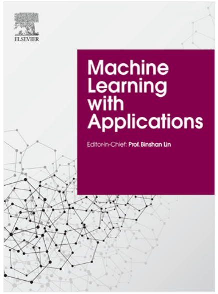
Journal Pre-proof



Deep learning-based multi-target regression for traffic-related air pollution forecasting

Taofeek Dolapo Akinosho, Muhammad Bilal, Enda Thomas Hayes, Anuoluwapo Ajayi, Ashraf Ahmed, Zaheer Khan

PII: S2666-8270(23)00027-0
DOI: https://doi.org/10.1016/j.mlwa.2023.100474
Reference: MLWA 100474

To appear in: *Machine Learning with Applications*

Received date : 6 December 2022
Revised date : 7 March 2023
Accepted date : 23 May 2023

Please cite this article as: T.D. Akinosho, M. Bilal, E.T. Hayes et al., Deep learning-based multi-target regression for traffic-related air pollution forecasting. *Machine Learning with Applications* (2023), doi: https://doi.org/10.1016/j.mlwa.2023.100474.

Revised manuscript (Clean Version)

Deep Learning-based Multi-Target Regression For Traffic-Related Air Pollution Forecasting

Taofeek Dolapo Akinosho* TAOFEEK.AKINOSHO@UWE.AC.UK

*Corresponding author
Big Data Enterprise and Artificial Intelligence Lab (Big-DEAL)
Bristol Business School
University of the West of England, Frenchay Campus,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Muhammad Bilal MUHAMMAD.BILAL@UWE.AC.UK
Big Data Enterprise and Artificial Intelligence Lab (Big-DEAL)
Bristol Business School
University of the West of England, Frenchay Campus,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Enda Thomas Hayes ENDA.HAYES@UWE.AC.UK
Air Quality Management Resource Centre,
University of the West of England, Frenchay Campus,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Anuoluwapo Ajayi ANUOLUWAPO.AJAYI@UWE.AC.UK
Big Data Enterprise and Artificial Intelligence Lab (Big-DEAL)
Bristol Business School
University of the West of England
Bristol, United Kingdom

Ashraf Ahmed ASHRAF.AHMED@BRUNEL.AC.UK
Department of Civil and Environmental Engineering
Brunel University London
Kingston Lane
Uxbridge, United Kingdom

Zaheer Khan ZAHEER2.KHAN@UWE.AC.UK
Department of Computer Science and Creative Technologies,
University of the West of England , Frenchay Campus,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Abstract

Traffic-related air pollution (TRAP) remains one of the main contributors to urban pollution and its impact on climate change cannot be overemphasised. Experts in developed countries strive to make optimal use of traffic and air quality data to gain valuable insights into its effect on public health. Over the years, the research community has developed advanced methods of forecasting traffic-related pollution using several machine learning methods albeit with persistent accuracy and insufficient data challenges. Despite the potentials of emerging techniques such as multi-target deep neural network to achieve optimal solutions, they are yet to be fully exploited in the air quality space due to their complexity and unavailability of the right training data. It is to this end that this study investigates the impact of integrating an updated data set including road elevation, vehicle emissions factor and background maps with traffic flow, weather and pollution data on TRAP forecasting. To explore the robustness and adaptability of our methodology, the study was carried out in one major city (London), one smaller city (Newport) and one large town (Chepstow) in the United Kingdom. The forecasting task was modelled as a multi-target regression problem and experiments were carried out to predict NO_2, $PM_{2.5}$ and PM_{10} concentrations over multiple timesteps. Fastai's tabular model was used alongside prophet's time-series model and scikit-learn's multioutputregressor for experimentation with fastai recording the overall best performance. Statistical tests run using Friedman and Wilcoxon test also revealed the significance of the fastai model with p-values < 0.05. Finally, a model explanation tool was then used to reveal the most and least influential features from the newly curated data set. Results showed traffic count and speed were part of the most contributing features. This result demonstrates the impact of these and other introduced features on TRAP forecasting and will serve as a foundation for related studies.

Keywords: Traffic-related pollution, Road Transport, Multi-target regression, Deep Learning, Pollution forecasting

1. Introduction

Highways are designed to facilitate intercity travels within a country while providing links to other public or private roads. However, commuters or residents living close to these highways are constantly exposed to numerous pollutants that can result in respiratory and cardiovascular health diseases. An average commuter in a car spends 4-7% of their day on or close to these major roads constantly polluted with vehicle emissions and atmospheric reactions of pollutants such as nitrogen oxides (NO_x) and particulate matter ($PM_{2.5}$,PM_{10}) (Matz et al. 2018). Studies have shown that continuous exposure to these kind of pollutants increases the risks of dying from stroke, heart failure and asthma attacks (Mabahwi et al. 2014). The particle sizes of particulate matter makes it one of the most difficult traffic-related pollutant to control despite its contribution to global mortality (Peeples 2020, Jida et al. 2021). In 2015 alone, 20% of $PM_{2.5}$-related deaths in developed countries such as the United States of America (USA), Germany and the United Kingdom (UK) were linked to traffic-related air pollution (Jerrett 2015). Unfortunately, traffic congestion aggravates this problem by increasing the time spent on these highways and exposure to these contaminants.

Research into UK highway pollution is limited, with few monitoring stations from the UK government operated national Automatic Urban and Rural Network (AURN) sparsely positioned in areas close to major roads to record concentration levels. Data captured from these stations are used by the government to detect long-term pollutant trends, evaluate the effectiveness of certain policy changes, and to determine compliance with UK health-based air quality objectives. Similarly, the AURN data is used to support the UK's air quality forecast system which is a modelled data set created by the Met Office in a bid to reduce morbidity and mortality from traffic-related pollution. However, instantaneous forecasting using real-time data is non-existent since it can be quite challenging. The process of estimating concentration levels of pollutants is complicated and often constrained by contributing factors such as weather conditions and traffic flow (Sun et al. 2021). In the last decade, a number of studies have focused on investigating innovative ways to address the challenges of accurate forecasting although with some persistent constraints.

1.1. Existing approaches to traffic-related air pollution (TRAP) forecasting

Conventional methods for TRAP forecasting can be broadly categorised into deterministic, statistical and machine learning-based approaches (Xie et al. 2020). Some studies have adopted a singular modelling approach while a considerable number combine these methods for better accuracy. Deterministic methods are generally less adopted due to limitations such as compute-intensiveness, lack of spatial and temporal dependencies and the need to mathematically represent chemical reactions between pollutants (Cabaneros et al. 2017, Hua et al. 2019). Statistical methods such as multiple linear regression, autoregression and linear-logarithmic regression are preferred alternatives for solving the shortcomings in deterministic methods. For example, the study of Comert et al. (2020) used several variants of linear regression models mixed with grey systems to predict ozone and $PM_{2.5}$ levels using historical traffic volume and air quality data. Machine learning (ML) methods like Neural Networks have also been exploited for TRAP forecasting: Jida et al. (2021) used the approach to estimate traffic-related $PM_{2.5}$ and PM_{10} in the city of Addis Ababa in Ethiopia. Cabaneros et al. (2017) used a hybrid of neural networks and stepwise regression to predict day-ahead roadside NO_2 concentration levels. Six ML algorithms - Boosted Regression Trees (BRT), Random Forest (RF), Cubist, Extreme Gradient Boosting(XGBoost), Support Vector Machine (SVM) and Generalised Additive Model (GAM) were evaluated in the study of Li et al. (2020) to address the limitations of statistical methods by predicting high temporal resolutions of roadside $PM_{2.5}$ and NO_x. In a similar research, Fong et al. (2020) used transfer learning (a process of adapting existing ML models for new prediction tasks) and Recurrent Neural Networks (RNNs) to make day-ahead predictions of particulate matter. Although these studies demonstrate the effectiveness of ML approaches, many of them still had notable limitations.

1.2. Limitations of existing approaches

The first and most pertinent limitation of existing approaches is inaccurate prediction and limited generalisability. This constraint can be attributed to the quality of the data sets that models are trained on. Traditional models are mostly trained on traffic flow, meteorological and historic pollution data collected over many years. Other highway and traffic-related data such as background air pollution concentrations, vehicle emission factor and highway topography are often ignored because of their unavailability. Consequently, many of the machine learning models only excel on the often limited data sets upon which they have been trained. The study of Fong et al. (2020) for example, could only make next day predictions and struggled with periods shorter than a day or even several days ahead. Another important limitation is the inability of these models to simultaneously and accurately predict multiple pollutants and the impact of contributing variables. Model predictions typically depend on the linear dependency between influential highway parameters (such as traffic flow and wind directions) and pollutants. However, these relationships are complex and non-linear, thereby making simultaneous predictions even more difficult (Masmoudi et al. 2020). Also, most of the developed models do not offer pragmatic solutions that can be deployed in a real-world scenario. Rigorous validation of these models in these kinds of scenarios is almost non-existent. Table 1 summarises the limitations of the reviewed studies in comparison to the proposed approach in this study.

1.3. The need for multi-target regression (MTR) and deep learning

Motivated by the aforementioned limitations, this study takes a different approach and models the prediction task as a multi-target regression problem with additional highway data such as background air pollution concentrations from the UK Pollution Climate Model (PCM), vehicle emissions factor and terrain data added to the conventional weather and historic pollution data. While MTR permits the simultaneous prediction of multiple dependent variables, its real-world application still poses numerous challenges due to the complexity of some domains (Borchani et al. 2015). Few studies that have explored MTR for pollutant concentration forecasting have either had limited accuracies or feature selection issues. None has evaluated a combination of the data set put together in this study. Similarly, it has been established in several studies that deep learning algorithms allow models to learn the fundamental relationships between variables of a data set (Guo & Berkhahn 2016) but some scholars argue the efficacy of deep learning algorithms developed for tabular data (Fayaz et al. 2022). Hence, this study also seeks to validate that claim.

In summary, the main contributions of this study to existing knowledge are:

- The study extends existing machine learning based air quality forecasting studies by integrating highway information in addition to meteorological and pollution data.
- Training and evaluating the performance of a single MTR model for multi-target prediction of traffic-related pollutants (NO_2, PM_{10} and $PM_{2.5}$) using these integrated data set.

- Exploring categorical embeddings in tabular data models and comparing the performance to time series and regression algorithms using state-of-the-art libraries.

- Evaluating the feature importance of the best performing algorithm to determine the most contributing features.

The rest of the manuscript is organised as follows: the next section highlights the data collection and preprocessing steps towards model training and evaluation, section 3 introduces the MTR approach and details the entire model training process. Experimentation steps and model validation results on four major UK regions are presented in sections 4 and 5. An analysis of the feature importance for the best performing algorithm is presented in section 6 while section 7 discusses the general findings of the study and its implication for practice. Section 8 concludes and summarises the study.

Table 1: Summary of existing approaches and comparison with proposed approach

S/N	Author(s)	Year	Method	Aim	Target Pollutants	Limitations	Region
1	Cabaneros et al. (2017)	2017	Neural Network	Investigate the effect of feature selection on NO_2 concentration prediction	NO_2	Impact of traffic data and emission factor were not considered	London
2	Suleiman et al. (2019)	2019	Support Vector Machine, Artificial Neural Network, Boosted Regression Trees	Evaluate the effectiveness of roadside pollutant reduction scenarios using ML-based models	PM_{10},$PM_{2.5}$	Limited dataset, No multi-target prediction	London
3	Wang et al. (2020)	2020	Artificial Neural Network, Gradient Boost, Land Use Regression(LUR)	To investigate the performance of LURs against machine learning models	$PM_{2.5}$, Black Carbon	No multi-target output, No hyperparameter tuning for improved performance	Toronto
4	Comert et al. (2020)	2020	Regression Models	Investigating the impact of traffic volume on air quality	$PM_{2.5}$, Ozone	Impact of weather and traffic parameters not considered	South Carolina
5	Li et al. (2020)	2020	Random Forest, Support Vector Machine, Gradient Boosting, Generalised Additive Model, XGBoost, Cubist	To predict street-level pollution at roadside stations	NO_x,$PM_{2.5}$	Single target models that could only make hourly predictions	Hong Kong
6	Fong et al. (2020)	2020	Recurrent Neural Network	Explore transfer learning for better accuracy on limited observed data	NO_2,PM_{10},$PM_{2.5}$, CO,NO	Could not make next day predictions	Macau
7	Jida et al. (2021)	2021	Artificial Neural Network	To investigate the contribution of roadside vehicles to particulate matter in Ethiopia	PM_{10},$PM_{2.5}$	Did not explore the impact of background pollution concentration	Ethiopia
8	**Current study**	**2023**	**Recurrent Neural Network, Categorical Embeddings, Tabular Models, Regression Models**	**Investigate the impact of additional highway data in TRAP forecasting**	NO_2,PM_{10},$PM_{2.5}$	**Addresses existing limitations by integrating diverse highway dataset and exploring categorical embeddings for improved performance**	**Newport,Chepstow, Southwark, Lewisham**

2. Study sites and data collection strategy

2.1. Study sites

To explore the robustness and adaptability of our methodology, the study was carried out in one major city (London), one smaller city (Newport) and one large town (Chepstow) in the UK and data was collected between November 2020 and November 2021. Despite the impact of Covid lockdowns in this period, there was adequate traffic flow that allowed us to study the effects of traffic movement trends on air quality. A total of fourteen custom built Internet of Things (IoT) devices named REVIS were employed and distributed on highways in these cities to capture real-time air pollution and weather data as illustrated in figure 1. Development, evaluation and performance details of the REVIS devices have previously been described in Akinosho et al. (2022). For London, six devices were deployed on each of the A302 highway in Southwark and A2209 highway in Lewisham with each device mounted on lamp posts 100m apart or custom poles in the absence of lamp posts. One device was deployed on the M4 highway in Newport and another was placed on the A48 highway in Chepstow.

Additional weather data not captured by the REVIS devices were integrated from the nearest AURN stations. Publicly available background mapping data was captured from the Department for Environment, Food and Rural Affairs' (DEFRA) website [1] while their emissions factor toolkit was used to estimate traffic exhaust emissions for different vehicle categories. Similarly, Highways England's webtris application [2] provided traffic congestion, average vehicle speed and traffic volume data as required. Finally, Google earth application was used to extract terrain information for the case study sites.

2.2. Data description

The approach used to collect data in this study was to imagine the highways as consisting of multiple segments. Deployed devices were mapped to different segments of the highway and data captured for each device represented that highway segment. This way, it was easier to match device measurements with other data set such as background concentration that are represented by 1x1km grids. This section describes the data set specification which is also summarised in Appendix A.

2.2.1. Pollution data

NO_2, PM_{10}, and $PM_{2.5}$ data captured every five minutes by the REVIS devices were included in the data set. After collocating the NO_2 readings of the devices with the nearest AURN stations in Chepstow [3], Newport [4] and London (Lewisham [5] and Southwark [6])

[1] https://uk-air.defra.gov.uk/data/laqm-background-home
[2] https://webtris.highwaysengland.co.uk/
[3] https://uk-air.defra.gov.uk/networks/site-info?site_id=CHP
[4] https://uk-air.defra.gov.uk/networks/site-info?site_id=NPT3
[5] https://uk-air.defra.gov.uk/networks/site-info?site_id=LW1
[6] https://uk-air.defra.gov.uk/networks/site-info?site_id=SK5

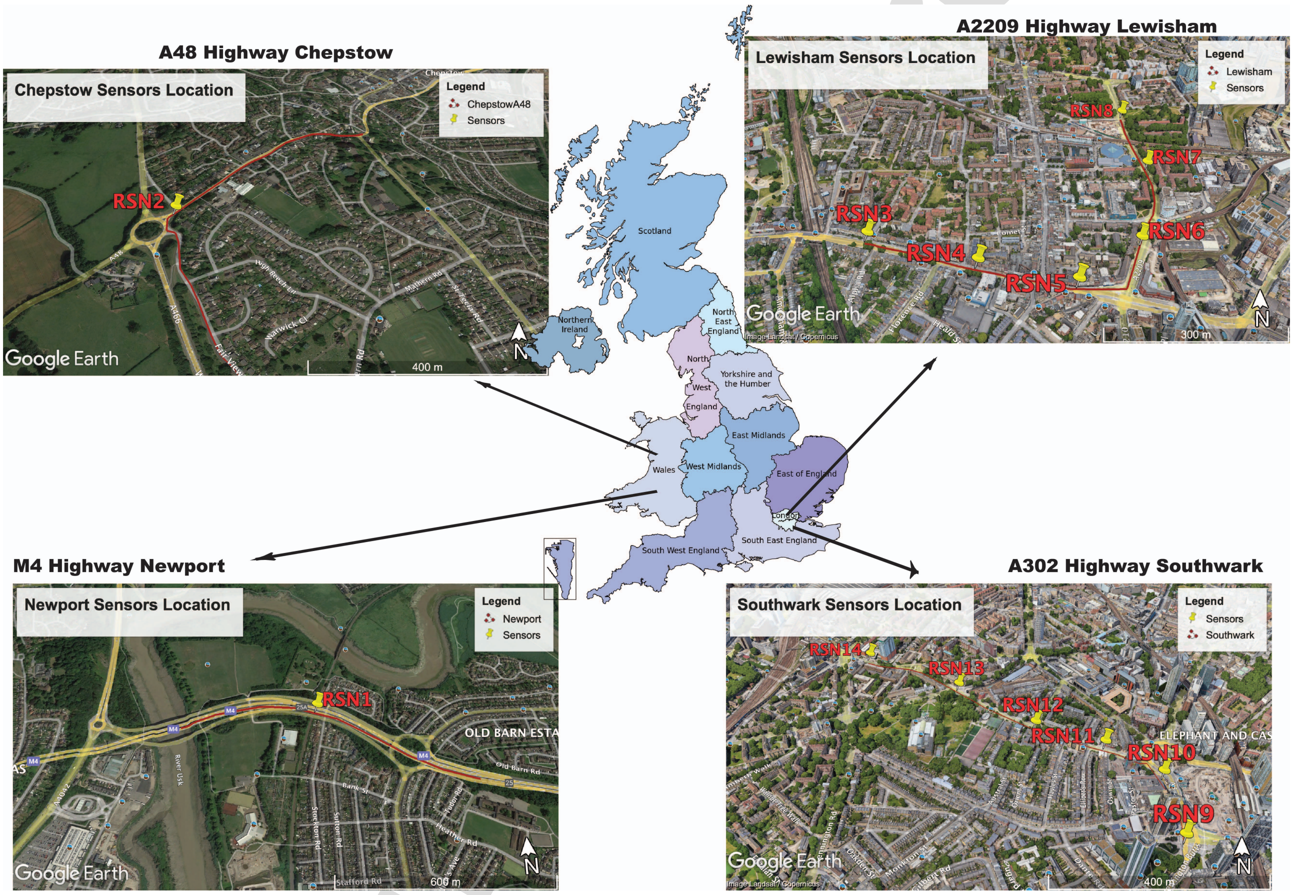


Figure 1: A map of case study highways and sensing device distribution in this study.

it was clear that the NO_2 readings were inaccurate with the average correlation of 0.07. This inaccuracy was linked to the analogue NO_2 sensors used on the REVIS devices, which responded strongly to changes in temperature and relative humidity, to get negative readings sometimes. As a result, NO_2 measurements from AURN stations were used in place of the REVIS NO_2 data. The REVIS data for PM_{10}, and $PM_{2.5}$ were retained since there was a good correlation of 0.73 and 0.8 with the AURN data. To ensure efficient data mapping, the REVIS data had to be summarised into hourly aggregates to match the hourly readings in the integrated AURN data.

