Contents

Chapter 1

Chemical model diagnostics using graph theory and metrics.

"The complexities of cause and effect defy analysis."

- Douglas Adams, *Dirk Gently's Holistic Detective Agency*

1.1 Case study Example

In this section the centrality metrics discussed in **??** are applied to a range of senarios. These range from polluted urban environments such as London [REF] and Beijing [REF], to marine and terrestial forrest- Cape Verde REF and Borneo REF. We determine the main drivers for the chemistry and compare the species which are important across each simulation.

1.1.1 Establishing Initial Conditions from observational data

Within experimental data assimulation it is not uncommon to face problems which resut in unreliable or missing data. These can range from anything as little as measuring below the instrument sensitivity to powercuts and equiptment damage/theft from the local wildlife. This can result in problems when analysing the results and combining them to create a simulation of the chemistry for that environment.

To overcome this, traditionally a combination of data filtration, smoothing and interpolation is required. Although it is possible to fit a diurnal profile, through itereative methods of comparison, and cubic splines, a much simpler way would be to use an Multi Layer Perceptron Regressor model (MLPRegressor) as provided by sklearn, [**?**]. This is described below.

1.1.1.1 The origin of Artificial Neural Networks

The concept of a neural network originated witin the field of neuroscience. In in bilogical neurons, signals are sent through the use of electrical impulses using their synapses. When a sufficient number of signals are recieved within a short timeframe, a neurone will respond, often firing a range of its own signals. Using this as a foundation,McCulloch and Pitts [1943] presented a computational model of the biological neuron - the artifical neuron. This has a series of binary inputs, and produces a single binary output. This idea was later imporoved with the invention of the perceptron - a linear classifier which classifies categories by separating them with a straight line. Invented by Rosenblatt [1958], this was popularised as a device representative of a modern day shallow neural network - [John Hay, 1960], Figure 1.1. Unlike the artificial neuron however, the perceptron is able to take non-binary (numerical) inputs of an associated weight which allows for the computation of simple linear binary classification. Much like Logistic regression, the preceptron produces a positive or negative classification based on a certain threshold[1].

[1] It is worth noting that while a Logistic Regression classifier can output a class probability, the use of a hard threshold means that this is not done within the preceptron algorithm [Géron, 2017]

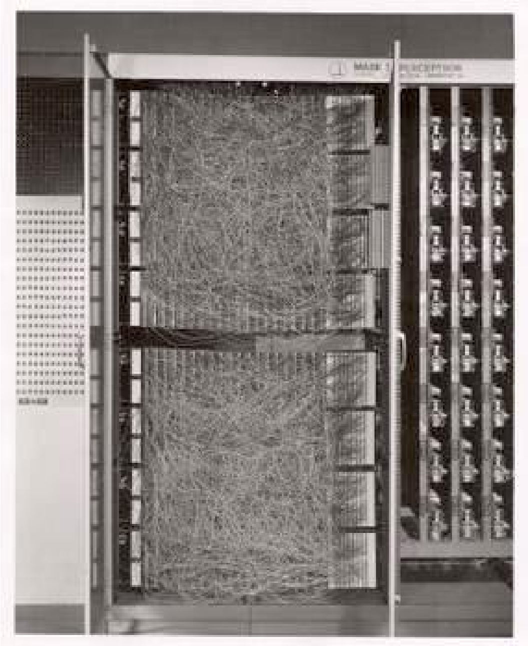


Figure 1.1: **The Mark 1 perceptron** Both software and hardware are different manifestations of a flow chart. The perceptron hardware accomplished what is now done using software. Source: Cornell [2020]

1.1.1.2 The Multi Layer Perceptron

Limitations of ther perceptron include the classifiction of complex patterns such as the XOR problem (where a category appears between two other categories e.g. 1|0|1 - this cannot be classified by a single linear split). In taking inspiration from nature, Figure 1.2, it is possible to overcome this with the use of multiple layers. This creates an a deep (> 2 two hidden (non-input) layers of perceptrons[2]) artificial neural network (ANN)

The multi layer perceptron (MLP) model now represents a simple feed-forwards network, much like a decision tree. However unlike a decision tree, the MLP ANN is able to describe the probability a branch is taken using non-linear activation (threshold) functions. These are discussed in detail as part of SECTION NEXT CHAPTER. The weighting thresholds for each neuron are then calculated by backwards propagation of results through the network until a suitably good result is produced.

> ***Example analogy:*** *Back propagation can be likened to the iterative calibration of scientific instrumentation. In the field of atmopsheric chemistry laser induced flourecence is used to calculate species concentrations and reaction rates within the troposphere, [Dillon et al., 2006; Bloss et al., 2004]. Here the frequency of a laser can be adjusted in contrast with a known target (e.g. an amount of SO_2) to produce a responce curve showing where the maximum resonance occurs.*

[2] These are sometimes refered as Linear Threshold Units.

Similarly a neural network can be 'trained' (calibrated). This is done through the use of a 'training dataset' - a set of input-output pairings which represent a random selectaion of 2/3rds of the total dataset. Next the neurons within each layer (similar to the potentiometer dials on an instrument) are adjusted in sequence through the layers to match the known result (a standard of known concentration) to the input values provided. This process is repeated until for a number of iterations, or until a sufficently 'good' prediction is attained for the entire training dataset (early termination). The power of ANNs comes from the ability to adjust neuron thresholds whilst moving both forwards and backwards through the network (Note: predictions of a MLP are still only passed forwards). Finally model perfomance is evaluated against the remaining 1/3rd of the total dataset.

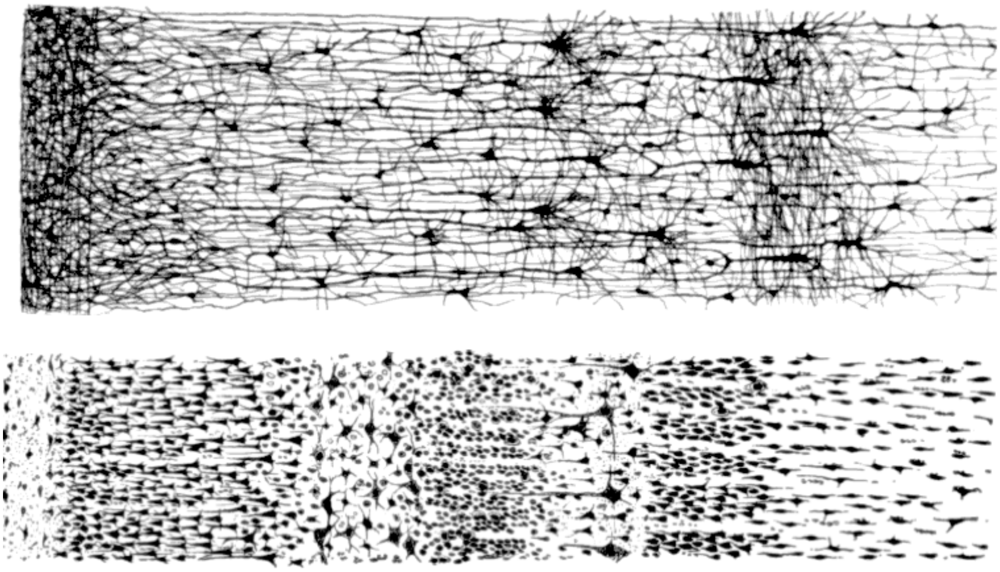


Figure 1.2: **The Human Cortex - A biological neural network.**. A vertical cross section of the human cortex between an adult (top) and 1.5 month old infant (bottom) showing a layer like structure with a change in depth (left to right). Source: Cajal [2020]

1.1.1.3 Application of the MLPRegressor to observational data

In the application of any type of machine aided algorithms it is important to evaluate the results provided. In this section the results of 12 years of data collected as part of the [CAPE VERDE CAMPAIGN] are shown (these contain measurements spanning the entirety of 12 years, which produce the clearest tests for the algorithm). A MLPRegressor of 10 hidden layers, and a hyperbolic tan (tanh) activation function is used. Additionally the limited-memory Broyden–Fletcher–Goldfarb–Shanno (l-BFGS) solver (a quasi-newton method which minimises the inverse of the Hessian matrix[3] to steer

[3] The hessian is square matrix of second-order partial derivatives of a scalar-valued function/field describing the local curvature of a function (of many variables).

through space and obtain a solution) and an adaptive learning rate[4] is used.

