012-2022. URL `https://seaborn.pydata.org/`.