	ROAD_NAME	REGION_NAME	SEGMENT_NAME	PM2.5 (µg/m³)	PM10 (µg/m³)	NO2
11985	A48 Road	Chepstow	A48 Road - S02	3.376741	7.228102	38.63250
11986	A302 Road	Southwark	A302 Road - S02	3.166877	7.033940	34.42500
11987	A48 Road	Chepstow	A48 Road - S02	3.367977	6.964162	47.62125
11988	A302 Road	Southwark	A302 Road - S02	2.223098	5.108366	38.25000
11989	A48 Road	Chepstow	A48 Road - S02	3.100474	7.021209	41.11875

Figure 2: Snapshot of pollution data.

2.2.2. Traffic data

Traffic information was integrated from Highways Englands' traffic monitoring unit (TMU) sites. The data which can be downloaded through an API or a web interface includes counts for vehicles less than 5.2m or greater than 11.6m in length, counts for each vehicle type, total traffic volume and average traffic speed. The measurements only included descriptions of vehicle lengths so it was necessary to map different vehicle types to the appropriate lengths for easy comprehension. Cars were mapped to 0-520cm, buses to 521-660cm, light goods vehicle (LGV) to 661-1160cm and heavy goods vehicle (HGV) to 1160cm+ (Bálint et al. 2014). TMU data are captured every minute so just like the historic pollution data, this data was also summarised into hourly aggregates.

	Bus Avg Speed	Bus Count	Car Avg Speed	Car Count	HGV Avg Speed	HGV Count	LGV Avg Speed	LGV Count
0	54.00	40.0	54.00	380.0	54.00	31.0	54.00	32.0
1	53.50	43.0	53.50	506.0	53.50	25.0	53.50	32.0
2	53.25	33.0	53.25	295.0	53.25	24.0	53.25	32.0
3	53.50	17.0	53.50	448.0	53.50	22.0	53.50	9.0
4	52.50	58.0	52.50	731.0	52.50	32.0	52.50	24.0

Figure 3: Snapshot of traffic data.

2.2.3. Weather data

The temperature, humidity and pressure for the four highways of interest were measured in real-time along with pollution data. However, previous studies have shown the impact of other meteorological parameters such as wind speed and wind direction in aiding pollutant dispersion (Chen & Ye 2019). The modelled wind speed and direction data were therefore integrated from same AURN stations used for NO_2 while data from REVIS devices were aggregated to match. Wind direction across the four regions ranged between 16° and 360° and the wind speed was between 0 and 16 knots.

	Humidity	Wind Direction	Wind Speed	Temperature	Pressure
11985	79.669189	329.8	10.0	7.233315	1000.168030
11986	76.290894	311.6	6.7	9.282722	1006.908813
11987	76.371329	336.0	8.8	8.219354	1001.844552
11988	78.375244	325.1	6.1	8.699036	1008.303304
11989	70.254517	336.5	8.1	11.230164	1002.729004

Figure 4: Snapshot of weather data.

2.2.4. Elevation data

Research into emission modelling in recent years has shown that vehicle exhaust outputs varies in uphill and downhill situations (Zhai et al. 2020, Xu et al. 2020). The vehicle's engine is under more pressure as it goes uphill and under less pressure downhill. It is unknown whether capturing this sort of highway information would result in an improved estimation accuracy. More importantly, highway terrain data such as elevation and gradient data are required to compute the vehicle emissions factor for different vehicle types. Google earth's desktop application was used to capture this information after the highway trajectories were drawn.

	Highway Elevation
11985	68.932396
11986	3.902021
11987	68.932396
11988	3.902021
11989	68.932396

Figure 5: Snapshot of elevation data.

2.2.5. Emissions factor data

Version 11.0 of DEFRA's emission factor toolkit (EFT) was used to compute the source apportionment of particulate matter and NO_2 for the different vehicle categories. EFT allows the specification of parameters such as the year of interest, road type, vehicle speed and vehicle type from the onset and automatically computes the required output based on

COPERT 5 specifications (COPERT is the standard EU vehicle emissions calculator). The traffic was selected as *'Detailed Option 2'* since the traffic data that was collected did not include information on vehicle types as either petrol, diesel or hybrid. This option allows non-detailed vehicle counts for cars, buses, LGVs and HGVs to be used as traffic flow input for EFT. The highway gradient information from Google earth was also fed into the tool while the *'flow direction'* was determined from the elevation chart in the application. As a result, the Newport, Lewisham and Southwark highways were specified as 'Up Hill' while Chepstow was specified as 'Down Hill' flow direction due to the single direction by which vehicles travelled. Finally, the below equations were used to verify the estimations from the toolkit and the values were close.

$$\text{For Uphill: } EF_2 = EF_1(1 + G \times [C_1 \times V + C_2]) \tag{1}$$

$$\begin{aligned} \text{For Downhill: } EF_2 &= EF_1(1 - G \times [C_1 \times V + C_2]) \text{ if } G \leq 2.5\% \\ EF_2 &= EF_1(1 - 0.025 \times [C_1 \times V + C_2]) \text{ if } G > 2.5\% \end{aligned} \tag{2}$$

where EF_1 and EF_2 denote emission factor for vehicles travelling at speed V on a level and uphill/downhill road respectively, G is the highway gradient and C_1 and C_2 are the gradient coefficients based on vehicle type and pollutant of concern (CERC 2019).

	No2 Emission Factor	PM Emission Factor
0	1735.0	2999.93920
1	2242.0	3836.07528
2	1367.0	2364.85630
3	1883.0	3193.08780
4	3178.0	5374.81560

Figure 6: Snapshot of emission factor data.

2.2.6. **Background air pollution concentration data**

Background concentration maps for a particular pollutant refers to data on contributions from other sources mixed with contributions from the source of interest (in this case road transport). These sources can range from natural to local sources like household coal burning, industries and even other means of transportation. It is therefore important to consider these other sources and eliminate them to avoid double counting (a situation where concentration for a pollutant is repeated unknowingly). This study utilizes the publicly available background pollution maps from DEFRA UK AIR resource website (UKAIR 2018) to capture this information for the four case study locations. It is noteworthy that this was the 2018 background maps covering 2020 and 2021 but do not account for long or short term impacts of Covid lockdowns on local sources. The data provides grid-based modelled background concentration for $PM_{2.5}$, PM_{10}, NO_x, and NO_2 from 2018 to 2030. The background concentration for 2020 and 2021 indicated in Table 2 below includes only rail, domestic, industrial and point sources. The minor road and motorway background concentration were not included to avoid double counting . This approach is similar to the one proposed in the study of Arunachalam et al. (2014).

Table 2: Pollutant background concentration for the four regions of interest in the year 2020 and 2021

Regions	**Grid_ref_x**	**Grid_ref_y**	$NO_2(ppb)$		$PM_{2.5}$ ($\mu g/m^3$)		PM_{10} ($\mu g/m^3$)	
			2020	2021	2020	2021	2020	2021
Newport	332500	189500	17.711	16.761	10.386	10.278	15.785	15.648
Chepstow	353500	193500	8.409	8.067	7.986	7.883	12.069	11.941
Lewisham	537500	177500	24.698	23.827	12.090	11.941	18.560	18.347
Southwark	531500	178500	28.954	27.997	12.706	12.555	19.768	19.552

	Background NO2	**Background PM2.5**	**Background PM10**
11985	8.067068	7.883311	11.941144
11986	27.997540	12.555302	19.552194
11987	8.067068	7.883311	11.941144
11988	27.997540	12.555302	19.552194
11989	8.067068	7.883311	11.941144

Figure 7: Snapshot of background concentration data.

3. Machine learning approach

This section describes the approach taken in this study to address the multi-target prediction problem. The pseudo-code for the proposed approach is highlighted below while the entire workflow is summarised in figure 8.

Algorithm 1 Multi-target algorithm for predicting NO_2,PM_{10} and $PM_{2.5}$.

Input: Dataset $\mathcal{D}(X, Y)$, Fastai tabular model $\mathcal{F}$, Prophet model $\mathcal{P}$, Multioutputregressor model $\mathcal{M}$, epochs ϵ, learning rate η, batch size β, estimators n, max depth d
Output: $(\hat{y}_1, \hat{y}_2, \hat{y}_3)$
Initialize: ϵ, η, β
Categorify($\mathcal{D}$)
FillMissing($\mathcal{D}$)
Normalize($\mathcal{D}$)
Split $\mathcal{D}$ into trainSet, testSet and validationSet
for $e = 1, ..., \epsilon$ **do**
    train $\mathcal{F}$ using trainSet, η and β
    validate($\mathcal{F}$, validationSet)
end
Return: Trained tabular model $\mathcal{F}_{trained}$
Initialize: $\mathcal{P}$
for $x_i, ..., x_n$ **do**
    $\mathcal{P}$.addRegressor(x)
end
train $\mathcal{P}$ using trainset
validate($\mathcal{P}$, validationSet)
Return: Trained model $\mathcal{P}_{trained}$
Initialize: n, d, $\mathcal{M}$
train $\mathcal{M}$ using trainSet, n and d
validate($\mathcal{M}$, validationSet)
Return: Trained model $\mathcal{M}_{trained}$
for $model \in (F_{trained}, \mathcal{P}_{trained}, \mathcal{M}_{trained})$ **do**
    for t = 1,..., 24 **do**
        Get: x_t
        if $t \neq 1$ **then**
        Predict: $(\hat{y}_1, \hat{y}_2, \hat{y}_3)_t$ using (model, $(\hat{y}_1, \hat{y}_2, \hat{y}_3)_{t-1}$,$x_t$)
        else if $t = 1$ **then**
        Predict: $(\hat{y}_1, \hat{y}_2, \hat{y}_3)_t$ using (model,x_t)
        Return: $\mathcal{F}$: $(\hat{y}_1, \hat{y}_2, \hat{y}_3)_t$,$\mathcal{P}$: $(\hat{y}_1, \hat{y}_2, \hat{y}_3)_t$,$\mathcal{M}$: $(\hat{y}_1, \hat{y}_2, \hat{y}_3)_t$
    end
end

3.1. Multi-target regression and RNNs

Neural Networks have become a familiar term among the artificial intelligence (AI) and machine learning research community. The ML approach which became more popular in 2012 as a result of its performance at the imagenet classification competition, has since grown into a widely adopted method for not just classification but also regression problems. Multi-target models in general refers to models that are able to automatically detect relationships

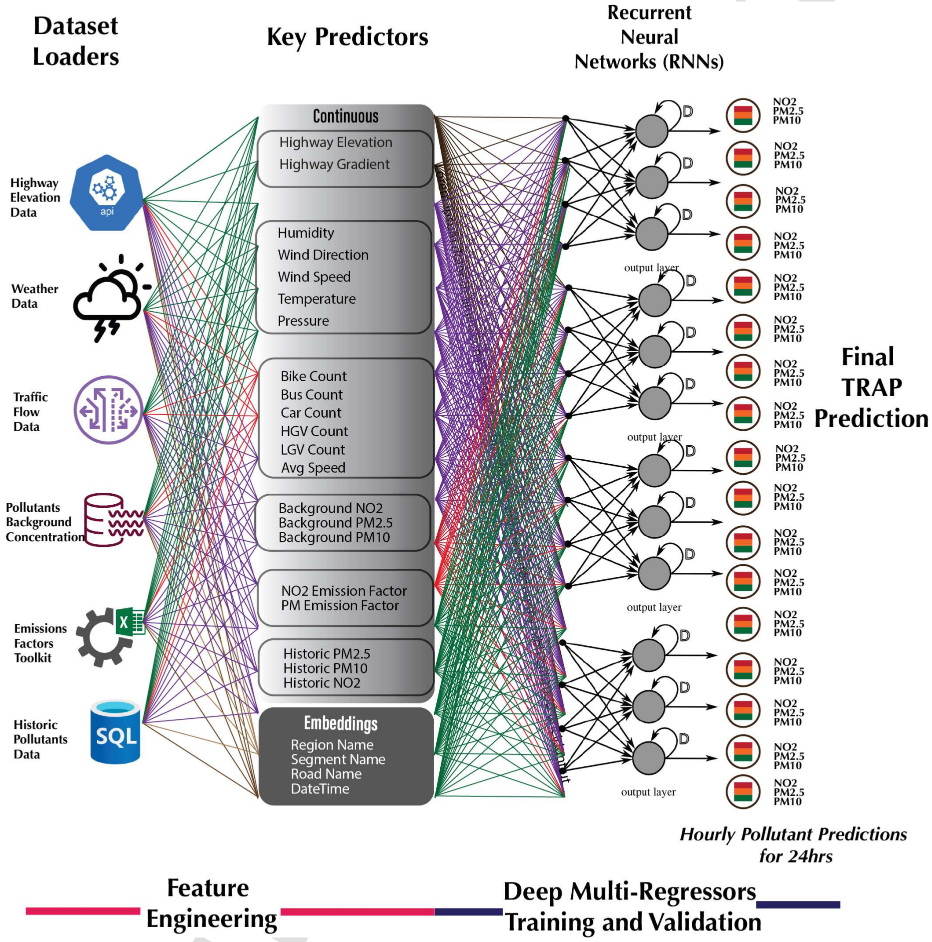
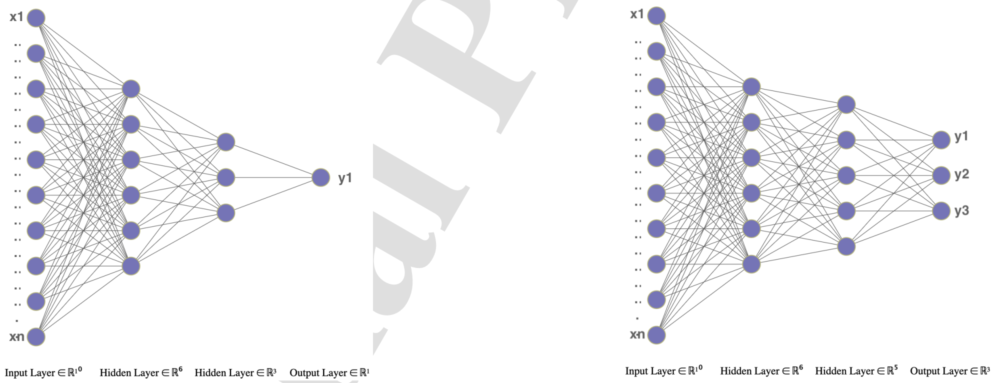


Figure 8: Multi-target model training architecture using the newly curated dataset. Feature engineering steps including normalisation and log transformation were carried out before training on three different algorithms used for experimentation.

between target variables, thereby resulting in better predictions (Korneva & Blockeel 2020). A multi-target regression neural network differs from its single-target counterpart by the number of predicted outputs. As illustrated in figures 9a and 9b, single-target predicts just one output using the set of features characterising the data set while multi-target can predict multiple outputs simultaneously. In terms of performance, multi-target outputs are simpler and faster to train than an ensemble of single-target models (Kocev et al. 2009). Multi-target models are more widely adopted for classification problems such as object classification, face recognition and sporadically used for regression problems (Spyromitros-Xioufis et al. 2012).