Input of the regressor is in the from of a month and a hour, to represent each measurement. This allows it to find not only daily trends, but also seasonal trends within the data. Once trained the regressor is then used to predict a diurnal profile for each month based on the observational data provided. For simplicity $\log_{10}$ values of the concentrations obtained have been used. To validate the results, the predicted MLPRegressor line is compared to a transparent scatterplot for all the results. In addition to this a boxplot showing the IQR, median and mean (green line) plotted alongside to evaluate the predictor output.

In providing the MLPRegressor with both month and hour inputs, the data is not only fitted hourly (a diurnal avarage), but also across the seasonal/monthly cycles. This accounts for the variation between years and datasets. Since $\log_{10}$ values of the concentrations are used, species such as ozone (Figure 1.3) which for the Cape Verde dataset (clean air) do not change more than one order of magnitude, the effects of neigbouring months, which shift the diurnal away from the mean (the green line on the boxplot), can be seen. However since this is overall a small change, and the diurnals lie within the inter quartile range, they still provide an adequate approximation. NO (Figure 1.4) on the other hand has a concentration change of several orders of magnitude. Here a distinct daytime peak is seen and is centred around a sesonally consistent mean value of the data. Here the multi-magnitude change in concentration also provides an effective silhouette of the data to which we may compare the fitted line. Finally the plots of NO_2 and iso-Pentane (Figure 1.5-1.6) vary both in diurnal magnitude and seasonally. Within these plots, changes in the data in the january and december months produce deceptively misleading results. Here although the diurnals are not symmetrix, they fit well within the median,mean and interquartile range values, as well as the general data silhouette behind them. This siggests that it is a property of the data that we are fitting, and not that the regressor is producing incorrect results. It is however noted that for a more accurate seasonal prediction, periodic boundary conditions should be employed in the training dataset, where an additional two months are added before January and after December.

Since I shall be using only a noon estimate from the summer region, this does not affect any of the results.

[4] Each time the model improvement fails to decrease the learning loss, the learning rate is reduced by 1/5. This means smaller jumps are made towards the curve peak.

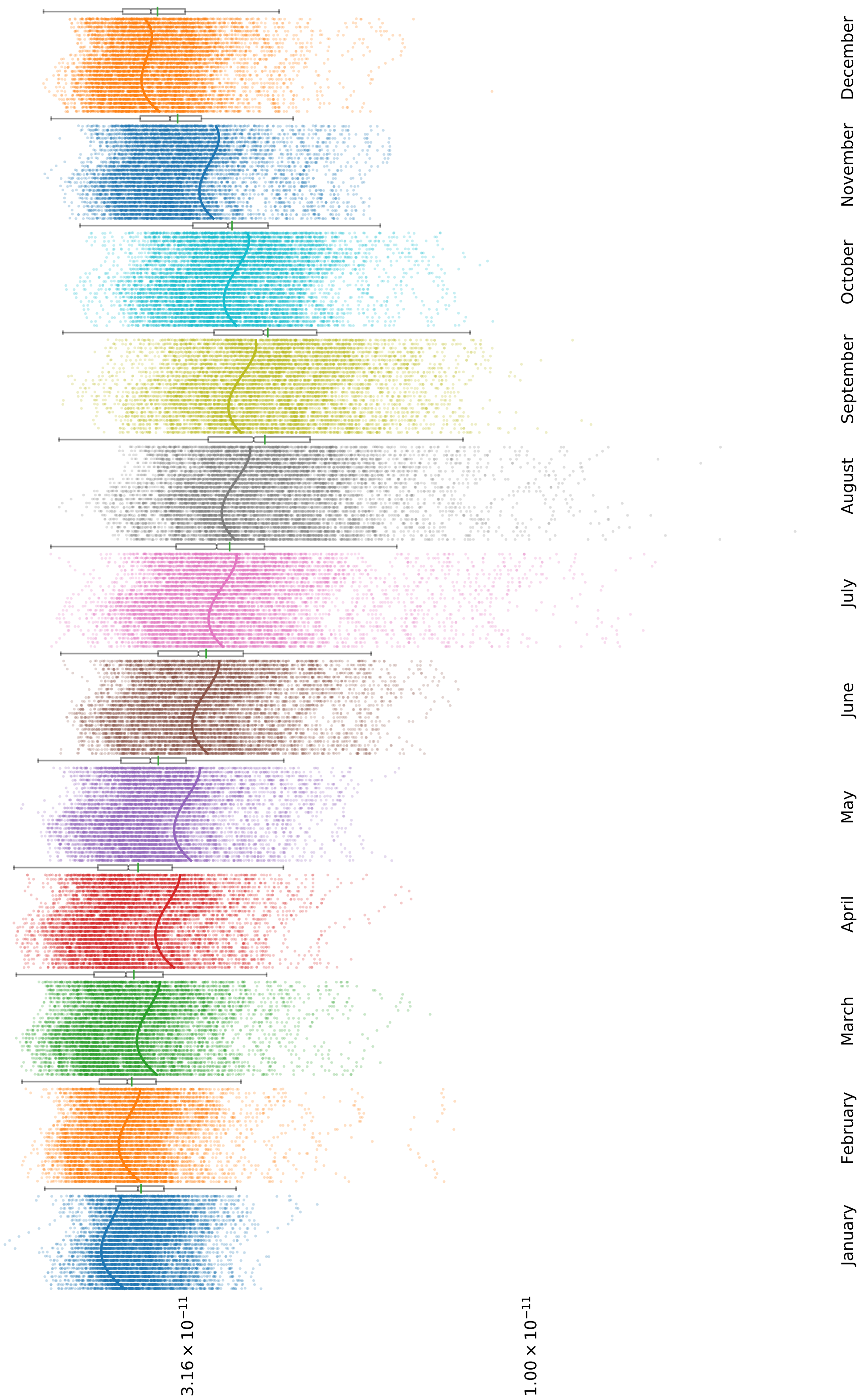


Figure 1.3: **Cape Verde MLP predicted and observational data of Ozone.** Each segment represents data from a different month. Within each month segment exists 24 hour segments to create a diurnal. Observational concentrations are plotted in the form of a translucent scatterplot and summarised using the boxplot on the right of the month segment. MLP predicted results are shown using the solid lines. Concentration in mixing ratio.