Recurrent neural networks are mainly associated with research involving time-series, sequence labelling and classification using visual, audio or text data. This class of neural networks and its variants - Gated Feedback Recurrent Neural Network (GRU) and Long-Short term memory (LSTM) are suitable for time-series problems since they are capable of keeping track of the temporal information within input data. Other neural network architectures like CNN and GANs struggle with these kind of data (Yu et al. 2019). Despite the competitiveness of RNNs over other architectures, its application to domains such as air quality forecasting is limited due to the inadequate understanding of its internal mechanisms (Shen et al. 2020). Fortunately, several libraries and frameworks have been introduced in recent times to take away the intricacies of the RNN implementation.



(a) Single-target neural network.

(b) Multi-target neural network.

Figure 9: Multi-target vs single-target neural networks.

3.2. Fastai, prophet and multioutputregressor methods

Fastai was first introduced in 2016 as a library built with a high level of abstraction to help AI enthusiasts with limited maths background to quickly develop deep learning models. With as little as 10 lines of codes, the complexities of developing such models are handled by fastai's customisable low, mid and high level APIs (Howard & Gugger 2020). The library is put forward as being capable of achieving state-of-the-art results in computer vision, natural language processing, collaborative filtering, and time-series problems. Another key attribute of the library which has caught the eye of researchers is the library's implementation of entity embeddings for encoding categorical features to achieve state-of-the-art results.

Prophet, on the other hand, is a library developed by Facebook to strategically introduce some modifications to traditional time-series algorithms. The library uses the idea of "changepoints" to generate additive regression models capable of automatically detecting and adapting to sudden changes in time-series trajectories (Taylor & Letham 2018). This implies a reduction in the efforts required to manually specify data shifts before training a model. The library is designed to be robust against missing data and is originally built for univariate daily, weekly and yearly time-series forecasting. However, with a few modifications to the library, such as the use of multiple regressors, multivariate prediction is possible. The default configuration in prophet is known to produce estimates similar to professional forecasters and therefore encourages quick experimentation. The library is famously used for sales as well as weather forecasting. The easiest way to install prophet is through its python or R package on PyPI and CRAN repositories.

Scikit-learn (Sklearn) is one of the most useful python library that houses different regression, classification and time-series algorithms. One of the wrapper regressor classes in sklearn is the MultiOuputRegressor class which permits the definition of one regressor from any of the available regression algorithms and then creates an instance for each output. One key advantage of the class is that it can be used to identify outputs that are independent of each other and also used to evaluate the performance of other multioutput models.

3.3. Data preprocessing

All the available data were first pulled together and merged into a single csv file using Oracle SQL procedures before preprocessing was initiated. It was important that these procedures were used to extract the data into separate database tables since they were generated as JSON strings directly from the IoT devices. The tables were joined using matching columns such as region or highway id and then loaded into a jupyter notebook for pre-processing and data cleansing. This data fusion technique is known as the early multi-view integration approach where the datasets are first joined together into a vector using a matching feature before training on a machine learning algorithm (Noble et al. 2004, Li et al. 2018, Guarino et al. 2022). The matching feature in this case is the region/highway id. Two versions of the data were created to adapt to the needs of the algorithms that were explored. The feature engineering steps that were taken are as follows:

- Data straight from the database had 232,553 rows and 10 columns. Each row represented a single reading for particular pollutant or weather data at 5 min intervals. One of the columns captured the *trend_type_id*, an integer which indicates the type of measurement (weather, pollutant, emission factor etc) that was measured. A dictionary was then created to convert these ids into meaningful and more descriptive strings. Pandas library was used for data manipulation and its pivot function was used to turn rows with matching dates into one single row while retaining the measurement type as columns. Missing measurements for a particular time point was represented with 'Nan'. The shape of the data set after this preprocessing step was 11,990 rows x 44 columns

- Next was to create the first version of the data set which includes extracted date information. Additional date attributes such as *day, month, year, dayofweek, ismonthend etc* were added to this data set. This step makes it easier for the algorithm to extract the date information from the datetime object. The second version of the data had just the date and pollutants data like a typical time series data set.

- Inspecting the data for missing values revealed 1111 missing data for the REVIS $PM_{2.5}$ and PM_{10} while the integrated AURN NO_2 had none. The missing values were replaced with data from the previous day using the last observation carried forward (LOCF) method which is one of the famous imputation methods for time series data (Hadeed et al. 2020). The same approach was used to fill missing values in other weather and traffic attributes.

- It was difficult to identify the underlying distribution of the pollutants since their min and max has a smaller scale of values as shown in Table 3. Hence, the log transform of all three pollutants was taken to make the distributions less skewed. The resulting plot of the distribution is shown in figure 10.

- Finally, the features were split into categorical and continuous features based on the type of values they hold as shown in Appendix A. This step facilitates the use of tabular models.

Table 3: Descriptive statistics of the pollutants data

Variable	count	mean	std	min	25%	50%	75%	max
NO_2 (ppb)	11990	21.954	16.405	0.631	9.753	16.910	30.379	132.370
$PM_{2.5}$ ($\mu g/m^3$)	10879	9.711	14.922	0.699	3.717	5.932	10.205	401.012
PM_{10} ($\mu g/m^3$)	10879	11.801	17.882	0.778	4.828	8.042	12.587	617.351

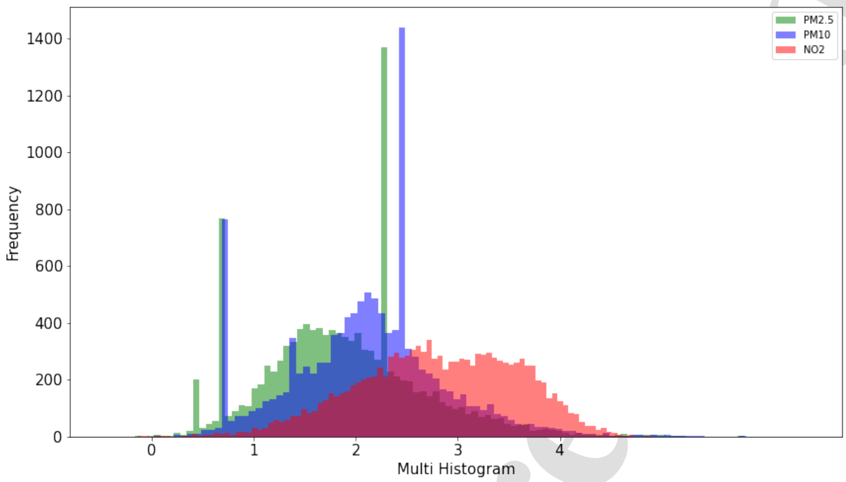


Figure 10: Data distribution for all three pollutants.

4. Experimentation and Model Training

This section highlights the experiments and optimisation techniques carried out in this study while results of each experiment are presented in subsequent sections. Figure 11 shows the difference between two sets of experiments carried out using fastai, prophet and multioutputregressor algorithms. Each experiment was carried out using separate jupyter notebooks and a dedicated high performance computer with 64gb RAM and Nvidia RTX 3080 GPU.

4.1. Experiment 1 - Comparing Fastai, Prophet and MultiOutputRegressor defaults

The first experiment involved training models with different combinations of data sets and methods. The aim was to initially try out the default configurations of the choice libraries and see how they perform with hourly, 3-hourly and 6-hourly MTR predictions before attempting any hyperparameter tuning. Out of the box, fastai permits the customisation of the number of features to predict and this can be set to as many as possible if a custom loss function is configured alongside. The default design of fastai's tabular learner (a class within its mid-level API) is a two-layered neural network with 200 neurons in the first layer and 100 in the second layer. Other fastai default parameters and values are shown on Table 4.

Prophet uses a conventional time-series method of forecasting and requires just the date column and one dependent variable (y). However, for this experiment we made use of

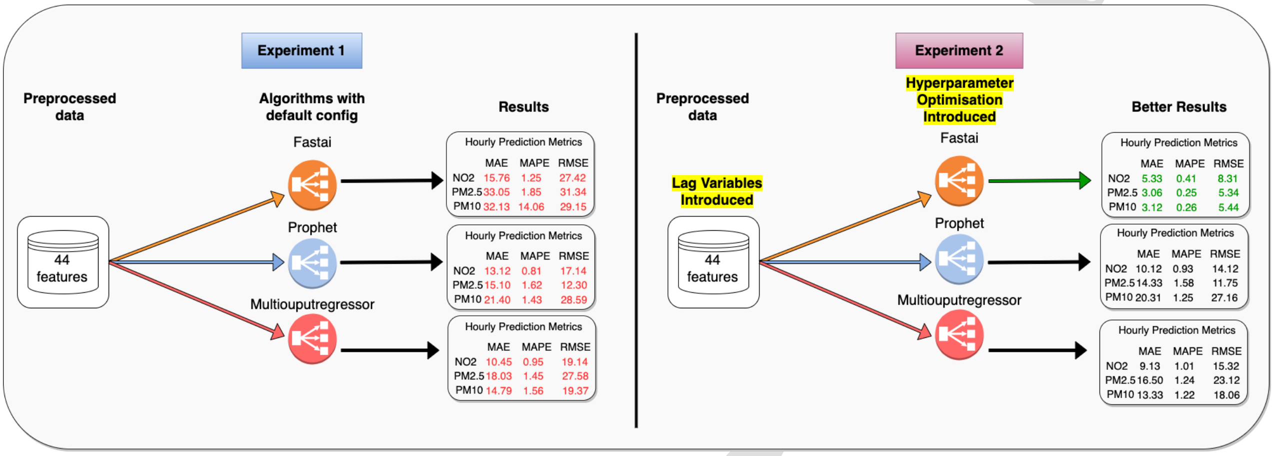


Figure 11: Summary of experiments carried out in this study.

Table 4: Hyperparameters used for experiment 1 - default configurations

Algorithm	**Hyperparameter name**	**Hyperparameter value**
Fastai	Number of layers	2
	First layer neurons	200
	Second layer neurons	100
	Dropout probability	0.04
	Learning rate	$1e^{-1}$
Prophet	Period	365
	Changepoint prior scale	0.001
MultiOutputRegressor	Number of estimators	100
	Learning rate	0.1
	Max depth	3
	Minimum samples split	2
	Minimum samples leaf	1
	Alpha	0.9

the library's *add_regressor* function to include the other features but it was required that the historic and future values of these additional regressors be included during training. Since prophet does not support multi-output forecasting by default, we made use of another package called *multi-prophet* which allowed us to predict all three pollutants simultaneously. Also, UK holiday effects were captured using the built-in country holidays feature.

Randomforestregressor, gradientboostingregressor and kneighboursregressor were explored with the MultiOutputRegressor to see which performed better. The best performing regressor with the default configurations was to then be used for subsequent experiments. Gradientboostingregressor produced the best result when compared in terms of the mean absolute error (MAE). The default configuration used is shown in Table 4. The result of experiment 1 is reported in section 5 but overall, it showed that most of the models did not perform too well and more experimentation or parameter optimisation was required.

Table 5: Details of Hyperparameters optimised using Optuna and GridSearchCV

Optimiser	**Hyperparameter**	**Search space**	**Result**
Optuna	Number of layers	(1,7)	3
	Neurons per layer	(50,200)	200,162,134
	Weight decay	(0.01,0.1)	0.01
	Learning rate	($1e^{-5}$,$1e^{-1}$)	$1e^{-3}$
	Dropout probability	($1e^{-3}$,$1e^{-1}$)	0.2
GridSearchCV	Number of estimators	(10,300)	250
	Learning rate	($1e^{-5}$,$1e^{-1}$)	$1e^{-1}$
	Max depth	(1,40)	12
	Minimum samples split	(0.01,1)	0.6
	Alpha	(0.1,2)	1.3

4.2. Hyperparameter tuning with optuna and gridsearchcv

Following the not-so-impressive results of experiment 1, it was essential that the training parameters were optimised. Optuna is a mildly famous parameter optimisation framework for deep learning models. It was chosen for the purpose of this study due to its ease of use and also its recently introduced integration module for fastai. Optuna requires the definition of an objective function to be optimised, and in our case was defined as the model's prediction of the three pollutants. Table 5 shows the search space for each of the optimised hyperparameter and the associated value after 50 optuna trials. GridSearchCV is an estimator within the sklearn library used to carry out brute force parameter search on regression algorithms such as the one being explored in this study. The technique uses cross-validation for this purpose while fitting and scoring each fold independently. GridSearchCV was used to optimise the number of estimators, learning rate, max depth, minimum sample split and alpha values for the gradientboostingregressor algorithm. Table 5 also shows the selected hyperparameter values after optimisation.

4.3. Experiment 2 - exploring lagged dependent variables (LDVs)

This experiment sought better model performance through the introduction of lagged variables. Introducing lagged variables in regression analysis is not new as discussed in the study of Wilkins (2018). The concept has been explored in several studies including air quality research with some scholars arguing that it may introduce bias in the data set if not defined properly (Grubb & Symons 1987). In this study we implemented the concept by carefully creating a structured data set which contained actual readings from previous time points leading to the current time point to be predicted. Each of these time points were depicted as separate columns and fed into each model to be trained. The effect of this experiment was that information of the previous time points needed to be provided for any future time point. This was the sensitive bit that could easily lead to data leakage. A function was therefore written to implement this idea while sequentially predicting all the timing points leading to the current one. Results of experiment 2 are also reported in section 5 and it shows an improvement from the previous experiment.

5. Model Validation and Results

This section highlights results of the experiments carried out in this study. Details of the choice evaluation metrics and the methods used to select our validation data are also highlighted.

5.1. Performance Metrics

Evaluation metrics are used to check the performance of models during and after training. Hence, it was necessary that suitable metrics for MTRs were first chosen even before training was started. More importantly, the metrics were also used to validate our models to make sure they were actually learning. Existing regression studies adopt metrics such as mean squared error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE) and mean square error (MSE) for model evaluation. Equations 3 to 5 illustrate the MAE, RMSE and MAPE metrics that were chosen as performance measures where y is the actual value and $\hat{y}_i$ is the predicted value. For fastai, a custom loss function that could compute the model's performance for each pollutant, average it and then update the model's weights accordingly was implemented. This was an important step to force the model to learn appropriately and not perform exceptionally on one pollutant and poorly on another.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i}| \quad (4)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2} \quad (5)$$

5.2. Test and Validation Data

70% of the entire data set was used for training while the remaining 30% was split into validation (20%) and test(10%) sets. However, the data had to be first sorted by date and then split by index to ensure no randomisation occurred and that seasonality within the data was maintained. As a result, 8953 rows were used for training, 2,398 rows for validation and 1,199 rows for testing. In days, this translated to 39 days for validation and 27 days for test. Each datapoint represents hourly reading for all 44 features. The validation set was used to optimise models' parameters after each training loop while the test set was used to evaluate the performance of the final model. Cross validation is one of the widely adopted validation methods in regression analysis (Morin & Davis 2017). Hence, the method was chosen for validating and testing the accuracy of the trained models. The implementation was different for all three algorithms but this generally meant that once the training was completed on the initial 8,953 rows, the model's performance is examined on the validation set, then a specified chunk of data is taken from the validation set and then used to train the model again and its performance evaluated on the remaining chunk. This process is repeated till there is no chunk left to cross validate with. For prophet, this chunk is referred to as the *period* while the number of days to be predicted is referred to as *horizon*. Sklearn's *cross_val_score* helper function was used to cross validate the fastai and multioutputregressor models. The horizon was successively set at 1h, 8h, 16h and 24h for different validation rounds while the period was set to hourly.

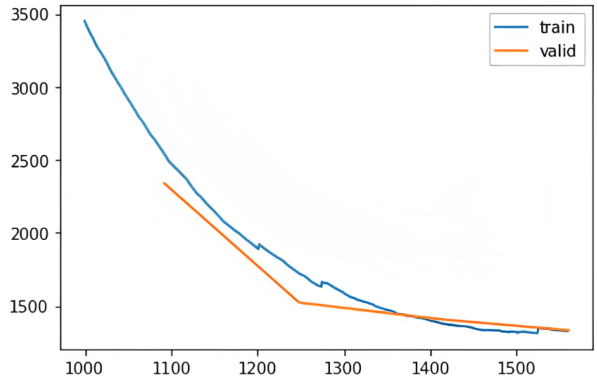
5.3. Experiment 1 Results

Models trained in the first experiment were evaluated over an hourly, 8-hourly, 16-hourly and 24-hourly timestep. These timesteps were chosen based on similar AQ studies that have also evaluated their models using the same method (Bui et al. 2018, Mao et al. 2021). figure 12a shows the training and validation loss for fastai after 1,500 epochs. From the plot, it can be seen that the training loss reduced progressively but this was not indicative of the final evaluation results shown in Table 6. The table shows the scores recorded for each algorithm in each timestep. It is evident that all the models struggled with the 24hr and 16hr predictions and performed slightly better with the hourly and 8hr predictions. The overall minimum MAE, MAPE and RMSE 1hr scores for NO_2 in this experiment was 10.452, 0.952, 19.145 respectively with the multioutputregressor model. Likewise, the best performance for $PM_{2.5}$ was on the prophet model with 15.103, 1.623 and 12.304 scores. For the most part, fastai recorded the worst performance in this experiment with scores as high as 40.099, 2.512 and 38.146. To further strengthen our assumptions that the scores recorded on these

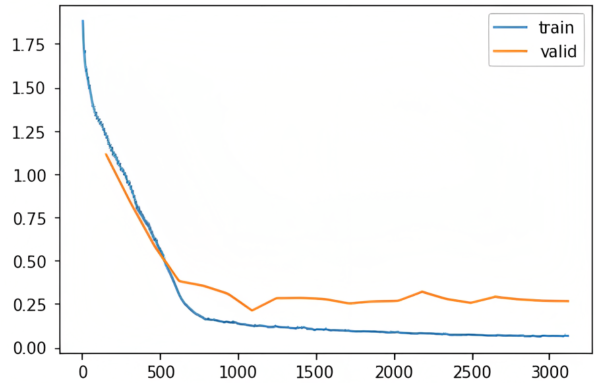
Table 6: Experiment 1 results of MTR models prediction for different timesteps

Pollutant & Timestep		Fastai			Multioutputregressor			Prophet		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
NO_2 (ppb)	1hr	15.760	1.256	27.420	**10.452**	**0.952**	**19.145**	13.128	0.811	17.142
	8hr	16.321	1.076	31.329	17.334	1.772	21.768	14.372	0.816	20.099
	16hr	18.167	1.321	34.771	21.982	2.306	24.911	14.714	0.852	23.146
	24hr	21.159	1.442	35.682	23.057	2.512	21.156	15.591	0.994	26.044
$PM_{2.5}$ ($\mu g/m^3$)	1hr	33.051	1.858	31.341	18.036	1.452	27.588	**15.103**	**1.623**	**12.304**
	8hr	34.111	2.328	33.142	23.911	1.641	33.612	19.145	1.815	18.142
	16hr	38.440	2.416	36.189	27.105	1.952	35.145	10.232	2.012	22.356
	24hr	40.099	2.512	38.146	26.830	1.835	36.875	15.344	2.458	23.198
PM_{10} ($\mu g/m^3$)	1hr	32.130	14.063	29.156	**14.798**	**1.568**	**19.376**	21.403	1.434	28.599
	8hr	37.156	7.342	31.002	18.233	1.734	22.157	20.123	2.583	32.048
	16hr	38.360	10.222	35.158	21.156	1.912	28.523	22.041	5.168	37.145
	24hr	33.127	8.066	36.360	24.076	1.820	32.142	23.487	3.443	33.640

models were too high, a graphical plot of the actual readings and models' predictions were made as illustrated in figures 13-15. None of the models were able to perform well on all three pollutants simultaneously. An ensemble of predictions from the two better performing models - multiouputregressor and prophet was also explored but there was no improvement with the achieved scores.

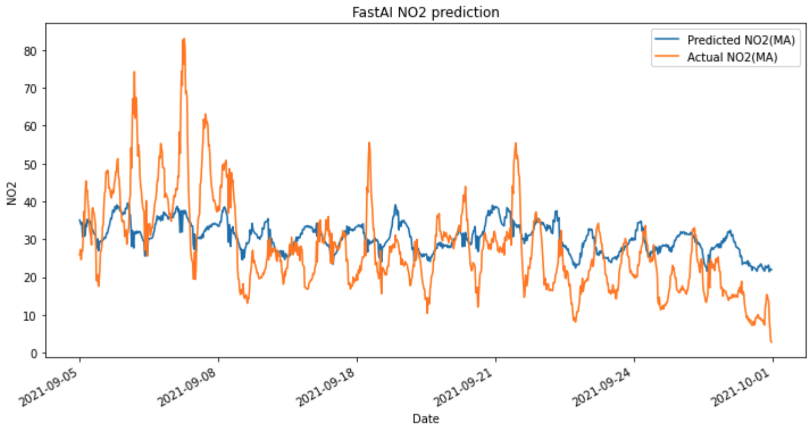


(a) Experiment 1 - Fastai's training and validation loss after 1500 epochs.

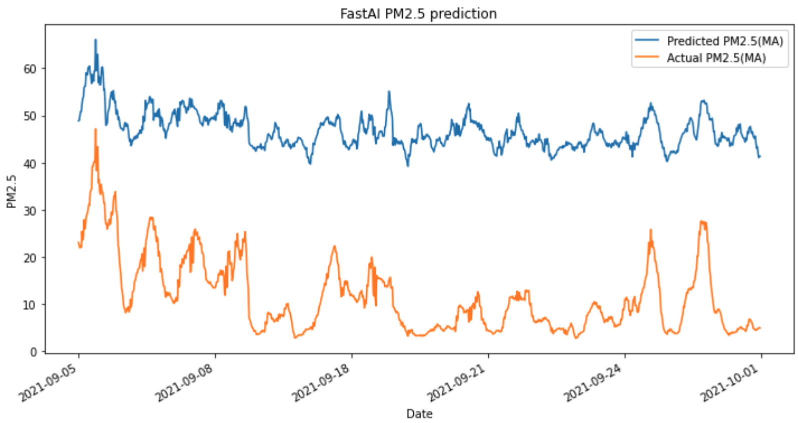


(b) Experiment 2 - Fastai's training and validation loss after 3000 epochs.

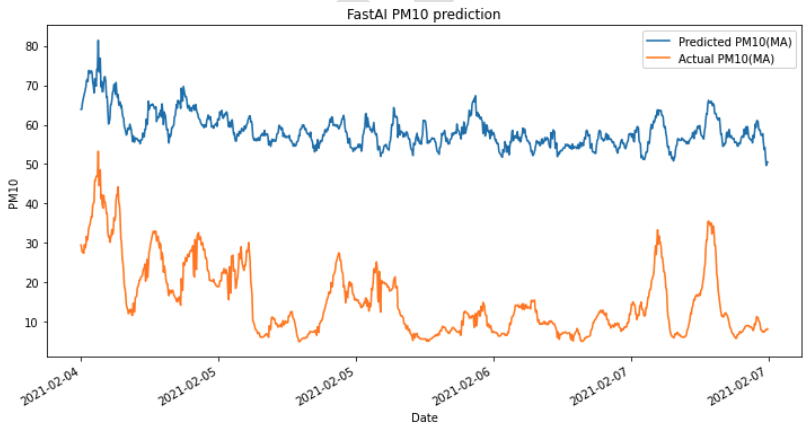
Figure 12: Training and validation losses on Fastai after 1500 and 3000 epochs for experiments 1 and 2 respectively.



(a) Predicted vs Actual hourly NO2 readings.

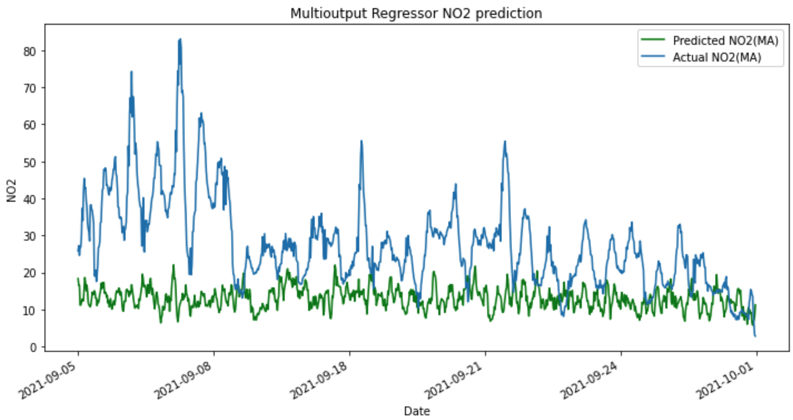


(b) Predicted vs Actual hourly PM2.5 readings.

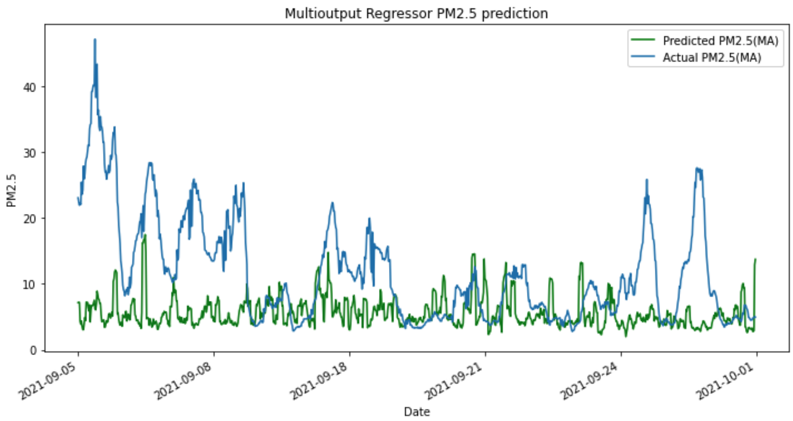


(c) Predicted vs Actual hourly PM10 readings.

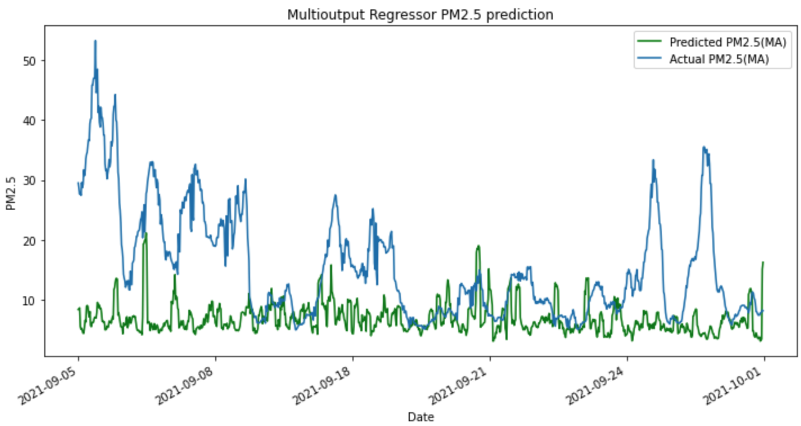
Figure 13: Experiment 1 - Fastai's model predictions.



(a) Predicted vs Actual hourly NO2 readings.

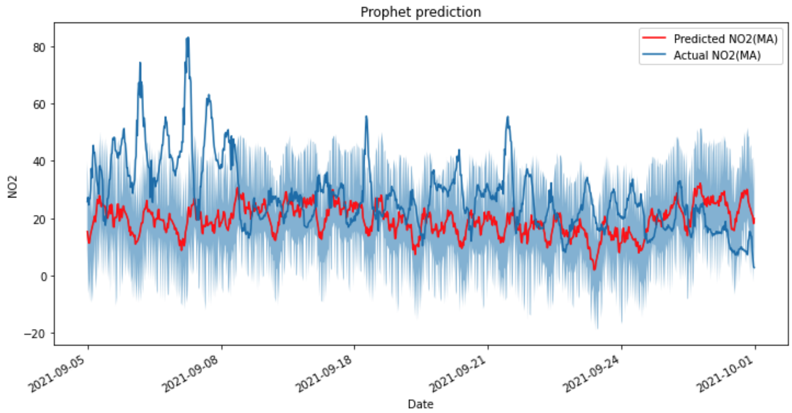


(b) Predicted vs Actual hourly PM2.5 readings.

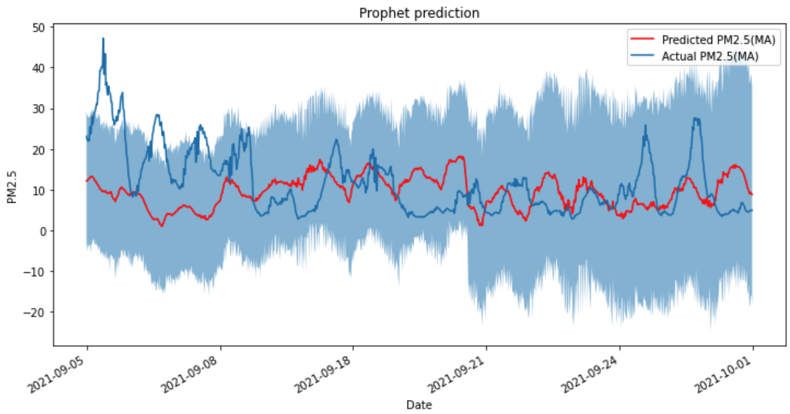


(c) Predicted vs Actual hourly PM10 readings.

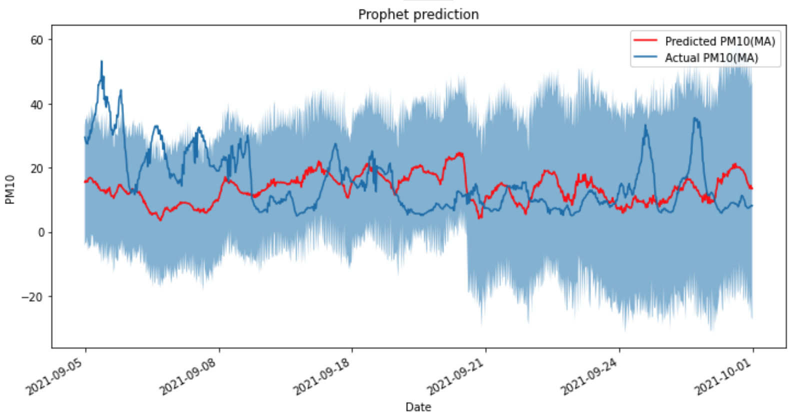
Figure 14: Experiment 1 - MultiOutputRegressor's model predictions.



(a) Predicted vs Actual hourly NO2 concentration levels.



(b) Predicted vs Actual hourly PM2.5 concentration levels.



(c) Predicted vs Actual hourly PM10 concentration levels.

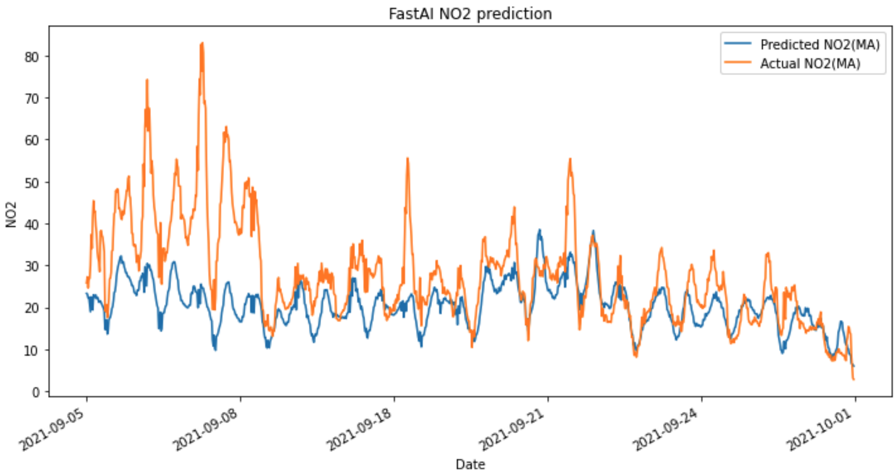
Figure 15: Experiment 1 - Prophet's model predictions.