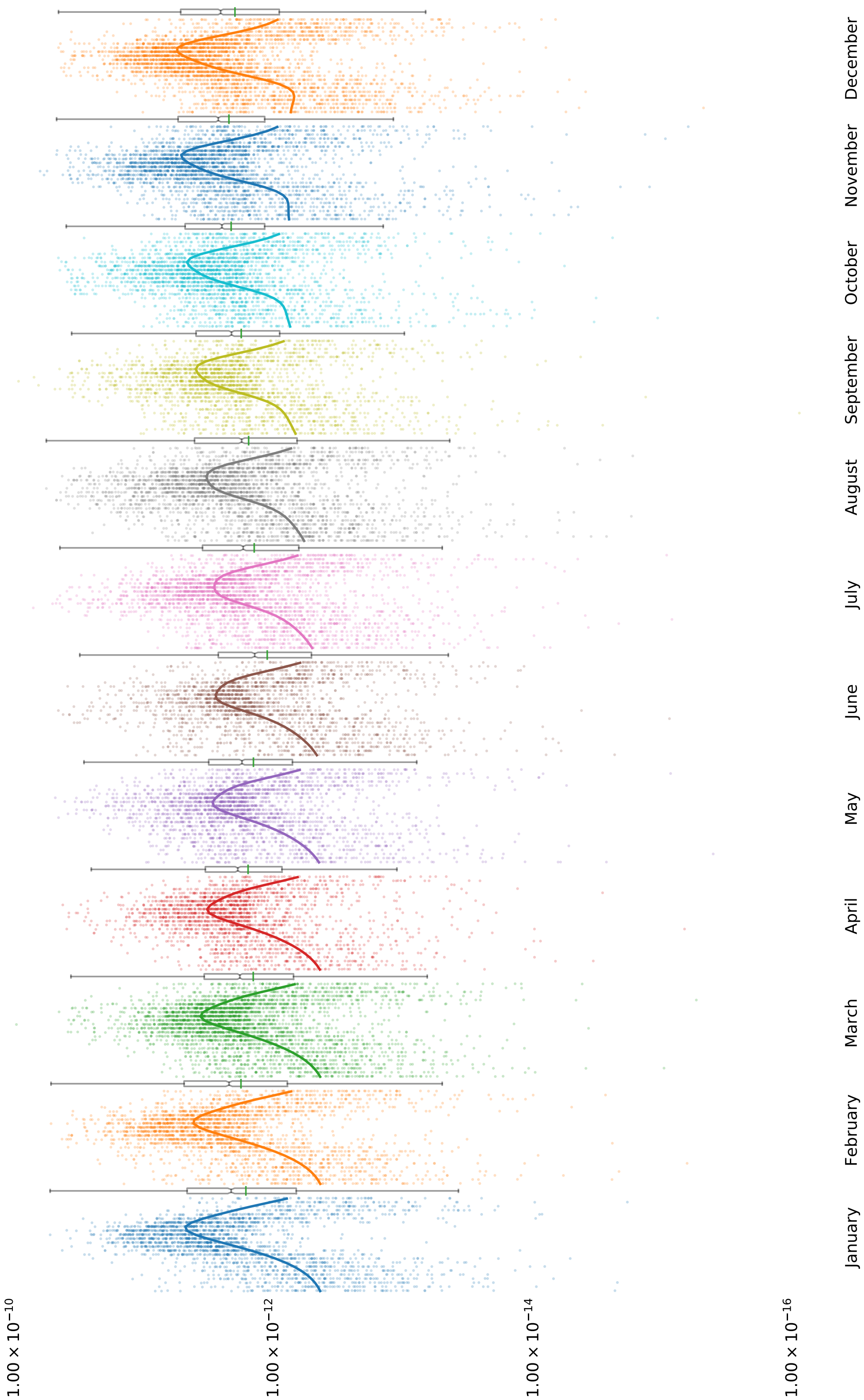


Figure 1.4: **Cape Verde MLP predicted and observational data of NO.** Each segment represents data from a different month. Within each month segment exists 24 hour segments to create a diurnal. Observational concentrations are plotted in the form of a translucent scatterplot and summarised using the boxplot on the right of the month segment. MLP predicted results are shown using the solid lines. Concentration in mixing ratio.

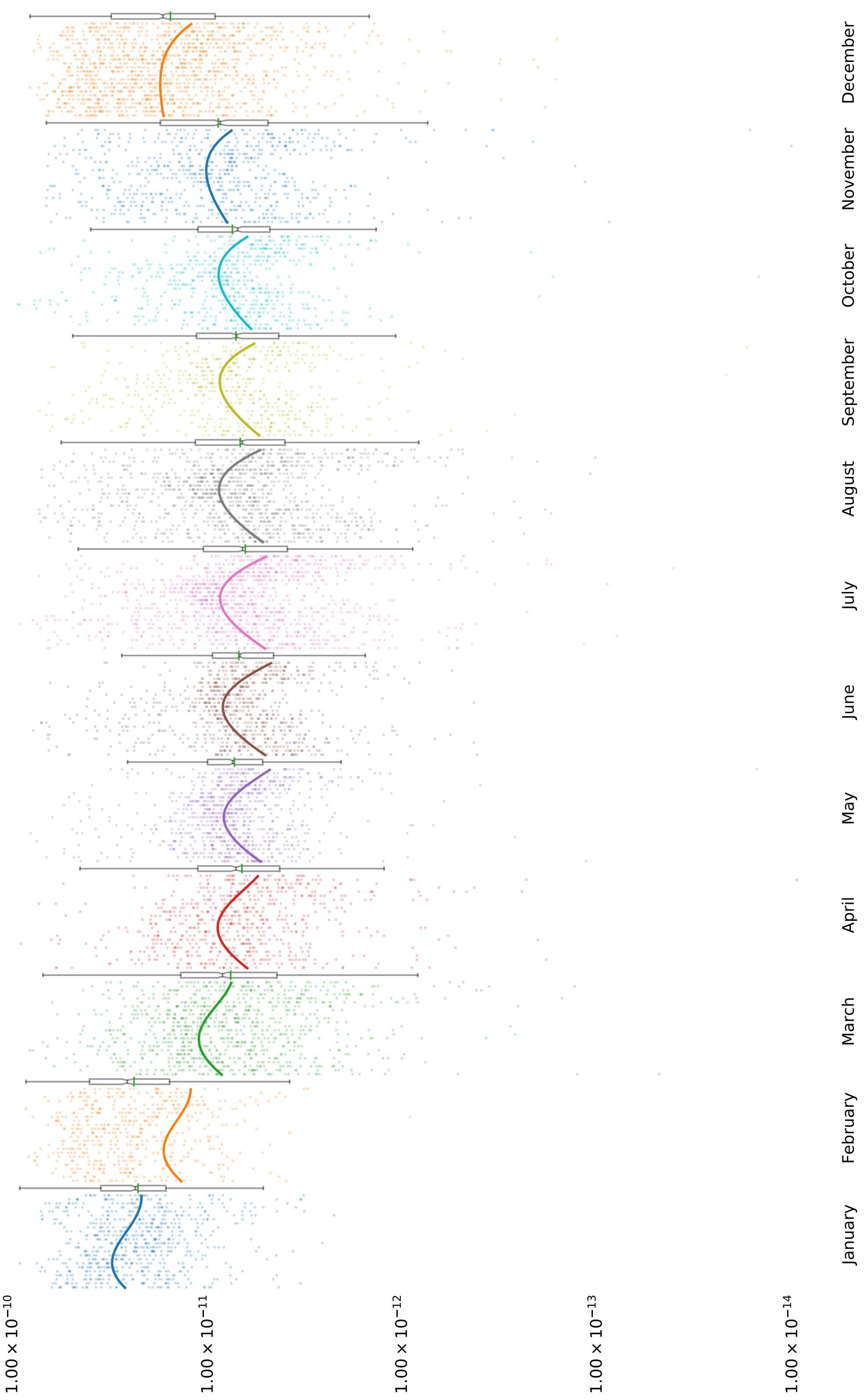


Figure 1.5: **Cape Verde MLP predicted and observational data of NO_2.** Each segment represents data from a different month. Within each month segment exists 24 hour segments to create a diurnal. Observational concentrations are plotted in the form of a translucent scatterplot and summarised using the boxplot on the right of the month segment. MLP predicted results are shown using the solid lines. Concentration in mixing ratio.

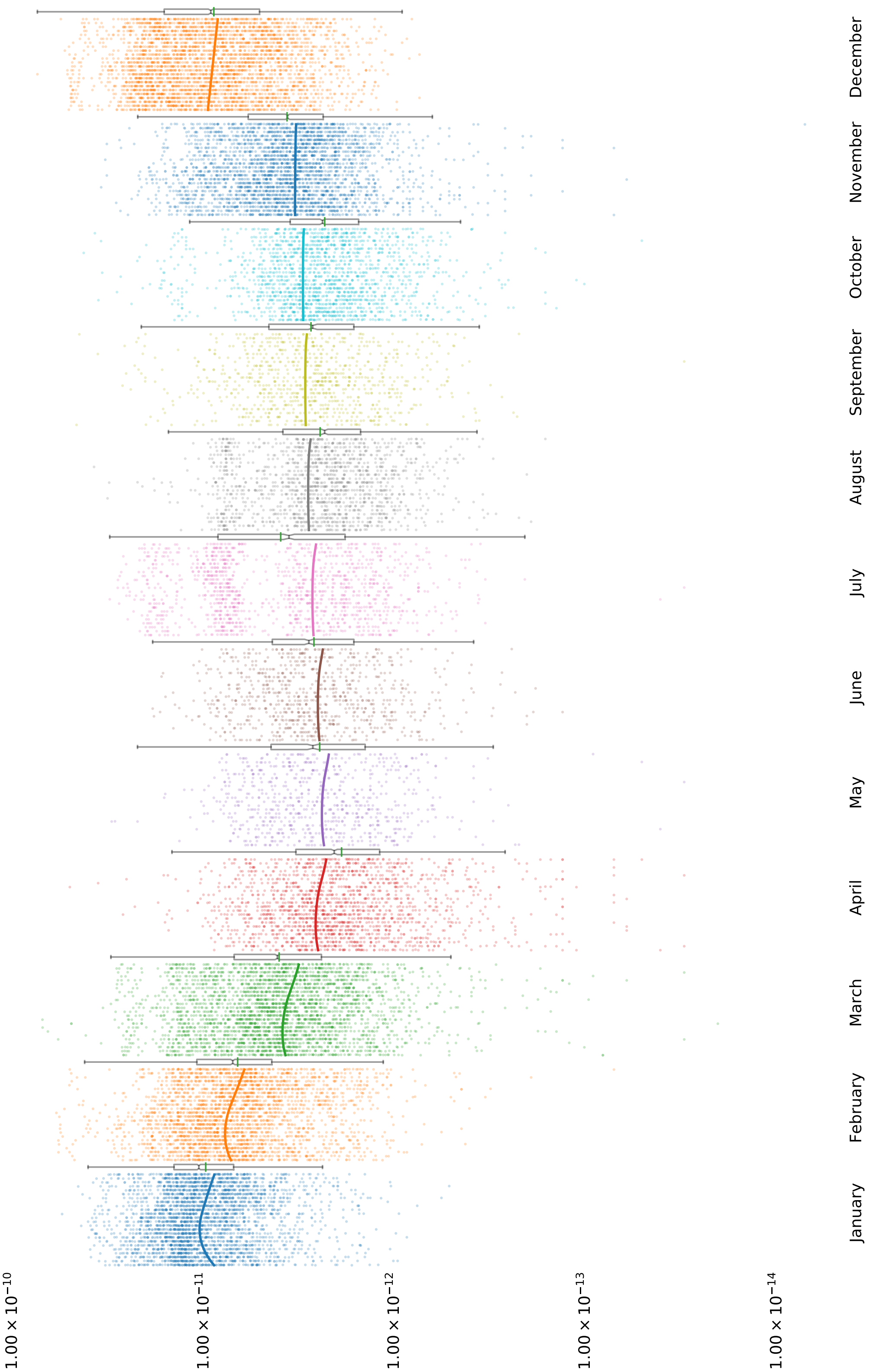


Figure 1.6: **Cape Verde MLP predicted and observational data of iso-Pentane.** Each segment represents data from a different month. Within each month segment exists 24 hour segments to create a diurnal. Observational concentrations are plotted in the form of a translucent scatterplot and summarised using the boxplot on the right of the month segment. MLP predicted results are shown using the solid lines. Concentration in mixing ratio.

1.1.1.4 Model Initialisation Procedure

To generate the initial concentrations used for a simulation an representative concentration for all species observed in the relevant campaigns are generated using the MLP regressor. Here concentrations are taken at noon (12:00) at the month closest to June in which reults were taken. This produces a montly error of $\pm 2 months$. As both nitrogen oxide and dioxide are supplied the total NO_x for each simulation are *not* constrained. The initial conditions are shown in Table **??**.

In this section we are not interested in the exact concentration modelling for specific times or senarios. Instead we seek to generate representative of the processed chemistry under a range of conditions.

In general observational measurements are not able to detect all the species presented within the MCM. This means that in order to be able to compare model senarios, the chemistry must first be spun up. In propatating the chemistry forwards in time, primary emitted and measued species are broken up forming the intermediate species which exist within a mechanism. In order to reach steady state, the model is initated at noon and the observational concentrations are rest every 24 hours. For each diurnal the fractional difference between the concentrations at each day are compared. If the difference between these is less than 0.001, the model is left to run unconstrained for 5 days (right of the dashed line in Figure **??**-**??**). Model results are then taken after 3 days of unconstrained runs. The reason for this is that the total RO_2 concentration takes longer to stabilise in the polluted environments (London and Beijing). This falls into a periodic cycle beginning noon on the third day, and can provide a representation of the processed chemistry within each environment.

1.1.1.5 Extracting the required results

Model diagnostics such as concentration and the net flux passing through a species may be extracted directly from the DSMACC box model. These provide the baseline comparison and can be directly compared to the graph metrics. Species concentration tells us the abundance of different species, and the net-flux tells us how fast this is changing with respect to time.

As some species may have a fast inwards and outwards flux (low net-flux), the aboslute flux is also included. Finally the sensetivity of each species with respect to other species is also extracted (the jacobian matrix). This serves to not only generate the graph used to represent the chemistry, (**??**), but also to identify the overall influence a species has on others in the network. This can be calculated by taking the net sum of the influence a species has on every other from the Jacobian. This is analogous to calculating the outdegree of a node in the jacobian network.

Species	Beijing(APHH)	Borneo(OP3)	London(ClearFlo)	CapeVerde
LAT	39.9	0.96	51.0	16.5
LON	116.3	114.5	0.00	23.4
O_3	6.883e-08	8.939e-09	3.819e-08	2.629e-11
NO	1.660e-09	2.668e-14	2.350e-09	2.358e-12
NO_2	1.226e-08	1.081e-13	7.445e-09	8.447e-12
HCHO	4.472e-09		1.119e-08	
C_2H_6	3.163e-09	7.315e-10	2.133e-09	4.539e-10
C_2H_4	1.004e-09	1.152e-10	4.893e-10	2.481e-11
C_3H_8	3.019e-09	1.924e-10	1.128e-09	1.728e-11
C_3H_6	1.335e-10	1.333e-11	1.784e-10	9.343e-12
IC_4H_{10}	6.412e-10	8.742e-11	5.142e-10	2.486e-12
NC_4H_{10}	1.593e-09	5.698e-11	1.058e-09	4.481e-12
C_2H_2	1.058e-09	1.825e-10	3.018e-10	1.848e-11
TBUT2ENE	4.198e-11		1.815e-11	
CBUT2ENE	4.454e-11		1.305e-11	
IC_5H_{12}	1.047e-09	2.883e-11	7.424e-10	3.470e-12
NC_5H_{12}	4.650e-10	2.090e-11	2.792e-10	2.513e-12
TPENT2ENE	3.939e-11			
CPENT2ENE	3.982e-11			
NC_6H_{14}	2.057e-10	6.437e-12	6.357e-11	
C_5H_8	7.134e-10	1.957e-09	1.640e-10	
NC_7H_{16}	7.905e-11		5.222e-11	
BENZENE	4.045e-10		1.137e-10	7.682e-12
NC_8H_{18}	3.091e-11		1.442e-11	
TOLUENE	6.767e-10		3.205e-10	3.121e-12
EBENZ	3.115e-10		6.017e-11	
OXYL	1.677e-10		5.049e-11	
CH_3CHO	4.783e-10		4.095e-09	
C_2H_5OH	4.655e-09		3.125e-09	
CH_3COCH_3	3.328e-09		2.924e-09	
NC_9H_{20}	1.336e-11		7.922e-11	
$NC_{10}H_{22}$	1.062e-12		1.602e-10	
α-PINENE[5]	7.341e-11	15e-11	1.105e-10	
LIMONENE	5.836e-11	1.351e-10	3.566e-11	
PXYL$^+$MXYL[6]	4.943e-10			
IPBENZ	4.567e-10			
PBENZ	3.996e-10			
HONO	6.479e-10		4.109e-10	
MACR		6.948e-11	1.862e-11	
$PENT_1ENE$			2.383e-11	
MVK			2.091e-11	
NPROPOL			2.883e-10	
NBUTOL			4.535e-10	
STYRENE			2.241e-11	
MEK			5.494e-11	
C_3H_7CHO			9.534e-12	
C_4H_9CHO			1.865e-11	
$C_5H_{11}CHO$			1.201e-11	
CYHEXONE			9.790e-12	
BENZAL			1.510e-11	
PAN			1.791e-10	

Table 1.1: The initial conditions created from the MLPRegressor prediction of observational data. Although not specified the concentration for methane is set by the model at 1770ppb.

[6] This is written as ?-pinene in the merged CEDA dataset for the Borneo OP3 campaign. This is due to character conversion errors.