Table 7: Experiment 2 results of MTR models prediction for different timesteps

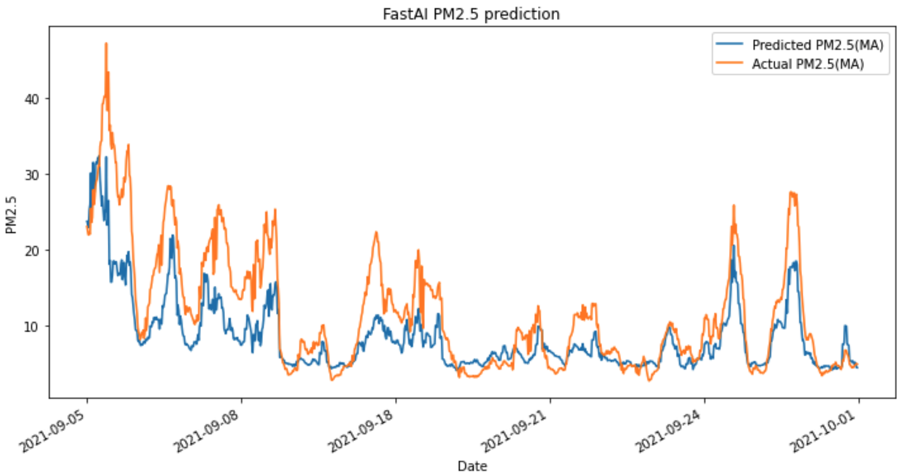
Pollutant & Timestep		Fastai			Multioutputregressor			Prophet		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
NO_2	1hr	**5.333**	**0.412**	**8.312**	9.132	1.012	15.325	10.122	0.931	14.122
	8hr	7.182	0.676	9.042	13.562	1.622	19.328	13.306	0.826	19.059
	16hr	6.325	0.521	8.763	20.152	2.133	22.541	14.334	0.782	22.326
	24hr	8.058	0.731	10.324	22.034	2.262	20.331	15.591	0.924	24.134
$PM_{2.5}$	1hr	**3.062**	**0.258**	**5.341**	16.506	1.243	23.124	14.332	1.589	11.752
	8hr	4.251	0.328	4.142	21.121	1.476	33.612	18.032	1.629	16.302
	16hr	4.430	0.399	5.189	23.105	1.432	35.145	9.112	1.892	20.126
	24hr	5.639	0.435	6.146	22.498	1.835	36.875	13.763	2.298	21.156
PM_{10}	1hr	**3.124**	**0.267**	**5.443**	13.332	1.228	18.069	20.313	1.254	27.169
	8hr	4.022	0.354	4.783	18.023	1.734	21.100	19.523	2.383	30.124
	16hr	4.129	0.378	5.034	19.326	1.912	26.613	20.376	4.198	32.225
	24hr	5.123	0.462	6.343	21.312	1.820	31.298	21.809	3.213	31.004

5.4. Experiment 2 Results

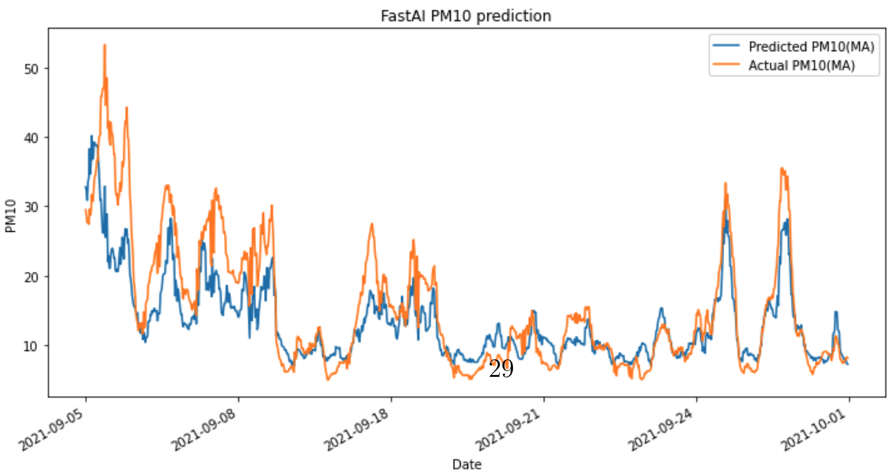
There was an immediately noticeable improvement in the results obtained in experiment 2. The metrics scores dropped considerably for the fastai model while the multioutputregressor and prophet models also saw some improvements. The best scores were recorded by fastai in this round of experiment for all three pollutants simultaneously. Although the model in this experiment was run for 1,500 more epochs than experiment 1, this was not the reason for the improved scores. The first experiment was only run for shorter epochs to avoid overfitting since the validation and training losses were not reducing as the epochs increased. A plot of the validation loss illustrated in figure 12b shows that the loss from this experiment was lower from the beginning and reduced in a stable manner as compared to experiment 1. The model's worst performance was on NO_2 24hr predictions with MAE as high as 8.058. However, this result still outperforms the previous NO_2 results for all the models in experiment 1. From Table 7, it is hard to determine the model's best prediction performance since the results for $PM_{2.5}$ and PM_{10} were quite similar on 1hr timestep predictions. The best average MAE, MAPE and RMSE scores was recorded as 3.062, 0.258 and 5.341 respectively. This improvement in the performance of the fastai model can be associated with the introduction of lagged variables as well as the hyperparameter tuning in this round of experiment. As illustrated in figures 17 and 18 and also Table 7, the prophet and multioutputregressor models also performed slightly better in this as a result of these changes but the improvement was not as significant as fastai's.



(a) Predicted vs Actual hourly NO2 concentration levels.

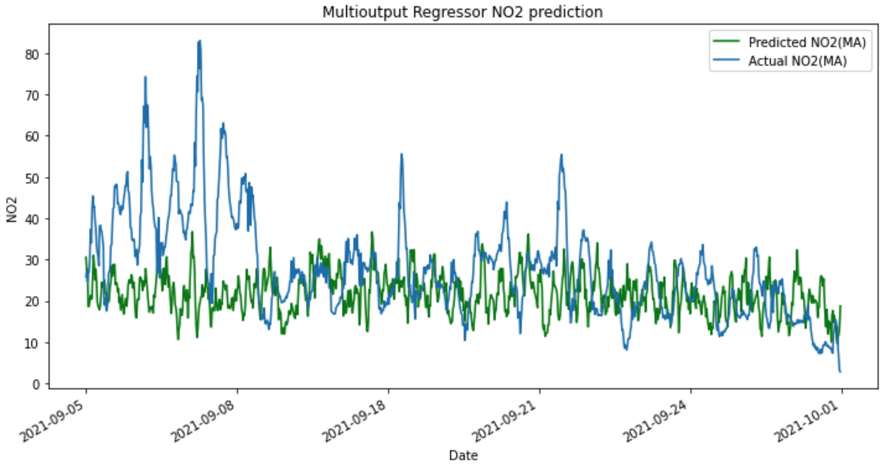


(b) Predicted vs Actual hourly PM2.5 concentration levels.

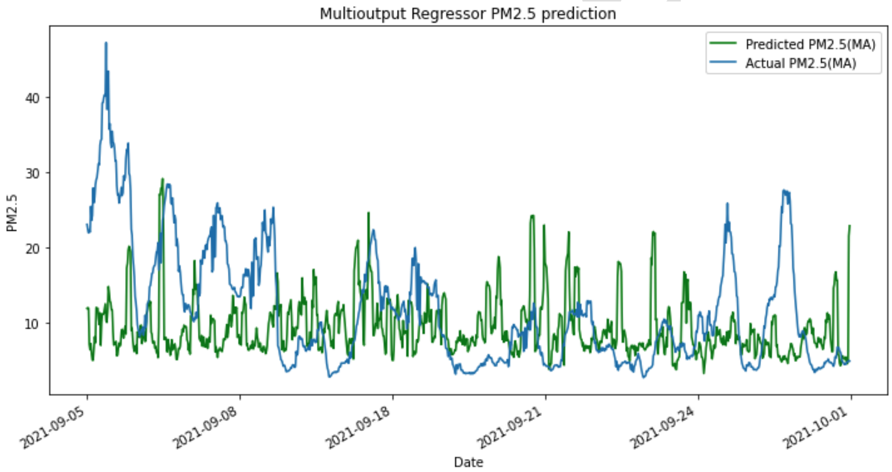


(c) Predicted vs Actual hourly PM10 concentration levels.

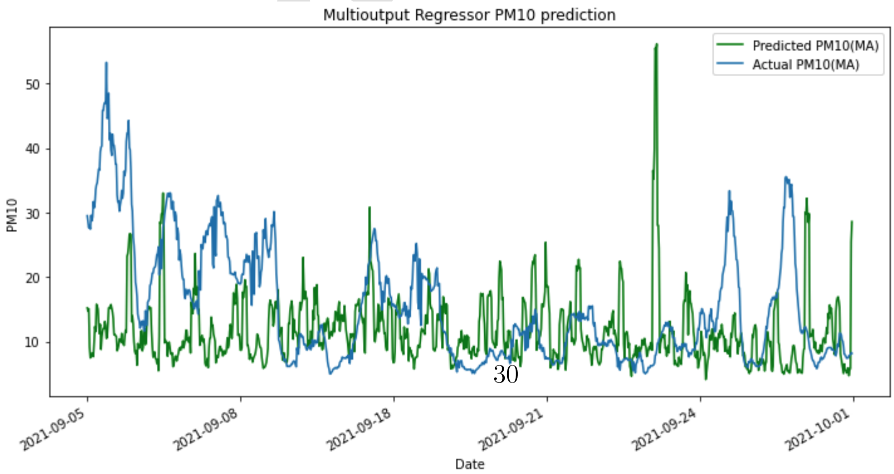
Figure 16: Experiment 2 - Fastai MTR predictions for NO_2,$PM_{2.5}$ and PM_{10}.



(a) Predicted vs Actual hourly NO2 concentration levels.

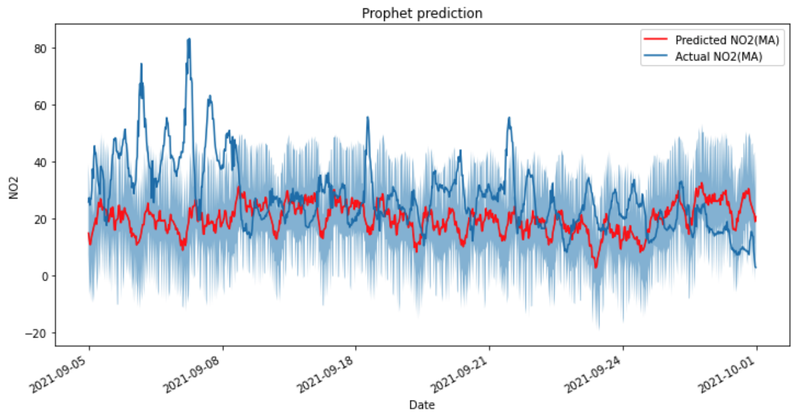


(b) Predicted vs Actual hourly PM2.5 concentration levels.

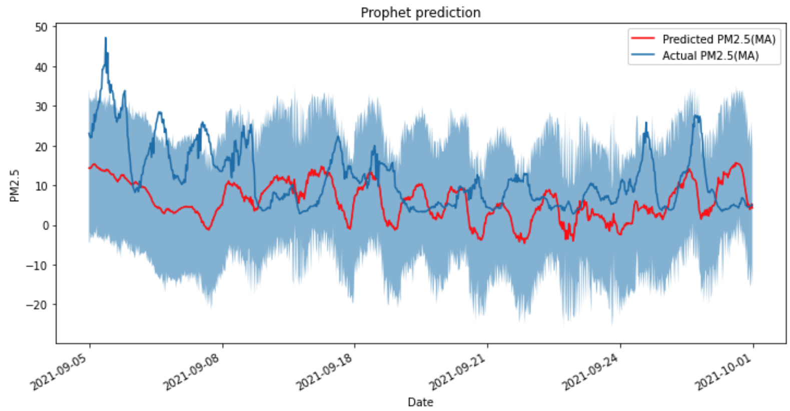


(c) Predicted vs Actual hourly PM10 concentration levels.

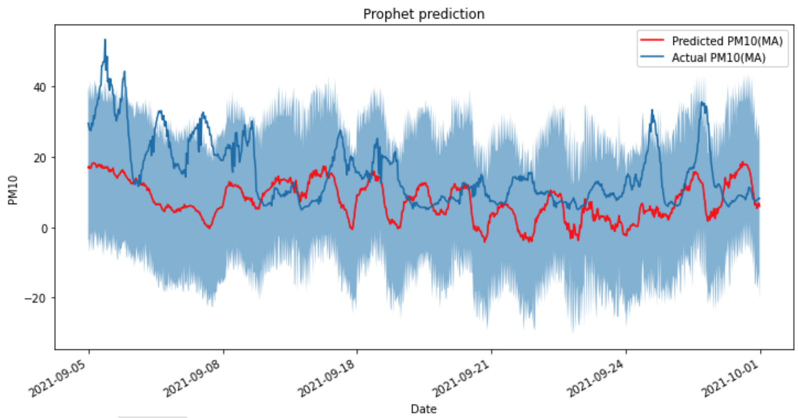
Figure 17: Experiment 2 - MultiOutputRegressor's MTR predictions for NO_2,$PM_{2.5}$ and PM_{10}.



(a) Predicted vs Actual hourly NO2 concentration levels.



(b) Predicted vs Actual hourly PM2.5 concentration levels.



(c) Predicted vs Actual hourly PM10 concentration levels.

Figure 18: Experiment 2 - Prophet's model predictions.

5.5. Statistical significance of results

To further strengthen the confidence in the results achieved with fastai, it was necessary that statistical hypothesis tests were carried out to weigh its performance against the two other models. The non-parametric Friedman and the Wilcoxon signed-rank test were selected with a null hypothesis (H_o) that there is no statistical difference between the predictions from the three models. This hypothesis would be rejected if the chi-square was ¿ 3.84 for the Friedman test and p-value was below 0.05 for both tests. Both tests were performed on 20 MAE, MAPE and RMSE error readings from cross-validation in experiment 2. The Friedman test for the 3 models resulted in a chi-square score of 6.45 and p-value of 0.03. Table 7 shows the result of the Wilcoxon test for pair-wise comparisons of the models. Just like the Friedman test, all the p-value scores were less than 0.05. The result of both statistical tests indicates that the hypothesis can be rejected and the predictions from fastai are statistically different from the multiouputregressor and prophet models.

Table 8: Statistical significance and model evaluation using Wilcoxon signed rank test

Pair-wise comparison	P-value	Significance
Fastai and Prophet	0.02	Yes
Multioutputregressor and Prophet	0.03	Yes
Fastai and Multioutputregressor	0.02	Yes

5.6. Results comparison with related work

Although there have been lots of studies focused on traffic related air pollution prediction, very few have looked into multi-target prediction of pollutants or the combination of data set used in this study. Similarly, the evaluation metric and validation approach in some of these studies are different from the ones explored in this study. For these reasons, it was unfeasible to make a direct comparison of the results of our proposed approach with existing ones. Nevertheless, the results of individual predictions for pollutants same as ours in a select few studies were compared with the result of our proposed approach. Table 8 shows the outcome of this comparison with our approach outperforming most of the reviewed studies. The study of (Suleiman et al. 2019) outperformed ours in $PM_{2.5}$ predictions but the validation approach used by the authors was different. The improved performance achieved with our approach can be attributed to the use of additional data for training and the adoption of categorical embeddings.

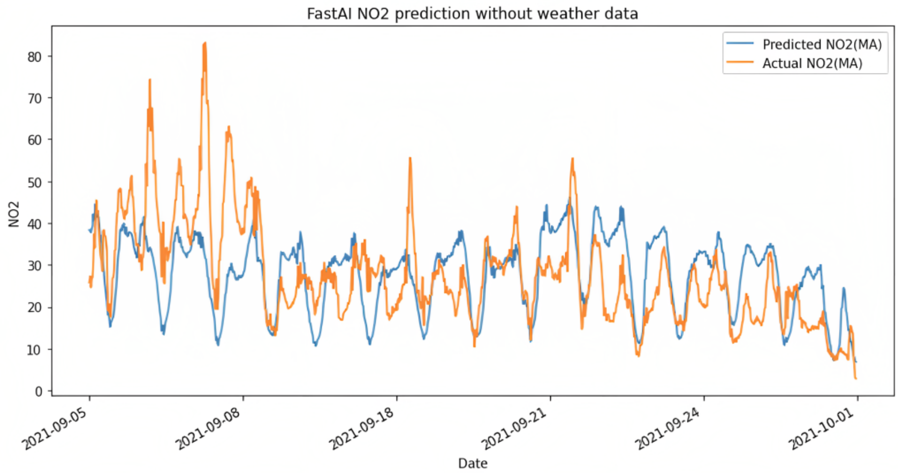
5.7. Model's performance on missing data

An additional test was carried out to evaluate the performance of the fastai model from experiment 2 in a real-life scenario where some of the integrated data might be missing. It is suggested that as much data as possible is sourced to get optimum performance, but this may not always be the case. To replicate this scenario, the values for the intended missing

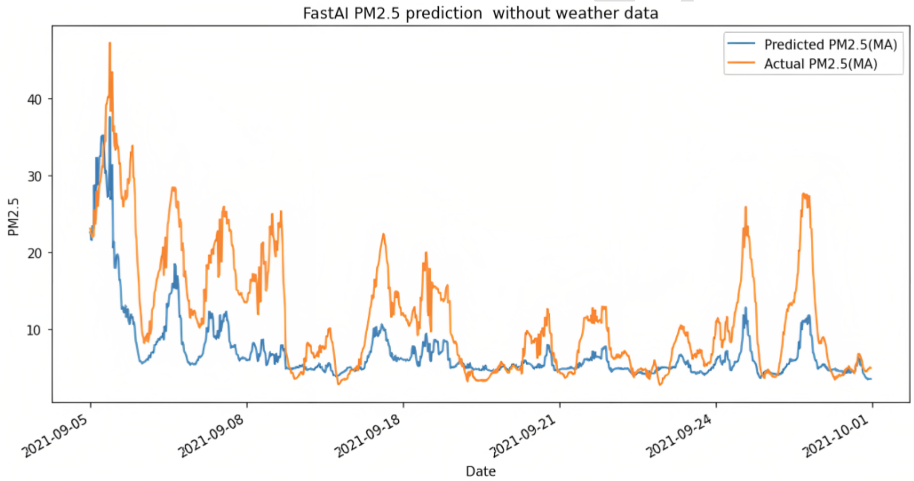
Table 9: Comparison of prediction results with existing studies based on RMSE score

Reference	Data Source	Method	Pollutant	RMSE (lowest)
Cabaneros et al. (2017)	Marlyeborne Road Monitoring sites	Hybrid Artificial Neural Networks	NO_2	22.05
Suleiman et al. (2019)	Monitoring sites	Artificial Neural Network, SVM, BRT	$PM_{2.5}$	4.67
			PM_{10}	10.05
Li et al. (2020)	Hong Kong Roadside station	SVM, GAM, XGBoost, RF, BRT	$PM_{2.5}$	7.90
			NO_x	30
Jida et al. (2021)	Aeroqual AQ sensor	Artificial Neural Network	$PM_{2.5}$	8.45
			PM_{10}	12.42
Wu et al. (2022)	Shanghai Roadside stations	Neural Networks - LSTM	NO_2	9.61
Mengara Mengara et al. (2022)	South Korea Roadside stations	LSTM, Auto Encoder, Convolutional Neural Networks	$PM_{2.5}$	7.40
			PM_{10}	9.81
Proposed Method	**REVIS sensors and integrated data**	**Deep Learning + Categorical Embeddings**	NO_2	**8.31**
			$PM_{2.5}$	**5.34**
			PM_{10}	**5.44**

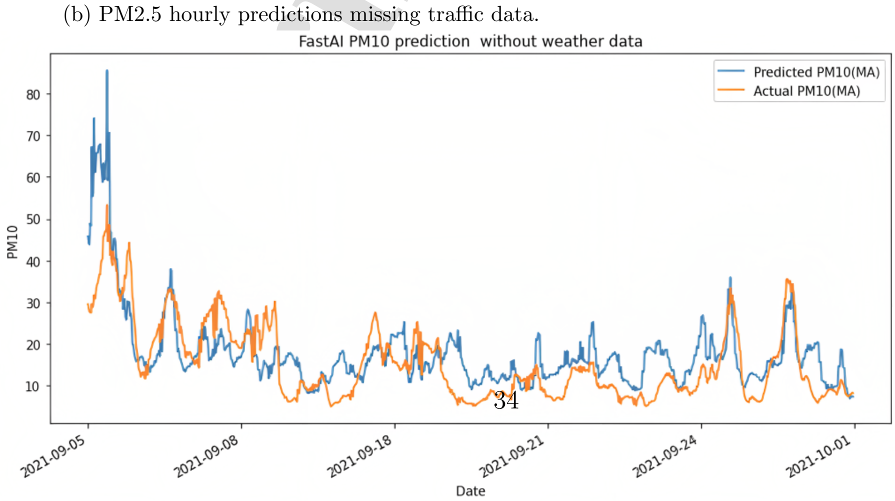
data were replaced with zeros in the test data before model inferencing. It was important to not drop the columns entirely since the model was originally trained on 44 features and dropping them would result in errors. Similarly, replacing with Nan instead of zeros results in errors too. The model's predictive performance when traffic, weather, emissions factor, background concentration or elevation data are missing can be seen on figures 19-23. The illustrations indicate varying predictive accuracy depending on the missing data. The model's performance is worse when weather data is missing and poor when elevation or background concentration data are missing. NO_2 prediction is the most affected in these missing data scenarios. This performance variation with certain missing data begs the question - *What are the most important features that must be captured for a reasonable prediction accuracy?*



(a) NO2 hourly predictions missing traffic data.