[6] The concentration for these is split evenly between both species

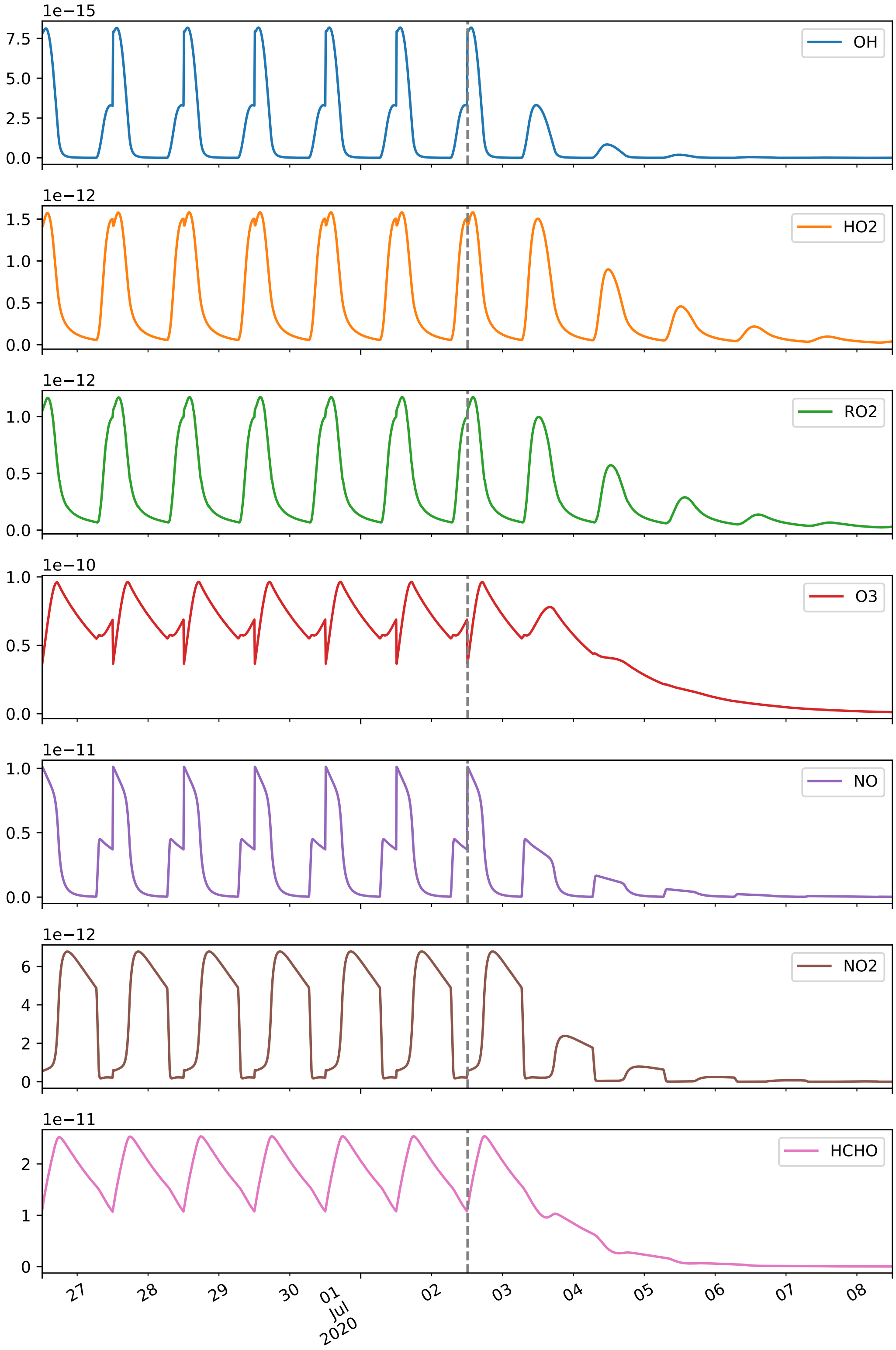


Figure 1.7: **The concentration profile for CapeVerde.**This shows a the change in concentration over time for HO_x,NO_x,Ozone and RO_2 species for a simulation run generated by the mlpregressor. Left of the dashed line shows the last 6 days of spinup, where the intial concentrations are reset at noon each day until the species fractional difference is less than 0.001 .

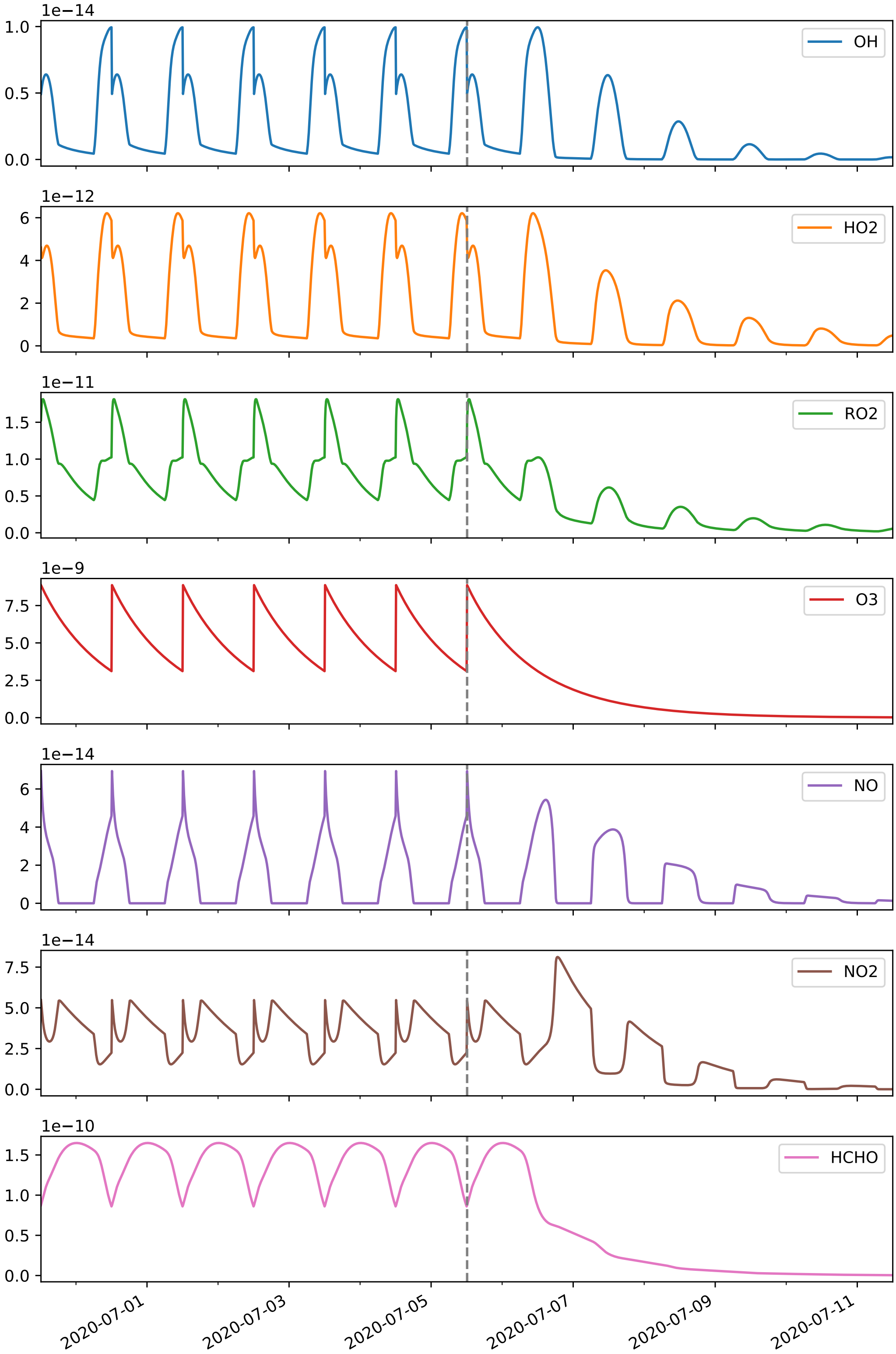


Figure 1.8: **The concentration profile for Borneo.**This shows a the change in concentration over time for HO_x,NO_x,Ozone and RO_2 species for a simulation run generated by the mlpregressor. Left of the dashed line shows the last 6 days of spinup, where the intial concentrations are reset at noon each day until the species fractional difference is less than 0.001 .

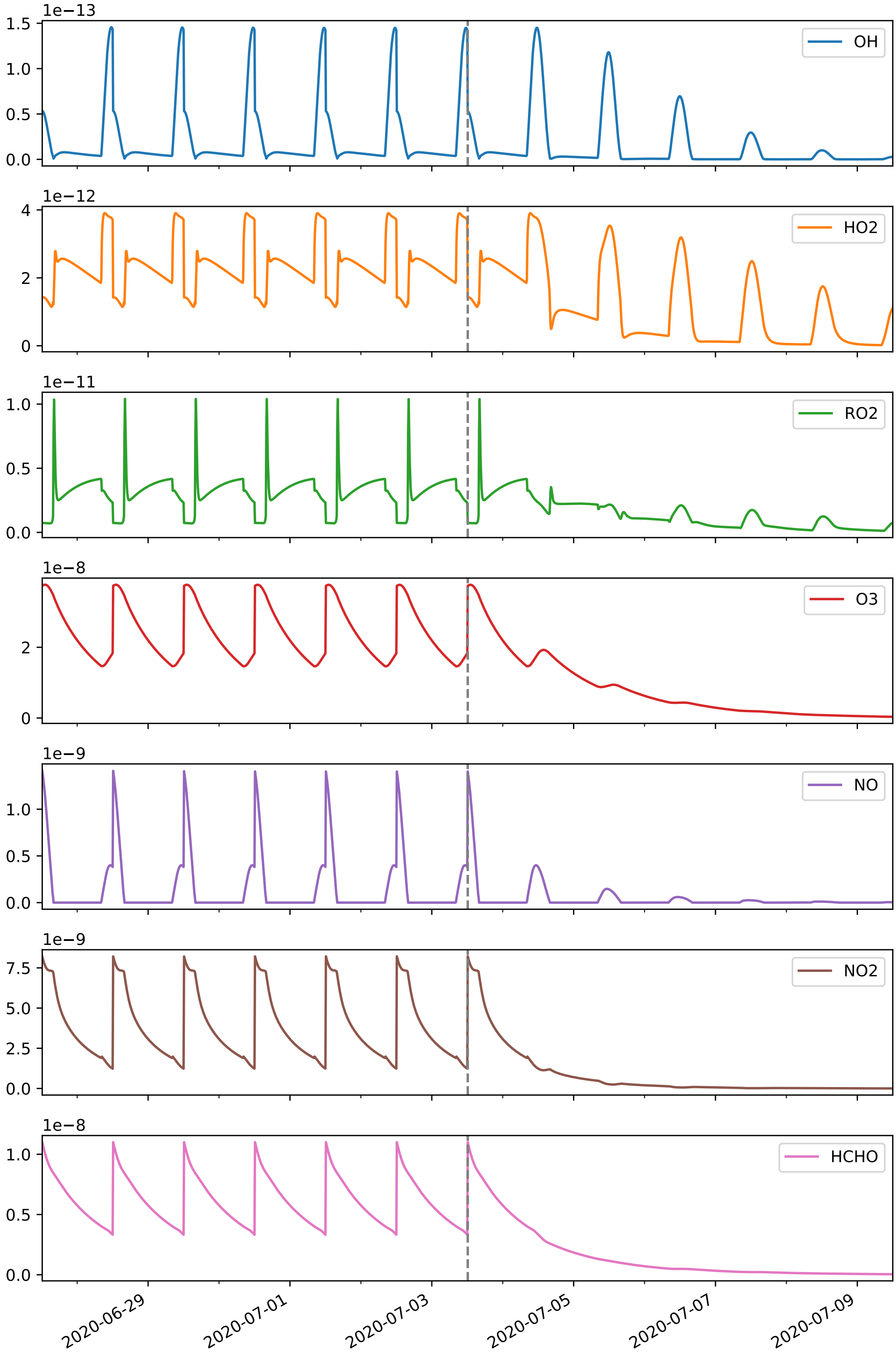


Figure 1.9: **The concentration profile for London.**This shows a the change in concentration over time for HO_x,NO_x,Ozone and RO_2 species for a simulation run generated by the mlpregressor. Left of the dashed line shows the last 6 days of spinup, where the intial concentrations are reset at noon each day until the species fractional difference is less than 0.001 .

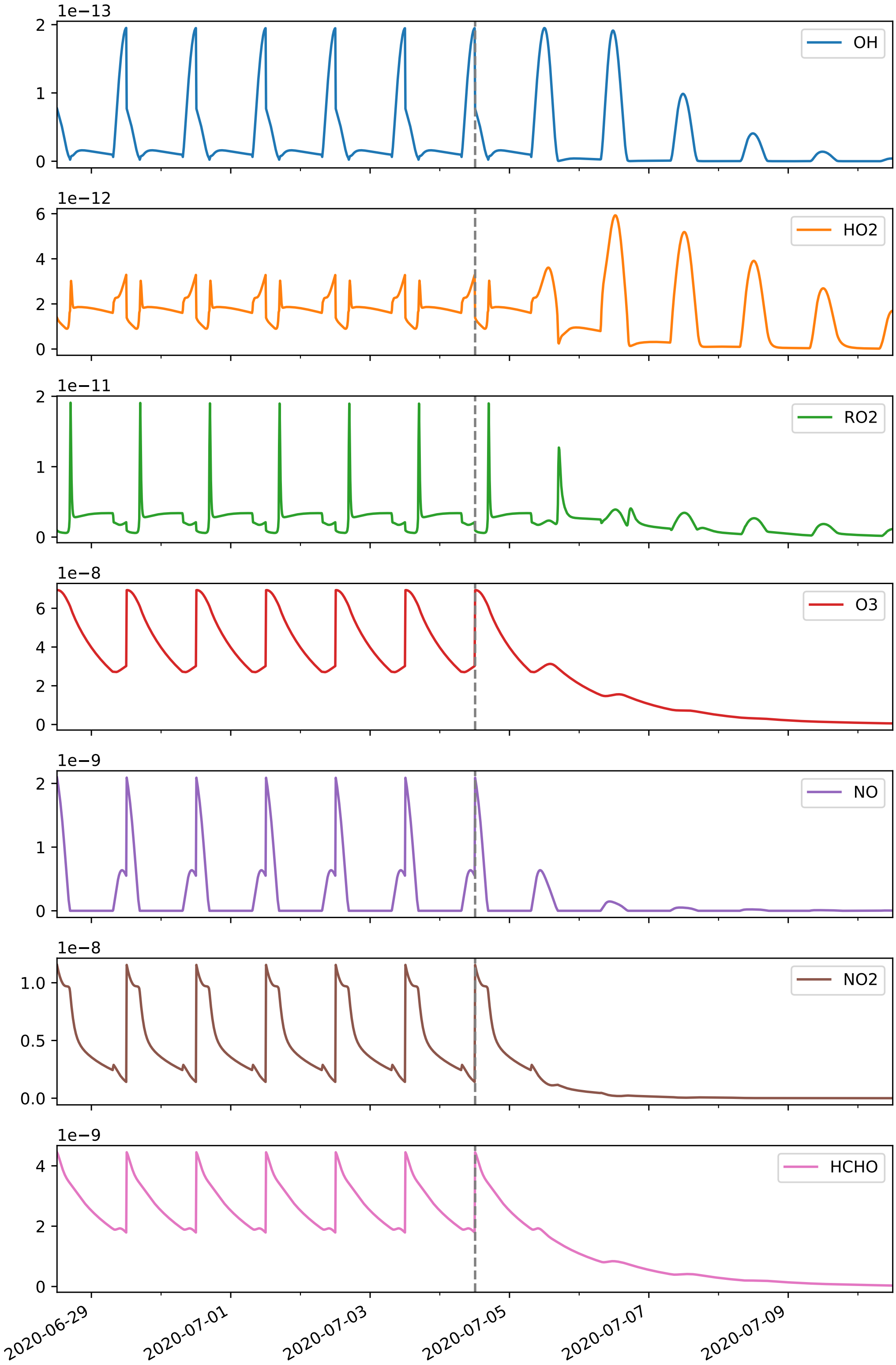


Figure 1.10: **The concentration profile for Beijing.**This shows a the change in concentration over time for HO_x,NO_x,Ozone and RO_2 species for a simulation run generated by the mlpregressor. Left of the dashed line shows the last 6 days of spinup, where the intial concentrations are reset at noon each day until the species fractional difference is less than 0.001 .

1.1.1.6 Unifying the results

Each metric provides a different range in which it ranks the importance of a node. In order to account for this all results are scaled to the range {0,1}, where 1 is the highest. Entries where the results span several orders of magnitude (e.g. concentration, flux, influence) are flattened using the $\log_{10}$ scale before being normalised.

1.1.2 Comparing Results

This subsection juxtaposes the use of traditional model diagnostic methods against a selection of graph metrics. As there are several thousand species within each simulation run, the keyword extraction algorithm Term Frequency - Inverse Document Frequency (TF-IDF), is used to identify the top most prominent species for each metric (traditiona and graph). From this the 10 highest ranking species from each category are collated into a single diagram for comparison.

1.1.2.1 What is TF-IDF

TF-IDF is a numerical statistuc used in text natural language processing and text mining. It is designed to be identify the importance of a word with regard to its context.

It provides a value for the frequency a word appears within a section, offset by the number of times it appears in other sections - It is for this reason that 83% of text recommender systems in digital libraries use TF-IDF, [Beel et al., 2016].

In [Ellis, 2019] I applied this to the chapters of frankenstein, and found the keywords extracted almost exactly replicated those from the synoptic description of the novel. Although TF-IDF is a text mining procedure, the algorithm itself is mathematical, meaning that it may be applied to our diagnostic dataset. The working of the algorithm are discussed below.