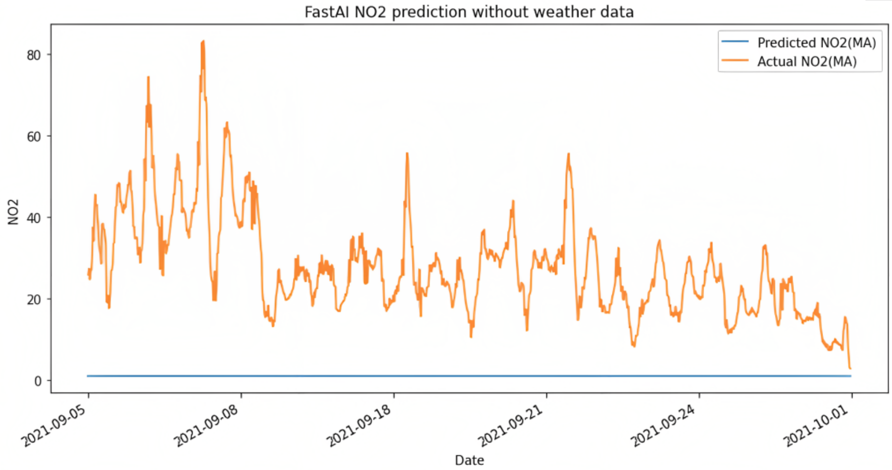


(b) PM2.5 hourly predictions missing traffic data.

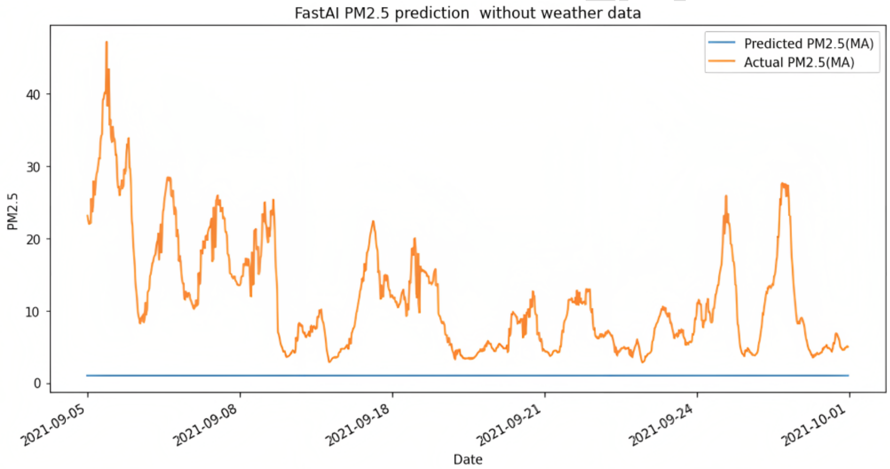


(c) PM10 hourly predictions missing traffic data.

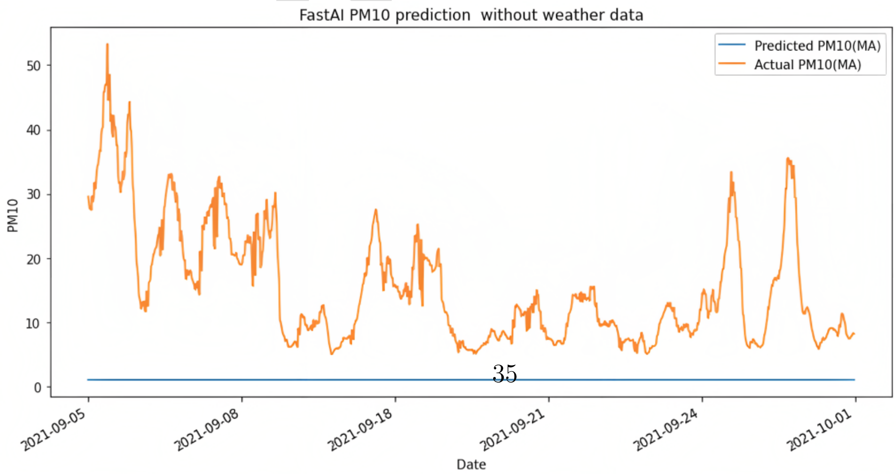
Figure 19: Fastai model's performance when missing traffic data.



(a) NO2 hourly predictions missing weather data.

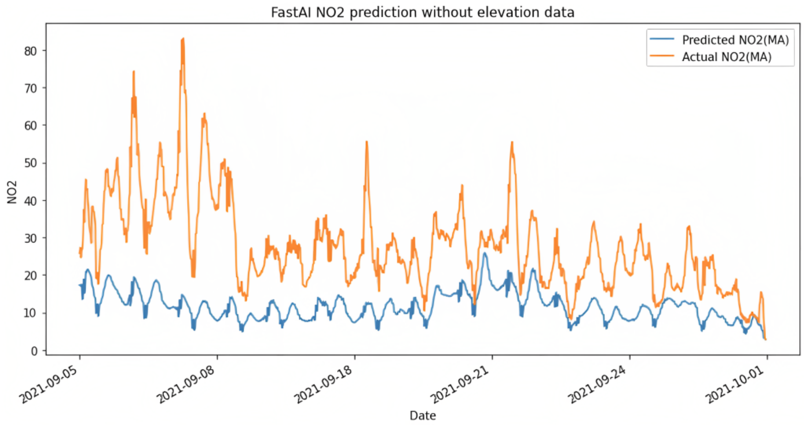


(b) PM2.5 hourly predictions missing weather data.

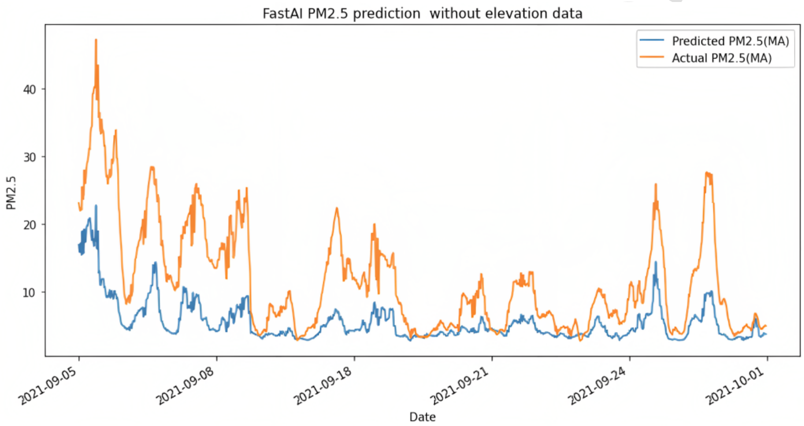


(c) PM10 hourly predictions missing weather data.

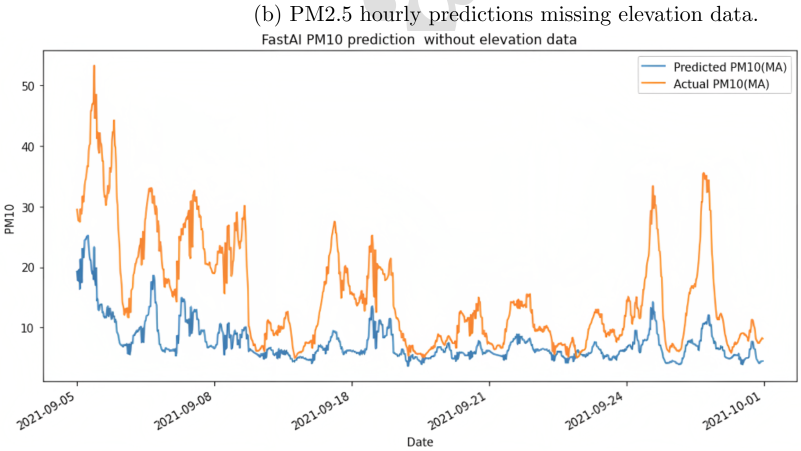
Figure 20: Fastai model's performance when missing weather data



(a) NO2 hourly predictions missing elevation data.

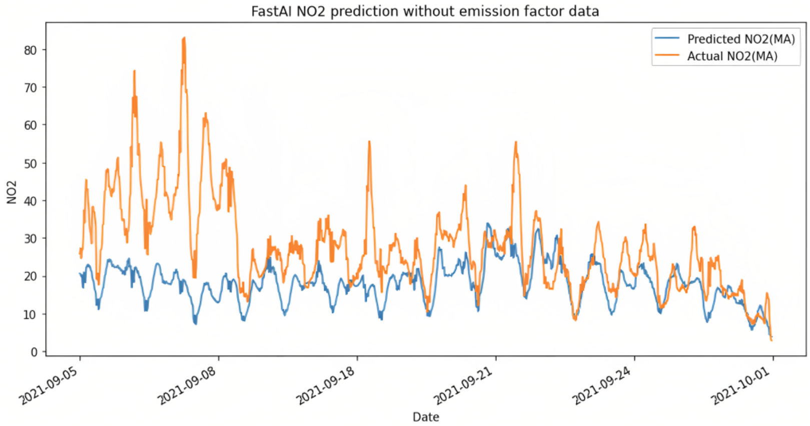


(b) PM2.5 hourly predictions missing elevation data.

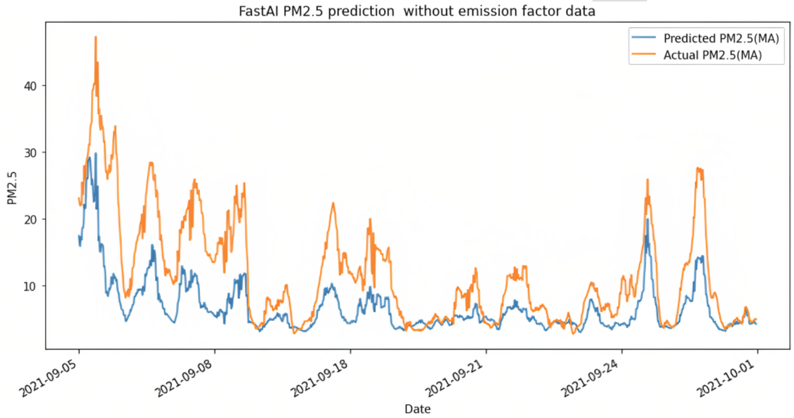


(c) PM10 hourly predictions missing elevation data.

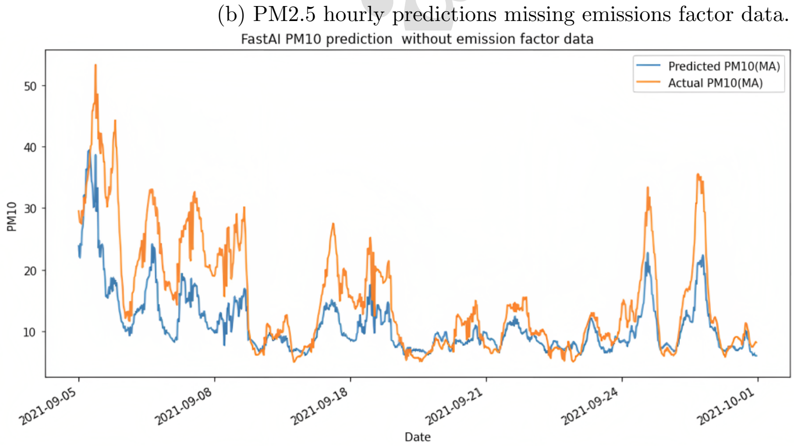
Figure 21: Fastai model's performance when missing elevation data



(a) NO2 hourly predictions missing emissions factor data.

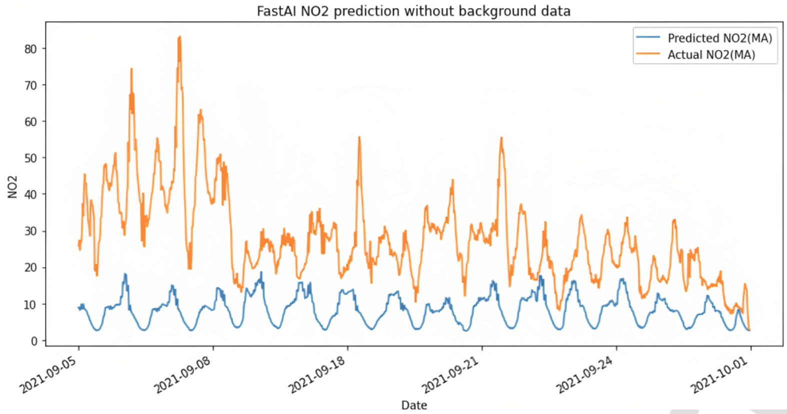


(b) PM2.5 hourly predictions missing emissions factor data.

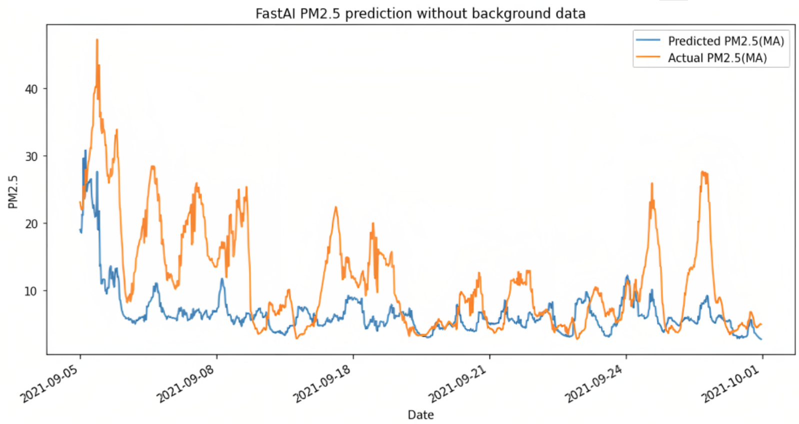


(c) PM10 hourly predictions missing emissions factor data.

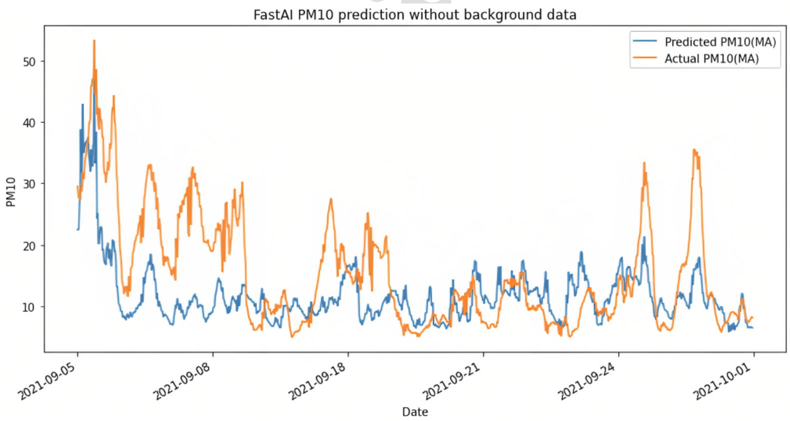
Figure 22: Fastai model's performance when missing emissions factor data



(a) NO2 hourly predictions missing background concentration data.