Term Frequency

The TF from the algorithm name stands for term frequency. This is an analysis of the number of times a word exists within a dataset. There are several ways in which this can be done, these are:

- **Raw Count** - The *number of times* a word exists within the document.
- **Boolean/Logistic** - *True* if the word exists, false otherwise.
- **Adjusted for Document Length** - $word\ frequency/total\ number\ of\ words$

- **Log Scaled** - $\log(1 + frequency)$

As the scaled values for each item are taken, we can liken our results to the 'Adjusted for Document length' equation and use the scaled ranking value for each group respectively.

Inverse Document Frequency

Inverse document frequency tell us how much information a word provides with respect to a certain context. Whilst a word may be used extensively throughout the corpus (i.e. term frequency) it is often that we are interested in words which are only frequent within a specific document. This is one of the reason TF-IDF is useful in the extraction of keywords from a document.

The inverse frequency of a word is usually calculated as the log of the fraction of documents N against the number of documents the word appeas in D_f, Equation 1.1.

$$IDF = \log(\frac{N}{D_f}) \tag{1.1}$$

If required, changes can be made to produce results which are show a better representation of words which are important for all documents (probabalistic, Equation 1.2) or individually (smooth, Equation 1.3). However in looking at Figure 1.11, it can be seen that the basic IDF formula mentioned has a limit of zero the greater the document frequency (D_f), which makes it easy to normalise against - i.e. divide by 2 as this is the the value tended to if the document freqnecy tends to 0.

$$IDF = \log(\frac{N - D_f}{D_f}) \tag{1.2}$$

$$IDF = \log(\frac{N}{1 + D_f}) + 1 \tag{1.3}$$

To complete the TF-IDF equation, the term frequency and inverse document frequency terms are multiplied together.

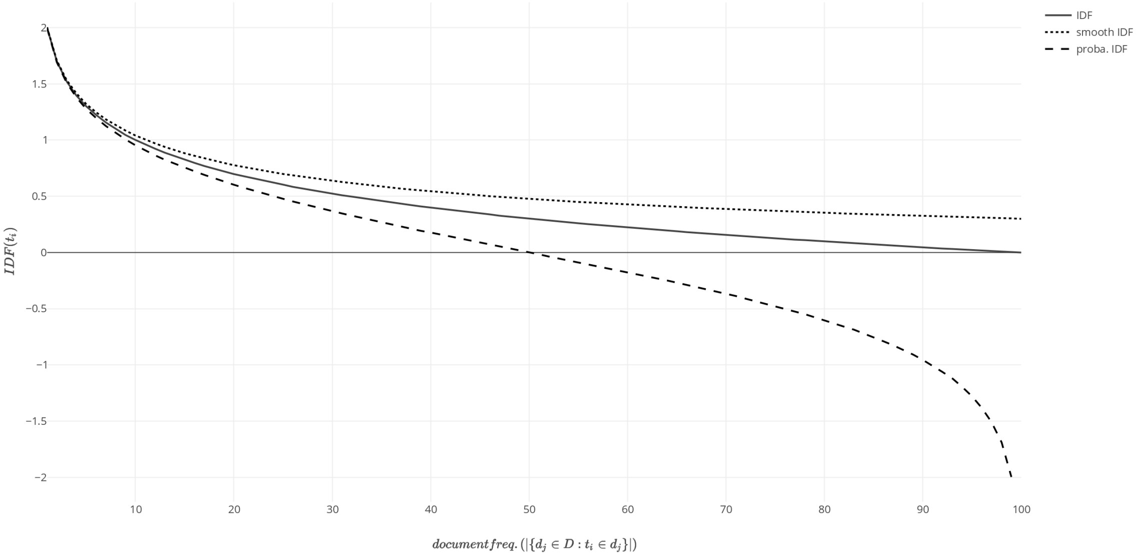


Figure 1.11: **The different IDF outputs.** A plot showing Inverse Document Frequency profiles against Document Frequency. This shows that the probabalistic IDF hilights words that are more important across all items, whilst the smooth IDF shows files which are more important individually. The general IDF (which is used) produces a result starting at 2 and tending to zero. This provides the best response and can easily be scaled between the range of [0,1] by dividing the output by 2. Source: [Mquantin, 2020]

Applying TF-IDF to chemical metrics

As the point of interest is in identifying species which are only important to a specific metric, and not across all, it is possible to adapt the TF-IDF equation. As the Term Frequency correspoinds to the number of times a value appears within the body of a document, the scaled metric value may be used. Next this is devided by the log of the Inverse Document Frequency. The Documnent Frequency can be given as the sum of values across all metrics. This makes the TF-IDF equation:

$$TF.IDF = metric\_value\ .\ \log(\frac{N_o\ documents}{\Sigma_\forall\ metric\_values}) \tag{1.4}$$

1.1.3 Metric Comparison

The aim of this section is to compare the efficiency of graph metrics against a list of traditional methods. To do this the use of a bivariate colourmap (Figure 1.12) is used. Each figure consists of a red hued image/heatmap representing the scaled values {0,1}:{white,red} for each of the individual columns. As each simulation contains thousands of species, only the top 10 species from each column/category are selected. These are then sorted by the average sum of their closeness, betweenness and page-rank values (blue column). Superimposed on this reds-only heatmap is a blue heatmap representing the average sum of the three metrics for comparison. Such a method allows for the comparison of individual values against a an approximation of species importance, by the sum of graph metrics. This allows us to partition the data into different categories.

- **Purple** - This value is high in both the individual category and the metric sum.
- **Red** - This value is high for the individual category but not the metric sum.
- **Blue** - This value is high for the metric sum but not the individual category.
- **White** - This value is low for all categories.

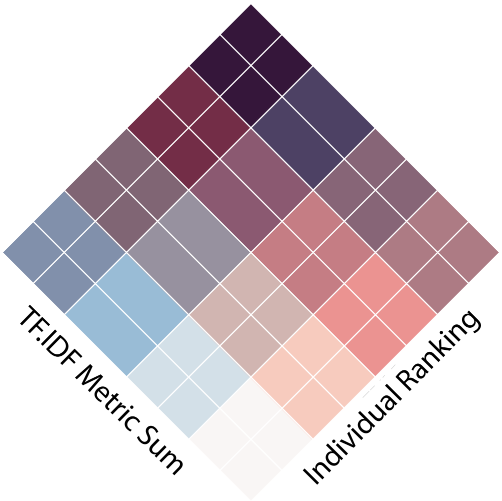


Figure 1.12: **The bivariate colourplot key.**

1.1.3.1 Individual Categories

Individual categories are split between traditional metrics and graph centrality metrics. To represent the importance of a species the following values may be extracted through the use of a simple box model:

- **Concentration** - This describes the abundance of a species within the atmosphere.
- **Net Flux** - This describes the rate of net (absolute) change of concentration over time for a species.
- **Absolute Flux** - Some species may have a large flux going through them (production and loss), resulting in a small net flux. This sums the production and loss fluxes.

- **Influence** - Influence is the total magnitude of an effect that changing a species concentration by 1% would have on other species within the network. Since the graph is generated using the Jacobian matrix, an alternative method for calculating this can be by calculating the total out-degree of a node.

The importance of a species is then compared through the use of three of the most common centrality metrics. These are:

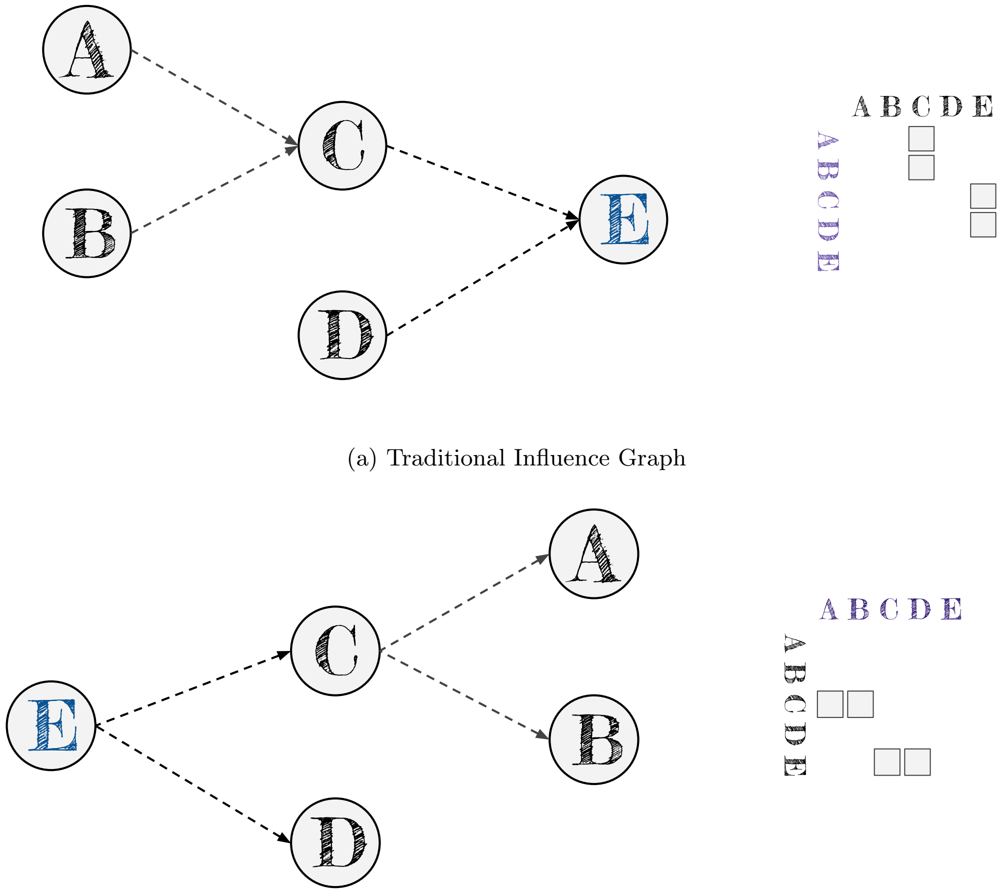
- **Centrality** - This describes how easily information from one node can be disseminated to all other nodes.
- **Betweenness** - This describes the number of shortest paths (fastest fluxes/greatest influences) that are routed between nodes adjacent to our chosen node. Species with a high betweenness hold a brokering position, and can act as a bottleneck between different groups of chemistry.
- **PageRank** - PageRank looks at the flow in a system. It ranks nodes not only on the number of species it reacts with, but also the importance of the species it has reacted with

Finally the 'Metric Sum' is the sum of all the metric values scaled between 1 and zero (the mean).

1.1.4 •

1.2 Calculating production sensetivity using personalised page rank.

Figure ??Figure ??



(a) Traditional Influence Graph

(b) Reversed-link (adjoint) influence graph

Figure 1.13: **Link reversal of the Jacobian Sensetivity matrix graph results in a graph of the Adjoint.** Showing how in changing the direction of the links in a graph is equivalent to applying the transpose to an adjacency matrix (right). In the case of a Jacobian based graph, this is analogous to using the adjoint to propagate the model back in time - something that can be used to identify the influence upon a species with a model.

1.2.1 Testing

Borneo

As with all scintific processes, it is importatnt to first test the algorithm on a small, comprehensible example. To do this we start with the creation of CH_2OO. This is a direct product of isoprene. In tracing back all the precursors the mechanism for its creation can be described as:

$$O_3 + C_5H_8 \xrightarrow{\kappa} CH_2OOE + (MACR\ or\ MVK)CH_2OOE \longrightarrow [\kappa_{dec}]\, CH_2OO \qquad (1.5)$$

In traversing the adjoint/reversed graph, this presenets a singe ‘shortest path’ between the product and its precursor. This creates for a base test for the algorithm. The PageRank algorithm is now run with a personalisation vector consisting with an value of 1000000 for the species of interest and -1 for all others. A damping factor value of 0.01 is also used for the algorithm.

As CH_2OOE only has one precursor (α-pinene) the initial test is done on this. From this the identification of isoprene as a source is sucessful, although since the algorithm is performed on the whole network, there are results for a number of additional species, Table **??**. This is because page rank works on using teleporation to change between items in the evolution of the system. With the design of the personalisation vector, these values will however be significantly smaller than any containing useful results.

	1
0	
C_5H_8	9.920000e-03
CH_2OOE	9.920000e-01
$C_{816}O$	-9.990000e-07
$NC_{101}CO$	-9.990000e-07
$C_{926}OH$	-9.990000e-07

Table 1.2: A reversed graph Page Rank test with $C_5H_8 + O_3 \longrightarrow CH_2OOE$ as the only reaction.

Next we apply the same methodology to CH_2OO. This creates the graph in Figure **??**. Here it is seen that CH_2OO is directly dependant on the radicals $CH_2OO[F, B, C, G, A]$, and CH_2OOE. This is then dependant on Isoprene, which then has a range of dependancies with all have precursors of their own (not shown).

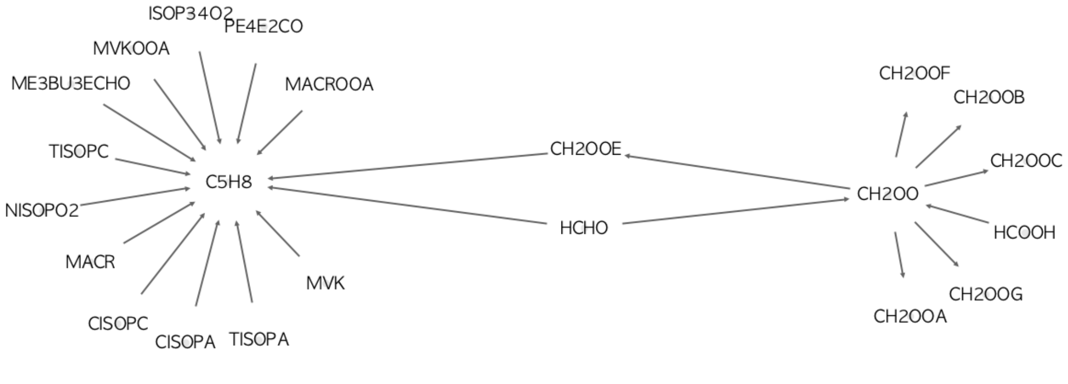


Figure 1.14: **The reversed subgraph between Isoprene, CH_2OOE and CH_2OO.** This is a subgraph of the afforementioned species, showing them and their neighbours. Here the arrows point towards a species precursor.

	1
0	
CH2OO	0.992000
CH2OOE	0.001670
CH2OOF	0.001660
CH2OOG	0.001660
CH2OOA	0.001660
CH2OOC	0.001640
CH2OOB	0.001640
C5H8	0.000016
MACR	0.000016
C2H4	0.000007
HMACR	0.000007
ISOP34NO3	0.000005
ME3BU3ECHO	0.000005
ISOPDNO3	0.000004
C622CHO	0.000002
C624CHO	0.000002
C518CHO	0.000002
C729CHO	0.000002
LIMAL	0.000002
C3H6	0.000001
BUT1ENE	0.000001
PE4E2CO	0.000001
MVK	0.000001
MVKOH	0.000001
ISOPBNO3	0.000001
ACR	0.000001

Table 1.3: A reversed graph Page Rank test with $C_5H_8 + O_3 \longrightarrow CH_2OOE$ as the only reaction.

Bibliography

Beel, J., Gipp, B., Langer, S., and Breitinger, C. (2016). Research-paper recommender systems: A literature survey. *International Journal on Digital Libraries*, 17(4):305–338.

Bloss, W. J., Lee, J. D., Bloss, C., Heard, D. E., Pilling, M. J., Wirtz, K., Martin-Reviejo, M., and Siese, M. (2004). Validation of the calibration of a laser-induced fluorescence instrument for the measurement of oh radicals in the atmosphere. *Atmospheric Chemistry and Physics*, 4(2):571–583.

Cajal, S. R. (2020). Cortex drawings. *web*. Accessed: 2020-2-4.

Cornell, L. (2020). Mark 1 Perceptron. `https://en.wikipedia.org/w/index.php?title=Perceptron&oldid=935763442`. Accessed: 2020-2-4.

Dillon, T. J., Tucceri, M. E., and Crowley, J. N. (2006). Laser induced fluorescence studies of iodine oxide chemistry part ii. the reactions of io with ch3o2, cf3o2 and o3. *Phys. Chem. Chem. Phys.*, 8:5185–5198.

Ellis, D. (2019). Using Tf-Idf To Form Descriptive Chapter Summaries Via Keyword Extraction. https://towardsdatascience.com/using-tf-idf-to-form-descriptive-chapter-summaries-via-keyword-extraction-4e6fd857d190. Accessed: 2020-2-5.

Géron, A. (2017). *Hands-On Machine Learning With Scikit-Learn And Tensorflow: Concepts, Tools, And Techniques To Build Intelligent Systems*. O'Reilly Media.

John Hay, Ben Lynch, D. S. (1960). Mark 1 Perceptron Operators' Manual. *Cornell Aeronautical Laboratory*.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mquantin (2020). Idf response functions. Accessed: 2020-2-5.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386.