(b) PM2.5 hourly predictions missing background concentration data.



(c) PM10 hourly predictions missing background concentration data.

Figure 23: Fastai model's performance when missing background concentration data

6. Feature importance on best model results

Following the improvement of fastai model's performance in experiment 2, further investigation was carried out to understand which of the input parameters were the most influential in the model's predictions. This section highlights the outcome of this analysis.

6.1. Fewer features, same accuracy

Machine learning models developed with advanced algorithms such as deep learning are considered black box models (Akinosho et al. 2020). This is as a result of the complexities involved in understanding what happens behind the scenes for most of these models. It is particularly important in the air quality domain to highlight the main contributors to pollution through this kind of understanding. Thankfully, various tools are now available to make models explanable and fastai's *Interpretation* classes further facilitate this task. A feature importance plot as shown in figure 24 was plotted using one of these tools and this gave many insights into which of the 44 input parameters were the least and most contributing. From the plot it is observable that 'LGV Count', 'Other Avg speed', 'Bus Count', 'Wind Direction', 'Car Count', 'HGV Count', 'NO2 emission factor' and 'DATETimeHour' were the most influential features. These are mainly traffic parameters except the 'Wind Direction' and 'DATETimeHour' features. All the additional date variables that were added to the data set had none to little impact with some even recording negative importance. Similarly, 'highway elevation', 'background NO2' and other weather parameters were not important for the model's predictions. The fastai model was retrained while dropping these low and negative influencing parameters to see if its performance would be any different and if the feature importance will be reshuffled.

Figure 25 shows the feature importance after retraining on just the top 12 features from experiment 2. The model's accuracy remained similar to what was achieved in experiment 2 but the feature importance was reorganised. It can be noticed that most of the traffic parameters maintained the top spot with only *car count* dropping behind. The date parameter were also influential with the hour of the day having the highest influence. The wind direction and NO_2 emission factor features dropped to the bottom of the list in this round. However, it is worth reiterating that these least influential features are only not so important for this minimised data set but had significant impact in the overall data set

6.2. Features Ablation Test

The result of running an ablation test on the fastai model to further corroborate the importance of the training features is illustrated in figure 26. The test was carried out by dropping each feature one at a time and then retraining the model on the remaining features to predict all three pollutants. The RMSE score on the test data for each pollutant was recorded once the model retraining process was complete and the model was cross validated. This score was then compared to the RMSE score when all the features were used. The x-axis on figure 26 represents each feature that was dropped while the y-axis represents the recorded RMSE score. It can be observed that the impact of dropping most of the additional date

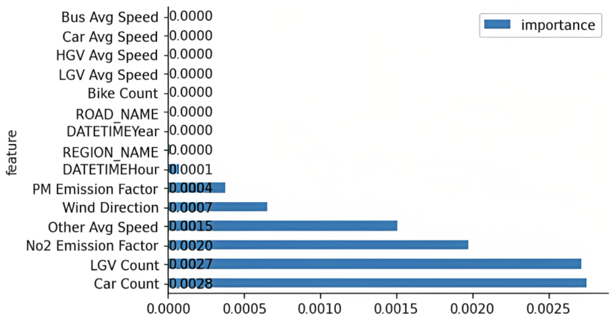


Figure 24: Feature importance from experiment 2. Traffic features including 'LGV count' and 'car count', 'average speed' were in the top list with the hour of the day, 'wind direction', 'PM emission factor' and 'No2 emission factor' also part of this list. Some of the least influential parameters were 'bike count', minute of the day and similar date parameters.

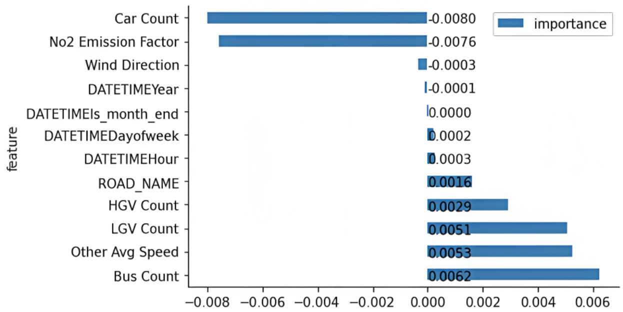


Figure 25: Feature importance after retraining on the top twelve features from experiment 2. All the traffic features except 'car count' maintained the top spot while 'wind direction' and 'No2 emission factor' dropped further down the importance list.

parameters was almost non-significant except for the hour parameter. Similarly, dropping the weather parameters, background pollution data and traffic parameters all resulted in a significant increase in the RMSE score to a level that is almost similar to experiment 1. Removing the other features had less impact on the model's performance. The result of this ablation test corresponds with the feature importance from the previous section where traffic and weather parameters were highlighted as important.

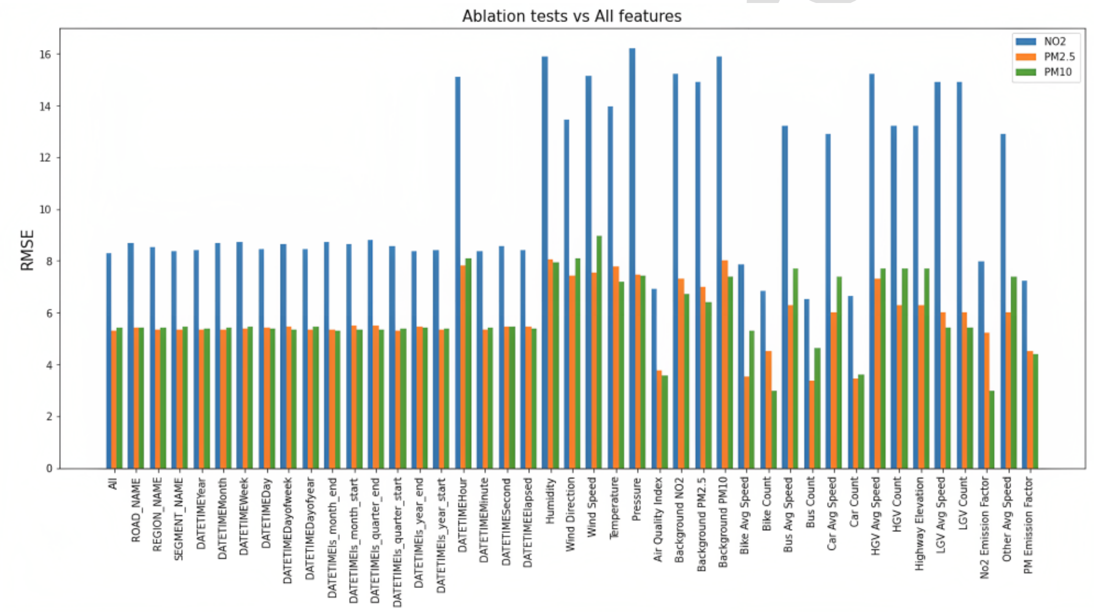


Figure 26: Feature ablation test to reveal features with the most impact on fastai model's predictions. The x-axis contains the feature list with each tick representing the feature that was removed when the model was retrained and RMSE score recalculated. The RMSE scores are represented on the y-axis. This chart indicates the importance of traffic and weather data as the RMSE scores increased when these features were removed from the data set.

7. Discussion and implication for practice

Currently there are several open-source and commercial traffic-related pollution modelling software available for different kinds of modelling and simulations. These software are considered robust and are largely adopted for local air quality management across the globe despite weaknesses such as the inability to integrate instantaneous data and retrain models on the fly (Forehead & Huynh 2018). The success of tools such as ADMS-Roads has been particularly linked to the incorporated data and explicit computation approach they use for important parameters (CERC 2022). With the growth of machine learning algorithms and demonstrable efficiency in the air quality domain (Wang et al. 2020), there is an excellent opportunity to emulate the kinds of data captured in these advanced modelling tools where available. The intrinsic computations and feature relationships can then be left to the algorithms to decipher for better accuracy. One immediate advantage of this approach is that it takes away the need for explicit parameter computation and can potentially address the limitation of model retraining using instantaneous data.

This study was able to integrate data from several sources albeit with some challenges. Only a portion of these data including historic pollution, some weather data and background concentration were publicly available. Extra research authorisation requests had to be carried out to access the rest. Traffic flow data especially was not within reach. The disparity in the data format for these data sets was another issue that had to be addressed using data integration maps. Similar fields from different sources had to be mapped together before integration was possible. These integrated data were then used to train models using three famous algorithms including deep learning, time-series and linear regression. It was important to demonstrate with these algorithms, if the forecasting performance of AI models with the newly curated data set are any close to what could be achieved using air quality modelling tools.

Our results show that just like any other machine learning task, sufficient hyperparameter tuning is required when training these models irrespective of the quality or type of data being used. Despite fastai's default incorporation of new deep learning techniques such as 'entity embeddings for categorical variables', the library's training parameters still needed to be tweaked for better results. The trained model was able to capture general pollution levels including rise in pollution and drop off but was not able to capture unpredictable peak events that could have been caused by specific occurrences such as an extra congestion. This is an indication that more features or peak events data can still be captured in the data set in order to model the specific causes of these peaks. Another approach is to tackle the prediction as a classification problem rather than a regression one. This will enable the use of advanced loss functions like *focal loss* which are designed to force an algorithm to learn rare trends in the data.

Since regular air quality review and assessment has now become a mandatory requirement for major cities across the globe (Zeng et al. 2019), this study could not have been carried out at a better time. From a social perspective, our proposed approach can help reduce traffic related pollution risks to citizens in different countries. There is evidence of increasing environmental injustice in developed countries where vulnerable citizens who are most susceptible to traffic pollution have less agencies in their area of residence (Barnes et al. 2019). An improved air quality management system backed by an accurate forecasting mechanism such as the one proposed in this study would enable government agencies to formulate targeted traffic restriction policies, provide early warnings on anticipated peak episodes and help spread its agencies to the most prone areas. Economically, the effect of air pollution has resulted in billions of dollars lost through healthcare provision or reduced yields from economically important agricultural crops in many countries (Pandya et al. 2022). A prediction system such as the one proposed in this study is not sufficient on its own to solve these economic problems but would have a significant input when integrated into existing air quality systems used for making informed decisions.

From a technological perspective, this study presents an opportunity for easily productionising air quality models for real-world use cases. The type of MTR models developed in this study solves the issue of deploying individual models for each pollutant of interest. Tools such as AWS lambda, Oracle ADS, mlflow are useful in automating this process and even provide more opportunities to get real-time predictions. One thing to be aware of when productionising MTR models is the possibility of *model (or concept) drift* which occurs when the environment becomes different from scenarios on which the model was trained leading to a depreciation in performance. One possible solution is to enable the automatic detection of these kinds of drifts and to put a process in place to retrain a model using updated data. The performance of the new models can then be compared with the already deployed model.

8. Conclusion

This study set out to contribute to existing body of air quality monitoring knowledge by investigating how additional data which are rarely integrated in TRAP forecasting could help improve accuracy. Unconventional training data for AI models such as terrain data, pollutants background concentration and emissions factor were integrated with the traditional traffic flow, weather and historic pollution data and used to train multi-target prediction models for NO_2, $PM_{2.5}$ and PM_{10}. The results of our experiment demonstrate the efficacy of the MTR models albeit with a lot of hyperparameter tuning required. The best performance was achieved with fastai on simultaneous hourly predictions for all three pollutants. The model performed well with $PM_{2.5}$ and PM_{10} and was able to capture peak episodes but struggled with similar spikes for NO_2. This indicates that the model was able to pick up the general trends of NO_2 pollution but struggled with localised pollution that resulted in peak episodes. We also evaluated key contributors to the model's performance and realised that traffic, weather, hour of the day and emission factor were at the top of the list. In conclusion, it is evident through this study that introducing additional highway features can effectively improve a model's prediction accuracy. However, there is still a persistent challenge of these models struggling with unusual spikes that are neither caused by transboundary air pollution effect or background pollution but by effects specifically localised to where the pollutant is being measured. Future research can look into these kind of scenarios and further investigate other pollutants and highway features that were not covered in this study.

Acknowledgement

The authors would like to express their sincere gratitude to InnovateUK (Grant Application No 10137 and File No 104367) for providing the financial support for this study.

Appendix A: List of attributes captured for MTR pollutant concentration forecasting

S/No	Column	Column Description	Range	Non-Null Count	Variable type
1	datetimehour	Hour variable extracted after preprocessing of datetime column	0-23	11990 non-null	Categorical
2	datetimeminute	Minute variable extracted after preprocessing of datetime column	0-59	11990 non-null	Categorical
3	datetimesecond	Second variable extracted after preprocessing of datetime column	0-59	11990 non-null	Categorical
4	datetimeelapsed	Time elapsed variable extracted after preprocessing of datetime column	1.60e+9-1.63e+9	11990 non-null	Continuous
5	datetimeyear	Year variable extracted after preprocessing of datetime column	2020-2021	11990 non-null	Categorical
6	datetimemonth	Month variable extracted after preprocessing of datetime column	1-11	11990 non-null	Categorical
7	datetimeweek	Week variable extracted after preprocessing of datetime column	1-47	11990 non-null	Categorical
8	datetimeday	Day variable extracted after preprocessing of datetime column	1-31	11990 non-null	Categorical
9	datetimedayofweek	Day of week variable extracted after preprocessing of datetime column	0-6	11990 non-null	Categorical
10	datetimedayofyear	Day of year variable extracted after preprocessing of datetime column	8-322	11990 non-null	Categorical
11	datetimeis_month_end	Boolean variable to indicate if the day is month end	0/1	11990 non-null	Categorical
12	datetimeis_month_start	Boolean variable to indicate if the day is start of the month	0/1	11990 non-null	Categorical
13	datetimeis_quarter_end	Boolean variable to indicate if the day is the end of a quarter	0/1	11990 non-null	Categorical
14	datetimeis_quarter_start	Boolean variable to indicate if the day is the start of a quarter	0/1	11990 non-null	Categorical
15	datetimeis_year_end	Boolean variable to indicate if the day is the start of the year	0/1	11990 non-null	Categorical
16	datetimeis_year_start	Boolean variable to indicate if the day is the end of the year	0/1	11990 non-null	Categorical
17	road_name	The name of the highway of interest	-	11990 non-null	Categorical
18	region_name	The name of the region where the highway is located	-	11990 non-null	Categorical
19	segment_name	The name of the highway segment where the IoT device is located	-	11990 non-null	Categorical
20	NO_2	Integrated average hourly NO_2 (ppb) reading from AURN station	0.63-132.37	11990 non-null	Continuous

21	$PM_{2.5}$	Captured $PM_{2.5}$ ($\mu g/m^3$) reading from REVIS IoT devices	0.69-401.01	10879 non-null	Continuous
22	PM_{10}	Captured PM_{10} ($\mu g/m^3$) reading from REVIS IoT devices	0.77-617.35	10879 non-null	Continuous
23	air_quality_index	The AQI for the highway segment of interest computed from the pollutant concentration readings	0-6.5	11990 non-null	Continuous
24	background_NO_2	The background NO_2 concentration for the highway segment of interest	8.06-27.99	11990 non-null	Continuous
25	background_$PM_{2.5}$	The background $PM_{2.5}$ concentration for the highway segment of interest	7.88-12.55	11990 non-null	Continuous
26	background_PM_{10}	The background PM_{10} concentration for the highway segment of interest	11.94-19.55	11990 non-null	Continuous
27	NO_2_emission__factor	Calculated NO_2 emission factor based on different vehicle types on the highway at that time point	0-14823	11990 non-null	Continuous
28	PM_emission_factor	Calculated PM_{10} emission factor based on different vehicle types on the highway at that time point	0-19982	11990 non-null	Continuous
29	bike_count	Captured bike count from REVIS IoT devices	-	6 non-null	Continuous
30	bike_avg_speed	Captured bike avg speed	-	6 non-null	Continuous
31	car_count	Integrated car count from TMU sites	0-3515	10949 non-null	Continuous
32	car_avg_speed	Captured car avg speed from REVIS IoT devices	-	6 non-null	Continuous
33	bus_count	Integrated bus count from TMU sites	0-412	10949 non-null	Continuous
34	bus_avg_speed	Integrated bus avg speed	-	6 non-null	Continuous
35	lgv_count	Integrated LGV count from TMU sites	0-245	10949 non-null	Continuous
36	lgv_avg_speed	Captured LGV avg speed from REVIS IoT devices	-	6 non-null	Continuous
37	hgv_count	Integrated HGV count from TMU sites	0-383	10949 non-null	Continuous
38	hgv_avg_speed	Captured HGV avg speed from REVIS IoT devices	-	6 non-null	Continuous
39	other_avg_speed	Integrated average travelling speed from TMU sites	0-76.25	10949 non-null	Continuous
40	humidity	Captured average hourly relative humidity from REVIS IoT devices (ϕ)	23.65-99.99	11990 non-null	Continuous
41	wind_speed	Integrated hourly modelled wind speed (knots) from AURN station	0-16.2	11990 non-null	Continuous
42	wind_direction	Integrated hourly modelled wind direction (true degrees) from AURN station	0-360	11990 non-null	Continuous
43	temperature	Captured average hourly temperature (°C) from REVIS IoT devices	-2.95-44.07	10879 non-null	Continuous

44	pressure	Captured average hourly pressure (hPa) from REVIS IoT devices	979.31-1042.72	10879 non-null	Continuous

Akinosho, T. D., Oyedele, L. O., Bilal, M., Ajayi, A. O., Delgado, M. D., Akinade, O. O. & Ahmed, A. A. (2020), 'Deep learning in the construction industry: A review of present status and future innovations', *Journal of Building Engineering* **32**, 101827.

Akinosho, T. D., Oyedele, L. O., Bilal, M., Barrera-Animas, A. Y., Gbadamosi, A.-Q. & Olawale, O. A. (2022), 'A scalable deep learning system for monitoring and forecasting pollutant concentration levels on uk highways', *Ecological Informatics* **69**, 101609.

Arunachalam, S., Valencia, A., Akita, Y., Serre, M. L., Omary, M., Garcia, V. & Isakov, V. (2014), 'A method for estimating urban background concentrations in support of hybrid air pollution modeling for environmental health studies', *International journal of environmental research and public health* **11**(10), 10518–10536.

Bálint, A., Fagerlind, H., Martinsson, J. & Holmqvist, K. (2014), 'Accident analysis for traffic safety aspects of high capacity transports'.

Barnes, J. H., Chatterton, T. J. & Longhurst, J. W. (2019), 'Emissions vs exposure: Increasing injustice from road traffic-related air pollution in the united kingdom', *Transportation research part D: transport and environment* **73**, 56–66.

Borchani, H., Varando, G., Bielza, C. & Larranaga, P. (2015), 'A survey on multi-output regression', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **5**(5), 216–233.

Bui, T.-C., Le, V.-D. & Cha, S.-K. (2018), 'A deep learning approach for forecasting air pollution in south korea using lstm', *arXiv preprint arXiv:1804.07891* .

Cabaneros, S. M. S., Calautit, J. K. S. & Hughes, B. R. (2017), 'Hybrid artificial neural network models for effective prediction and mitigation of urban roadside no2 pollution', *Energy Procedia* **142**, 3524–3530.

CERC (2019), 'Cambridge environmental research consultants - detailed modelling of no2 at market hill aqma, maldon'.

CERC (2022), 'Cambrige environmental research consultants - adms-roads comprehensive software for modelling road traffic pollution'.
URL: *http://www.cerc.co.uk/environmental-software/ADMS-Roads-model.html*

Chen, X. & Ye, J. (2019), 'When the wind blows: spatial spillover effects of urban air pollution in china', *Journal of Environmental Planning and Management* **62**(8), 1359–1376.

Comert, G., Darko, S., Huynh, N., Elijah, B. & Eloise, Q. (2020), 'Evaluating the impact of traffic volume on air quality in south carolina', *International Journal of Transportation Science and Technology* **9**(1), 29–41.

Fayaz, S. A., Zaman, M., Kaul, S. & Butt, M. A. (2022), 'Is deep learning on tabular data enough? an assessment', *International Journal of Advanced Computer Science and Applications* **13**(4).

Fong, I. H., Li, T., Fong, S., Wong, R. K. & Tallon-Ballesteros, A. J. (2020), 'Predicting concentration levels of air pollutants by transfer learning and recurrent neural network', *Knowledge-Based Systems* **192**, 105622.

Forehead, H. & Huynh, N. (2018), 'Review of modelling air pollution from traffic at street-level-the state of the science', *Environmental Pollution* **241**, 775–786.

Grubb, D. & Symons, J. (1987), 'Bias in regressions with a lagged dependent variable', *Econometric Theory* **3**(3), 371–386.

Guarino, A., Lettieri, N., Malandrino, D., Zaccagnino, R. & Capo, C. (2022), 'Adam or eve? automatic users' gender classification via gestures analysis on touch devices', *Neural Computing and Applications* **34**(21), 18473–18495.

Guo, C. & Berkhahn, F. (2016), 'Entity embeddings of categorical variables', *arXiv preprint arXiv:1604.06737* .

Hadeed, S. J., O'Rourke, M. K., Burgess, J. L., Harris, R. B. & Canales, R. A. (2020), 'Imputation methods for addressing missing data in short-term monitoring of air pollutants', *Science of The Total Environment* **730**, 139140.

Howard, J. & Gugger, S. (2020), 'Fastai: A layered api for deep learning', *Information* **11**(2), 108.

Hua, H., Jiang, S., Sheng, H., Zhang, Y., Liu, X., Zhang, L., Yuan, Z. & Chen, T. (2019), 'A high spatial-temporal resolution emission inventory of multi-type air pollutants for wuxi city', *Journal of Cleaner Production* **229**, 278–288.

Jerrett, M. (2015), 'The death toll from air-pollution sources', *Nature* **525**(7569), 330–331.

Jida, S. N., Hetet, J.-F., Chesse, P. & Guadie, A. (2021), 'Roadside vehicle particulate matter concentration estimation using artificial neural network model in addis ababa, ethiopia', *journal of environmental sciences* **101**, 428–439.

Kocev, D., Džeroski, S., White, M. D., Newell, G. R. & Griffioen, P. (2009), 'Using single-and multi-target regression trees and ensembles to model a compound index of vegetation condition', *Ecological Modelling* **220**(8), 1159–1168.

Korneva, E. & Blockeel, H. (2020), Towards better evaluation of multi-target regression models, *in* 'Joint European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 353–362.

Li, Y., Wu, F.-X. & Ngom, A. (2018), 'A review on machine learning principles for multi-view biological data integration', *Briefings in bioinformatics* **19**(2), 325–340.

Li, Z., Yim, S. H.-L. & Ho, K.-F. (2020), 'High temporal resolution prediction of street-level pm2. 5 and nox concentrations using machine learning approach', *Journal of Cleaner Production* **268**, 121975.

Mabahwi, N. A. B., Leh, O. L. H. & Omar, D. (2014), 'Human health and wellbeing: Human health effect of air pollution', *Procedia-Social and Behavioral Sciences* **153**, 221–229.

Mao, W., Wang, W., Jiao, L., Zhao, S. & Liu, A. (2021), 'Modeling air quality prediction using a deep learning approach: Method optimization and evaluation', *Sustainable Cities and Society* **65**, 102567.

Masmoudi, S., Elghazel, H., Taieb, D., Yazar, O. & Kallel, A. (2020), 'A machine-learning framework for predicting multiple air pollutants' concentrations via multi-target regression and feature selection', *Science of the Total Environment* **715**, 136991.

Matz, C. J., Stieb, D. M., Egyed, M., Brion, O. & Johnson, M. (2018), 'Evaluation of daily time spent in transportation and traffic-influenced microenvironments by urban canadians', *Air Quality, Atmosphere & Health* **11**(2), 209–220.

Mengara Mengara, A. G., Park, E., Jang, J. & Yoo, Y. (2022), 'Attention-based distributed deep learning model for air quality forecasting', *Sustainability* **14**(6), 3269.

Morin, K. & Davis, J. L. (2017), 'Cross-validation: What is it and how is it used in regression?', *Communications in Statistics-Theory and Methods* **46**(11), 5238–5251.

Noble, W. S. et al. (2004), 'Support vector machine applications in computational biology', *Kernel methods in computational biology* **71**, 92.

Pandya, S., Gadekallu, T. R., Maddikunta, P. K. R. & Sharma, R. (2022), 'A study of the impacts of air pollution on the agricultural community and yield crops (indian context)', *Sustainability* **14**(20), 13098.

Peeples, L. (2020), 'News feature: How air pollution threatens brain health', *Proceedings of the National Academy of Sciences* **117**(25), 13856–13860.

Shen, Q., Wu, Y., Jiang, Y., Zeng, W., Alexis, K., Vianova, A. & Qu, H. (2020), Visual interpretation of recurrent neural network on multi-dimensional time-series forecast, *in* '2020 IEEE Pacific Visualization Symposium (PacificVis)', IEEE, pp. 61–70.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W. & Vlahavas, I. (2012), 'Multi-label classification methods for multi-target regression', *arXiv preprint arXiv:1211.6581* pp. 1159–1168.

Suleiman, A., Tight, M. & Quinn, A. (2019), 'Applying machine learning methods in managing urban concentrations of traffic-related particulate matter (pm10 and pm2. 5)', *Atmospheric Pollution Research* **10**(1), 134–144.

Sun, H., Fung, J. C., Chen, Y., Chen, W., Li, Z., Huang, Y., Lin, C., Hu, M. & Lu, X. (2021), 'Improvement of pm2. 5 and o3 forecasting by integration of 3d numerical simulation with deep learning techniques', *Sustainable Cities and Society* p. 103372.

Taylor, S. J. & Letham, B. (2018), 'Forecasting at scale', *The American Statistician* **72**(1), 37–45.

UKAIR (2018), 'Background mapping data for local authorities'.
URL: *https://uk-air.defra.gov.uk/data/laqm-background-home*

Wang, A., Xu, J., Tu, R., Saleh, M. & Hatzopoulou, M. (2020), 'Potential of machine learning for prediction of traffic related air pollution', *Transportation Research Part D: Transport and Environment* **88**, 102599.

Wilkins, A. S. (2018), 'To lag or not to lag?: Re-evaluating the use of lagged dependent variables in regression analysis', *Political Science Research and Methods* **6**(2), 393–411.

Wu, C.-l., Song, R.-f., Peng, Z.-r. et al. (2022), 'Prediction of air pollutants on roadside of the elevated roads with combination of pollutants periodicity and deep learning method', *Building and Environment* **207**, 108436.

Xie, M., Wu, L., Li, B. & Li, Z. (2020), 'A novel hybrid multivariate nonlinear grey model for forecasting the traffic-related emissions', *Applied Mathematical Modelling* **77**, 1242–1254.

Xu, J., Dong, Y. & Yan, M. (2020), 'A model for estimating passenger-car carbon emissions that accounts for uphill, downhill and flat roads', *Sustainability* **12**(5), 2028.

Yu, Y., Si, X., Hu, C. & Zhang, J. (2019), 'A review of recurrent neural networks: Lstm cells and network architectures', *Neural computation* **31**(7), 1235–1270.

Zeng, Y., Cao, Y., Qiao, X., Seyler, B. C. & Tang, Y. (2019), 'Air pollution reduction in china: Recent success but great challenge for the future', *Science of the Total Environment* **663**, 329–337.

Zhai, Z., Tu, R., Xu, J., Wang, A. & Hatzopoulou, M. (2020), 'Capturing the variability in instantaneous vehicle emissions based on field test data', *Atmosphere* **11**(7), 765.

Revised manuscript (Clean Version)

Deep Learning-based Multi-Target Regression For Traffic-Related Air Pollution Forecasting

Taofeek Dolapo Akinosho* TAOFEEK.AKINOSHO@UWE.AC.UK

*Corresponding author

Big Data Enterprise and Artificial Intelligence Lab (Big-DEAL)
Bristol Business School
University of the West of England, Frenchay Campus,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Muhammad Bilal MUHAMMAD.BILAL@UWE.AC.UK

Big Data Enterprise and Artificial Intelligence Lab (Big-DEAL)
Bristol Business School
University of the West of England, Frenchay Campus,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Enda Thomas Hayes ENDA.HAYES@UWE.AC.UK

Air Quality Management Resource Centre,
University of the West of England, Frenchay Campus,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Anuoluwapo Ajayi ANUOLUWAPO.AJAYI@UWE.AC.UK

Big Data Enterprise and Artificial Intelligence Lab (Big-DEAL)
Bristol Business School
University of the West of England
Bristol, United Kingdom

Ashraf Ahmed ASHRAF.AHMED@BRUNEL.AC.UK

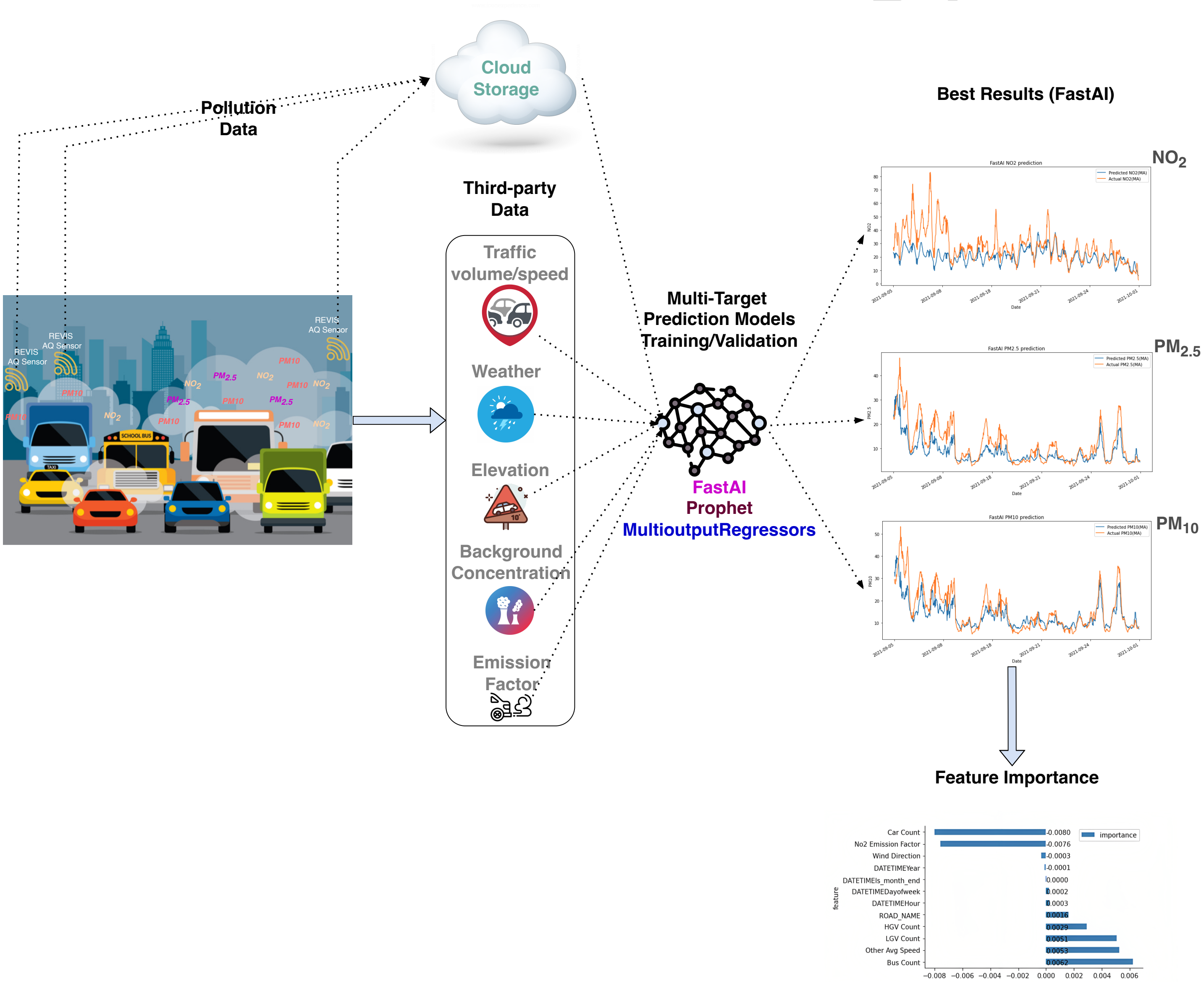
Department of Civil and Environmental Engineering
Brunel University London
Kingston Lane
Uxbridge, United Kingdom

Zaheer Khan ZAHEER2.KHAN@UWE.AC.UK

Department of Computer Science and Creative Technologies,
University of the West of England , Frenchay Campus,
Coldharbour Lane, Bristol BS16 1QY, United Kingdom

Research Highlights

- Challenges of traffic-related air pollution forecasting methods are highlighted.
- Additional highway data integrated for improved traffic pollution forecasting.
- FastAI's tabular model performs best for NO2, PM10 and PM2.5 prediction.
- Model's performance improved through Lagged variables and categorical embeddings.
- Traffic and weather data contributed the most to model's TRAP forecasting.


Cloud Storage
Pollution Data
Third-party Data
REVIS AQ Sensor
REVIS AQ Sensor
REVIS AQ Sensor
Traffic volume/speed
Weather
Elevation
Background Concentration
Emission Factor
Multi-Target Prediction Models Training/Validation
FastAI
Prophet
MultioutputRegressors
Best Results (FastAI)
NO2
PM2.5
PM10
Feature Importance
Car Count
No2 Emission Factor
Wind Direction
DATETIMEYear
DATETIMEIs_month_end
DATETIMEDayofweek
DATETIMEHour
ROAD_NAME
HGV Count
LGV Count
Other Avg Speed
Bus Count
importance
feature

CRediT Author Statement

Taofeek Dolapo Akinosho: Conceptualization, Methodology, Validation, Writing -original draft, Writing - review & editing, Investigation, Data Curation, Formal analysis
Muhammad Bilal: Conceptualization, Methodology, Validation, Supervision, Project administration
Enda Thomas Hayes: Methodology, Validation, Supervision
Anuoluwapo Ajayi: Methodology, Validation, Supervision
Ashraf Ahmed: Visualization, Writing - Review & Editing
Zaheer Khan: Visualization, Writing - Review & Editing

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: