

UNIVERSITY OF CRETE - DEPARTMENT OF PHYSICS



UNDERGRADUATE THESIS

Machine Learning-driven Deweathering of air pollutant concentrations at Finokalia Monitoring Station

Danai Filippou

Heraklion, June 2021

Supervisor: Prof. Maria Kanakidou

Acknowledgements

First and foremost, I would like to express my gratitude to Prof. Maria Kanakidou for her valuable guidance and support throughout this project. I would also like to thank Prof. Giorgos Tsironis and Prof. Nikolaos Mihalopoulos for agreeing to be part of my Thesis Presentation committee. Special thanks to all the members of the ECPL group and especially Dr. Nikos Kalivitis who was always willing to help with whatever question I encountered. Of course this work would not have been completed if not for the valuable help of Prof. Zongbo Shi and Dr. Congbo Song from the University of Birmingham. Finally, I would like to thank my family and friends for always supporting my decisions and for being there to encourage me during this endeavor.

Abstract

For the most part, air pollution is governed by emissions, but it can be affected by meteorological conditions as well. Due to the complex nature of the atmosphere, it is no easy task to decouple the effect of weather on measured concentrations of aerosols and thus uncover the true sources of pollution. In this study, the robust method of Deweathering (also referred to as Meteorological Normalization) will be used for the first time on concentrations measured at the Finokalia Monitoring Station, to examine the impact of meteorology in the area. The variables that will be considered are ground measurements from the station, elements of air mass back trajectories analysis, ERA5 reanalysis meteorological data as well as temporal variables. The pollutants of interest include the Total Particle Number concentration, the Aitken, Accumulation and Nucleation Particle Number concentrations, and finally Carbon Monoxide and Black Carbon concentrations. The deweathered values will be compared to the observations in order to draw conclusions about the sources of local pollution. Tracing the anthropogenic emission sources and investigating them separately from natural causes is crucial to evaluate environmental policies regarding the decrease of air pollution. An attempt will be made to examine the impact of the IMO2020 shipping fuel regulation. It is also discussed how the models appear to have difficulties in handling the dependencies of back trajectory clusters and signals of mixed sources.

Contents

1. Introduction and Objectives.....	4
2. Methodology	7
2.1. Machine Learning.....	7
2.2. Decision Trees	12
2.3. Random Forests.....	14
2.4. Meteorological Normalization	15
3. Modeling with R.....	18
3.1. Data Sources and Preprocessing	18
3.2. Random Forest model and Meteorological Normalization.....	19
4. Results and Discussion	21
4.1. October 2019 – March 2020	21
4.2. October 2018 – March 2019	28
4.3. Evaluation of the IMO2020 regulation	33
4.4. Conclusions and Future Perspectives.....	34
5. References	36
Appendix A: Codes and Methods	39
A1. Trajectory Clustering	39
A2. Final Preprocessing	40
A3. Meteorological Normalization.....	40

1. Introduction and Objectives

Air pollution greatly affects public and individual health at a global scale. According to the World Health Organization, poor ambient air quality is responsible of nearly four million deaths worldwide every year [17], while recent studies also show that it can harm the majority of organ systems in the human body.[18] Apart from health, emissions of pollutants lead to the increase of greenhouse gas concentrations. Thus, air pollution is closely linked to climate change as well.

To counter air pollution issues, various environmental policies have been imposed by governments and organizations. For example, the Beijing Municipal Government released a 5-year action plan (2013-2017) according to which the mean concentrations of $PM_{2.5}$ should be reduced to less than $60 \mu\text{g}/\text{m}^3$ by 2017. Evaluating the effectiveness of such policies can be challenging [8], mainly because apart from reduction in emissions, decrease in measured concentrations can be attributed to the effect of meteorology. Weather conditions can significantly affect air pollution and aerosol distribution. Strong wind on a windy day can clean a city's atmosphere, while high humidity rates can worsen it. Also, intense sunshine and high temperatures can cause or contribute to chemical reactions. Finally, rainfall can normally result in less pollution, since it might lead to washout of particles. These meteorological effects, which can falsely reinforce or cloak atmospheric measurements, must be taken into consideration, in order to uncover the true efficiency of environmental policies.

A novel way to decouple the impact of weather is the Meteorological Normalization technique, also referred to as Deweathering. [4, 5, 8] It makes use of the Random Forest algorithm, one of the most powerful models that exist in the field of Machine Learning. The Random Forest model is trained on meteorological data of the study period, and consequently it is used to calculate meteorologically normalized values of aerosols and gas pollutants.

As of 1st January 2020, the International Maritime Organization has put into practice the IMO2020 global regulation, according to which the sulphur content of shipping fuels should not exceed the limit of 0.5%. This is a 3% difference compared to the 3.5% limit that existed before this policy was applied. Following the implementation of the new regulation, a decrease as high as 77% in SO_x emissions is anticipated. [19]

As the Mediterranean Sea is one of the world's busiest waterways, and given the fact that the sulphur content in aerosol concentrations is significant, changes in aerosol distribution should emerge in the area. The aim of this study is to use Meteorological Normalization to detect these changes by comparing deweathered values of aerosol size distributions of the year 2020

to previous years. This way, evaluation of the IMO2020 regulation can be performed as well, for the area of the Mediterranean. The aerosol data that are analyzed were obtained from the Finokalia Monitoring Station, located in Northeast Crete and operated by the Environmental Chemical Processes Laboratory (ECPL) of the Department of Chemistry (University of Crete). The station has an altitude of 150m and it is located 70 km from Heraklion.

Aerosols in general are defined as tiny solid or liquid particles dispersed in a gas. Their diameters range from a few nanometers to tens of micrometers. There are different categories of aerosols resulting from their various characteristics. Depending on whether they are emitted directly or emerge as products of chemical processes in the atmosphere, they are characterized as primary and secondary aerosols accordingly. Their sources can be either anthropogenic (aviation, industrial activity, energy production, shipping etc.) or natural (sea salt, mineral dust, volcanic ash etc.). Apart from primary emissions, aerosols can be changed or formed from condensation of vapor, by coagulation with other particles or by taking part in chemical reactions. Their removal mechanisms include dry and wet deposition. Dry deposition occurs through direct transfer on the Earth's surface, whereas wet deposition is the transfer through rain, snow or fog, in which cases aerosols are trapped within droplets and washed out with precipitation. Aerosols can also serve as cloud condensation nuclei at first and subsequently they can be swept away, again through precipitation. In general, aerosols also vary in size and they can be grouped based on their diameters. The two main categories are coarse particles (with $D > 2.5 \mu\text{m}$) and fine particles (with $D < 2.5 \mu\text{m}$). Fine particles are further classified into accumulation mode ($D = 100 \text{ nm} - 2.5 \mu\text{m}$), nucleation mode ($D < 25 \text{ nm}$) and Aitken mode ($D = 25 - 100 \text{ nm}$).

Aerosols may have direct and indirect effects on global climate. The direct effects refer to the scattering of solar radiation that would otherwise reach the Earth's surface. This decrease in the incoming solar radiation that eventually makes it on the surface results in its cooling. Aerosols can also absorb significant amounts of this incoming solar radiation, which in turn leads to temperature increase. On the other hand there is also a certain indirect effect that has to do with aerosols acting as cloud condensation nuclei and affecting cloud formation, along with their properties. All of these effects that aerosol have on climate is what makes them important and worth investigating. In this study we will focus separately on each of the three modes: the accumulation, the nucleation and the Aitken mode.

In order to examine the composition of atmospheric aerosols with regard to ship fossil fuel emissions we will investigate Black Carbon content (BC, also known as soot or elemental carbon). It is produced by the incomplete combustion of fossil fuels and biomass burning. It is dark in color (hence the term 'black' carbon) and it can strongly absorb sunlight entering the Earth's

atmosphere, resulting in its warming. BC also contributes to the melting of glaciers and snow, because it darkens their surfaces and lowers their albedo. So, instead of reflecting the sunlight, these cold surfaces absorb it and melt faster than usual. [21] However, even if BC can do great harm, it is a short-lived pollutant with a lifetime of a few days to a few weeks. This is quite beneficial because any measures that might be taken to reduce BC concentration would lead to immediate and apparent results.

Another compound of interest for this study and the only in the gas phase that we will examine is Carbon Monoxide (CO). CO is a colorless and odorless gas that can be lethal when inhaled in large amounts, because it reduces the amount of oxygen transported to vital organs of the human body, like the heart and brain. It is considered a primary pollutant, its lifetime in the troposphere spans from 30-90 days and its effect on the environment includes the reaction with the hydroxyl radical OH, a process that produces CO₂. Also, in an environment where NO_x (NO+NO₂) concentration is high, the rate of O₃ production increases linearly with CO concentration. That being said, the presence of CO may indirectly contribute to the concentrations of greenhouse gases, like CO₂ and O₃. CO is the product of CH₄ oxidation, biomass burning and incomplete combustion of fossil fuels. It is estimated that 60% of CO concentrations stem from anthropogenic activities [15], and thus CO is a pollutant of great interest for this study, since it could be closely linked to shipping emissions.

2. Methodology

2.1. Machine Learning

Machine Learning is a field of Artificial Intelligence that focuses on enabling computers to learn from data without being explicitly programmed. Over the years, the area of Machine Learning has grown significantly as many of its applications are taking part in our everyday lives. Image and Speech recognition, Spam filters for e-mails, Plagiarism Detectors for school assignments and Smart Personal Assistants are only a few examples. Apart from this usage, Machine Learning algorithms can prove to be valuable tools for scientific research as well since they are very efficient in data modeling and prediction.

A key procedure in Machine Learning is the training process, where the Machine Learning algorithm receives data to learn from. Although Machine Learning methods can be categorized in many ways, perhaps the most fundamental is the classification based on the supervision they receive during the training process. The main categories are Supervised Learning, Unsupervised Learning and Reinforcement Learning.

In Supervised Learning, the model receives training data which contains inputs and their corresponding outputs. For example, in the spam filter algorithm the model receives a training set of e-mails along with their class (spam or ham), from which it will learn to categorize new input e-mails that do not belong in the initial training set (Fig. 2.1). This is a classification task, the first variation of Supervised Learning problems. The second variation is the regression tasks, where the model needs to predict target values of a variable, for example the daily temperature of a certain period, given parameters like the month, the daily humidity and the year. In both of these variations there are two very important attributes: the features, which are measurable quantities that are fed to the model during training, and the labels, which are the desired outputs during the model's prediction stage (but note that they are also included in the training set along with their corresponding features). In the spam filter example, the features are the e-mails that will be used for training, while the labels are the two possible classes (spam or ham). In the temperature forecasting example, the features are the month, the daily humidity and the year, while the label is the target value of the temperature. The most common Supervised Learning models are Support Vector Machines, Decision Trees and Random Forests.

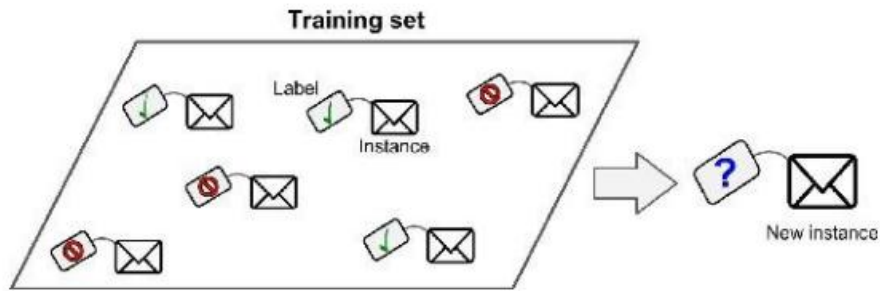


Figure 2.1: Supervised Learning – Spam Filter example [3]

In Unsupervised Learning, the training data is unlabeled, meaning it contains only inputs, and the model needs to uncover patterns of this input on its own. A typical Unsupervised Learning task is Clustering (Fig. 2.2) where the model processes the training data and tries to split it to matching groups.

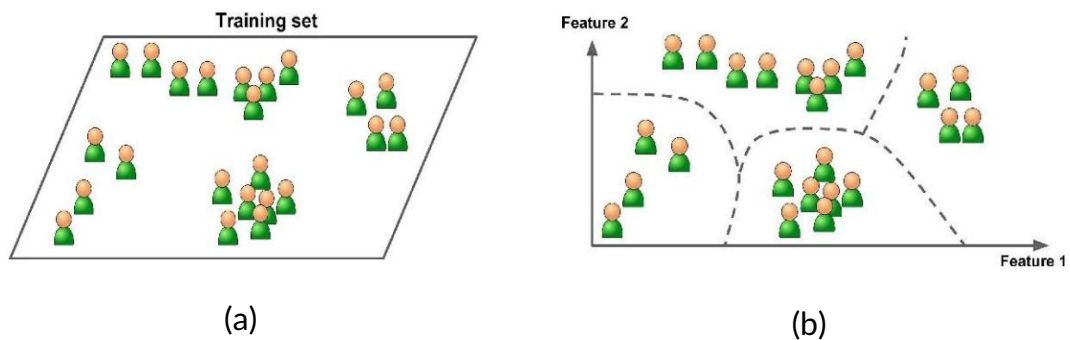


Figure 2.2 : a) Unsupervised Learning dataset, b)Clustering [3]

In Reinforcement Learning, the Machine Learning model observes an environment, within which a goal must be achieved, for example a car controlled by the model must climb a hill. In this case, the car is called agent, and it performs certain actions that result in rewards or penalties, depending on whether these actions contributed to the completion of the task assigned to the agent. The model needs to identify the best approach (called policy) in order to reach the initial goal, based on the feedback it receives. Another example of this concept, pictured in Fig. 2.3, is that of a robot making use of Reinforcement Learning in order to learn how to walk.

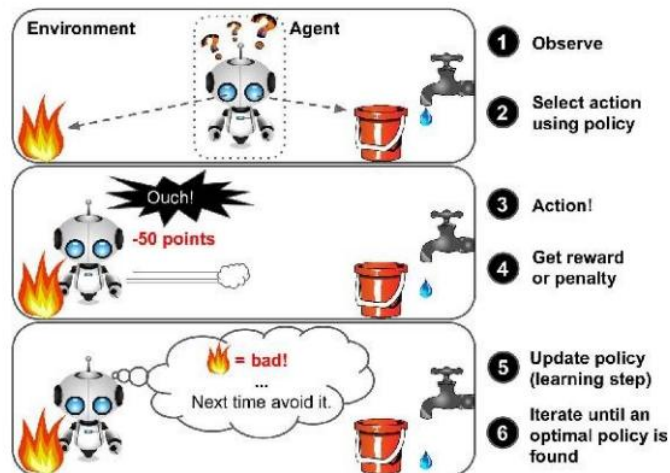


Figure 2.3: Reinforcement Learning with a robot as agent [3]

As data is the bedrock of Machine Learning, it must be thoroughly studied and prepared before given as input to a model. This strategy includes various methodologies like Feature Importance investigation (during which features irrelevant to the problem are dropped, since they are not helpful for making predictions) or outlier and missing values detection. Outliers in general are out-of-range numeric values. Another useful method for data preprocessing is Feature Scaling, according to which features that have very different scales are transformed to obtain values belonging to a certain range (e.g. 0-1). In some cases there is a huge number of instances for the same feature, which significantly slows down training and induces undesired complexity to the problem. To address this issue it is best to apply Dimensionality Reduction, a process that, as the name implies, lowers the dimension (and thus the number) of features. PCA (Principal Component Analysis), one of the most widely used Dimensionality Reduction techniques, finds the plane that is closest to a set of features, and projects the data onto it (Fig. 2.4).

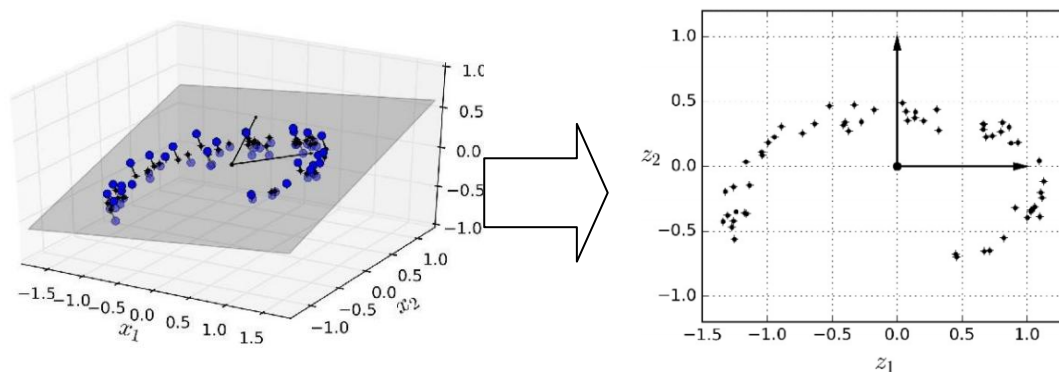


Figure 2.4 : Data projection and Dimensionality Reduction [3]

After the preprocessing stage is done, the data must be split to training and test set, usually with a percentage ratio 80% to 20% or 70% to 30%. The model will learn from the data that is part of the training set, while the test set will be used to see whether the model is able to adapt to new data and make accurate predictions. Sometimes a third set is utilized as well, called validation set, to evaluate the model's performance before it starts making predictions on the test set. The validation set is a subset of the training set, usually at a percentage of 10-20%.

The data split is followed by the model selection, depending on the problem's category (Supervised, Unsupervised, or Reinforcement Learning). For the Supervised Learning case it must be further specified whether we are dealing with a Classification or Regression problem. Then the training data must be fed to the model, which processes it and makes a prediction. The difference between a predicted and a true value of the dataset is measured by a cost function. So during training, the model attempts to minimize the cost function, indicating that its predictions approach the true values. Examples of cost functions include the Mean Squared Error, the Root Mean Squared Error and the Mean Average Error:

$$MSE = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2 \quad (\text{Equation 2.1})$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2} \quad (\text{Equation 2.2})$$

$$MAE = \frac{1}{m} \sum_{i=1}^m |\hat{y}^{(i)} - y^{(i)}| \quad (\text{Equation 2.3})$$

In these equations, m is the number of instances in the dataset, i is the index of the instances, $\hat{y}^{(i)}$ is the predicted value, and $y^{(i)}$ is the true value.

After training is complete, the next step is to use the model to make predictions on the test set, and evaluate its performance. Common metrics for this procedure are the following (where P : Predicted value and O : Observed value):

- $FAC2$, represents the percentage of predictions distanced within a factor of 2 from the true values: $0.5 \leq \frac{P}{O} \leq 2.0$
- Mean Bias error: $MB = \frac{1}{n} \sum_{i=1}^n (P_i - O_i)$
- Mean Absolute error: $MAE = \frac{1}{n} \sum_{i=1}^n |P_i - O_i|$

- Normalized Mean Bias: $NMB = \frac{\sum_{i=1}^n P_i - O_i}{\sum_{i=1}^n O_i}$
- Normalized Mean Gross Error: $NMGE = \frac{\sum_{i=1}^n |P_i - O_i|}{\sum_{i=1}^n O_i}$
- Root Mean Squared Error: $RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (P_i - O_i)^2}$
- Pearson correlation coefficient r , represents the strength of the linear relationships between two variables, the closer it is to 1 or -1, the more linear the relationship is, while if it is 0, the relationship is in no way linear
- Coefficient of Efficiency, describes the model's ability to make predictions close to the observed mean. A perfect model has $COE=1$, while $COE=0$ indicates that the closer the prediction can get to the actual observed value is the observed mean, and not further than that:

$$COE = 1 - \frac{\frac{1}{n} \sum_{i=1}^n |P_i - O_i|}{\sum_{i=1}^n |O_i - \bar{O}|}$$
- Index of Agreement IOA , which ranges from -1 to +1 (with +1 corresponding to a perfect model). It describes the relationship between the sum of the error-magnitudes with regards to the sum of the observed-deviation magnitudes. An IOA of 0.5, for example, indicates that the sum of the error-magnitudes is one half of the sum of the observed-deviation magnitudes. When $IOA = 0.0$, it signifies that the sum of the magnitudes of the errors and the sum of the observed-deviation magnitudes are equivalent.

It is not unusual for a model to perform extremely well on the training set, and then fail to make accurate predictions on the test set. That being said, the good scores the model achieves on the training set can be misleading. This is called Overfitting and it is one of the main problems Machine Learning algorithms face. The model learns the training set 'too well' and it is not able to generalize on new data. Overfitting is generally confronted with regularization: the model is constrained and its structure becomes less complex. The amount of regularization applied is controlled by the hyperparameters, model parameters that are set before training starts and remain constant until it is finished. Since they are a convenient way of editing the learning process of the model, hyperparameter tuning is a really crucial stage. The perfect balance must be found between keeping the model simple enough and acquiring the best prediction accuracy possible.

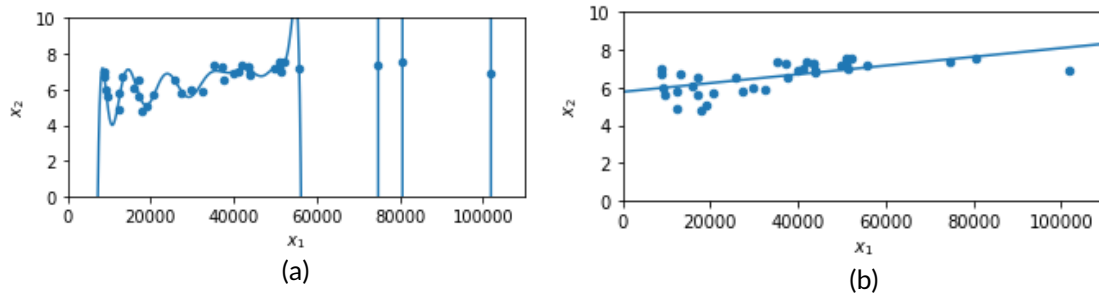


Figure 2.5: a) Overfitting the training data with a Polynomial Regression (degree=60), b) Linear Regression, which is more trustworthy regarding predictions in this case

2.2. Decision Trees

Decision Trees are among the most important Machine Learning algorithms. They are also known as the CART algorithm (Classification and Regression Tree) and they consist of repetitive binary splits of the dataset, according to certain conditions at each split. To put it simply, the algorithm asks questions and answers them based on the data, and the procedure continues until a prediction is reached.

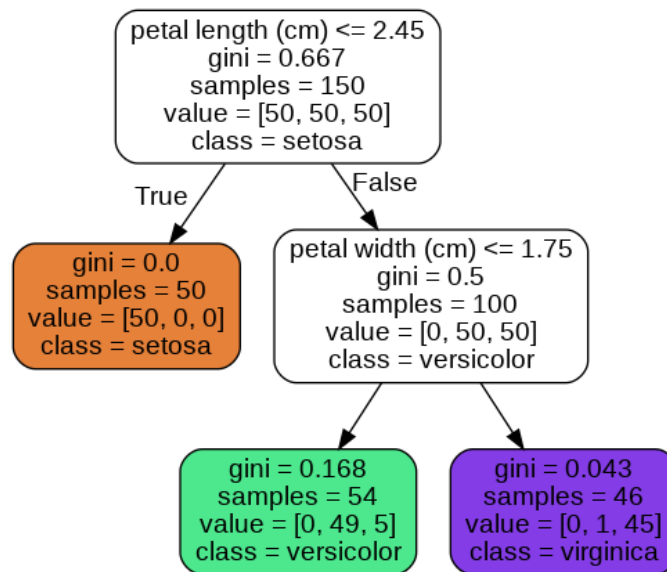


Figure 2.6: Classification using Decision Trees [3]

A Classification task is depicted in Fig. 2.6. The first condition, placed in the root node, is whether the value of a feature, in this case the ‘petal length’, is smaller than 2.45 cm. The algorithm then divides the data in two subsets, according to the condition’s outcome. The True outcome results in only one final child node (called leaf node). On the other hand, the False outcome results in a child node that is then split once again, with a different question asked (Is the petal width smaller than 1.75 cm?) and then the two final leaf

nodes are reached. Apart from the conditions placed, on each node there are more values depicted as well. The 'gini' (or 'entropy' in some cases) is a measure of a node's impurity. A pure node has a 'gini' value of 0 and all instances that it encloses belong to the same class, as it can be seen from the orange node in Fig.2.6. The 'samples' attribute corresponds to that dataset instances for which the node's condition is True. The 'value' attribute expresses the class distribution of the node's 'samples'. Finally, the 'class' attribute is the predicted class of the Decision Tree.

Decision Trees can also be used for Regression tasks. Again, the splitting method is applied according to the outcome of each node's condition. This time, however, the Decision Tree predicts a mean value, not a class, and this mean value is matched to a certain number of samples. For example, in Fig. 2.7a), in the far left leaf node, 20 samples have been matched to a mean value of 0.854. Mean Squared Error is used for evaluation of the predictions. In Fig. 2.7b) the analogy of the mean values and the number of samples can clearly be observed: the plot represents the results of the 4 leaf nodes.

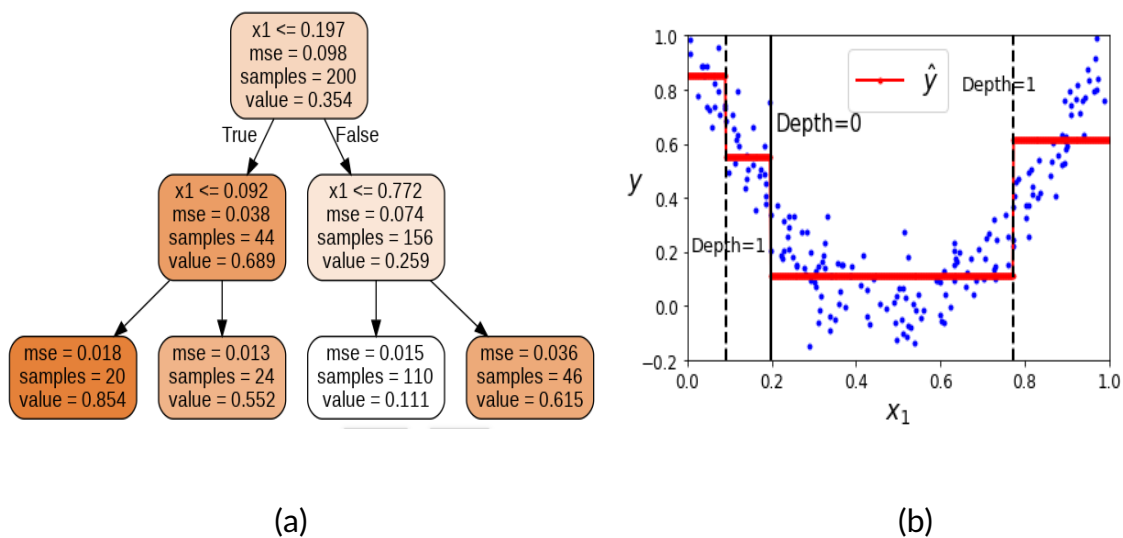


Figure 2.7: a)Regression using Decision Trees, b)Visualization of results [3]

Decision Trees, as all Machine Learning algorithms, are prone to Overfitting. If left unconstrained, they will fit the training data too closely (Fig.2.8), a problem that can be resolved by applying regularization. The most important hyperparameters that can be tuned are: `min_samples_split` (minimum number of samples before a node can split), `min_samples_leaf` (minimum number of samples within a leaf node), `max_leaf_nodes` (maximum number of leaf nodes) and `max_features` (maximum number of features that are evaluated before a node splits), and `max_depth` (depth of the tree).

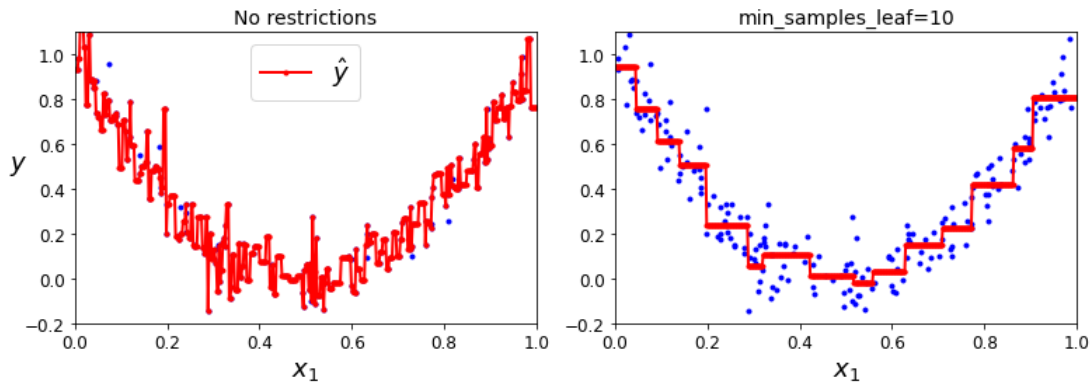


Figure 2.8: The difference of fitting between a) a Decision Tree that is left unconstrained (heavy overfitting occurs), and b) a Decision Tree with the `min_samples_leaf` hyperparameter regularized [3]

2.3. Random Forests

In Machine Learning the process of grouping together models in order to obtain a computationally better one is called Ensemble Learning. A Random Forest is an Ensemble method composed of many Decision Trees, trained on different subsets of data. When these subsets are selected with replacement, the process is called bagging (or bootstrap aggregating). If replacement does not occur, the process is called pasting. Random Forests usually work with bagging, and each Decision Tree is trained on a different bootstrapped dataset. This is exactly why Random Forests are a very powerful Machine Learning model: they are simultaneously trained on different parts of the original dataset, enabling variety in the learning process. When used on Classification, the predictions are made based on which class has been predicted more frequently from the Decision Trees. On the other hand, a prediction in a Regression task is the mean value of all the predictions made by the Decision Trees of the Random Forest.

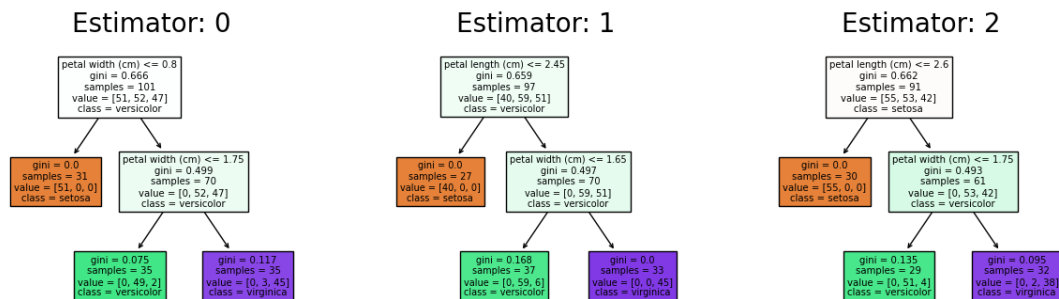
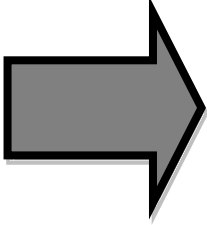


Figure 2.9: Visualization of a Random Forest with only 3 Decision Trees (`n_estimators=3`, `max_leaf_nodes=3`)

The creation of a bootstrapped dataset, using a small part of the IRIS dataset (containing 3 iris flower classes, 50 instances of each class), is depicted in Fig. 2.9. Bootstrap aggregating enables some training instances to be selected multiple times (pink row) while others may not be selected at all (blue row). The latter are called Out-Of-Bag instances and typically they exist in a percentage of 30% of the original data. They often serve as a validation dataset. The most significant hyperparameters of a Random Forest are the aforementioned Decision Tree hyperparameters, along with `n_estimators` (number of Trees in the Forest), `max_samples` (maximum number of samples included in the subsets of data), and `bootstrap` (boolean, if it is False then the whole dataset is used by each tree and there are no subsets).

Table 2.1: Creating a bootstrapped dataset. Pink indicates instances selected multiple times, while blue indicates the out-of-bag instance

Petal Length (cm)	Petal Width (cm)	Iris Flower Species		Petal Length (cm)	Petal Width (cm)	Iris Flower Species
1.4	0.2	Setosa		1.4	0.2	Setosa
5.9	2.1	Virginica		2.7	4.6	Versicolor
2.8	4.6	Versicolor		3.2	1.3	Setosa
2.7	4.6	Versicolor		3.2	1.3	Setosa
3.2	1.3	Setosa		5.9	2.1	Virginica

2.4. Meteorological Normalization

Meteorological Normalization is a technique aimed at decoupling the impact of meteorology on atmospheric time series and it uses the Random Forest algorithm to do so. The Random Forest is able to predict concentrations for a given timestamp but with randomly selected weather conditions, hence the term 'Normalization'. [5]

The original dataset contains both meteorological and temporal variables, which serve as the features. There is also the target variable (the label of the problem), which is usually an aerosol's concentration. This whole initial dataset is split into a training and a test set, with the training set being randomly resampled, following the bootstrap aggregating method (mentioned in Section 2.3). Then a certain number of Decision Trees is prepared, while each tree is trained on a different bootstrapped dataset. The combination of the Decision Trees results in a Random Forest which, after the training is done, is evaluated on the test set and is also used to conduct the Meteorological Normalization (Fig. 2.10).

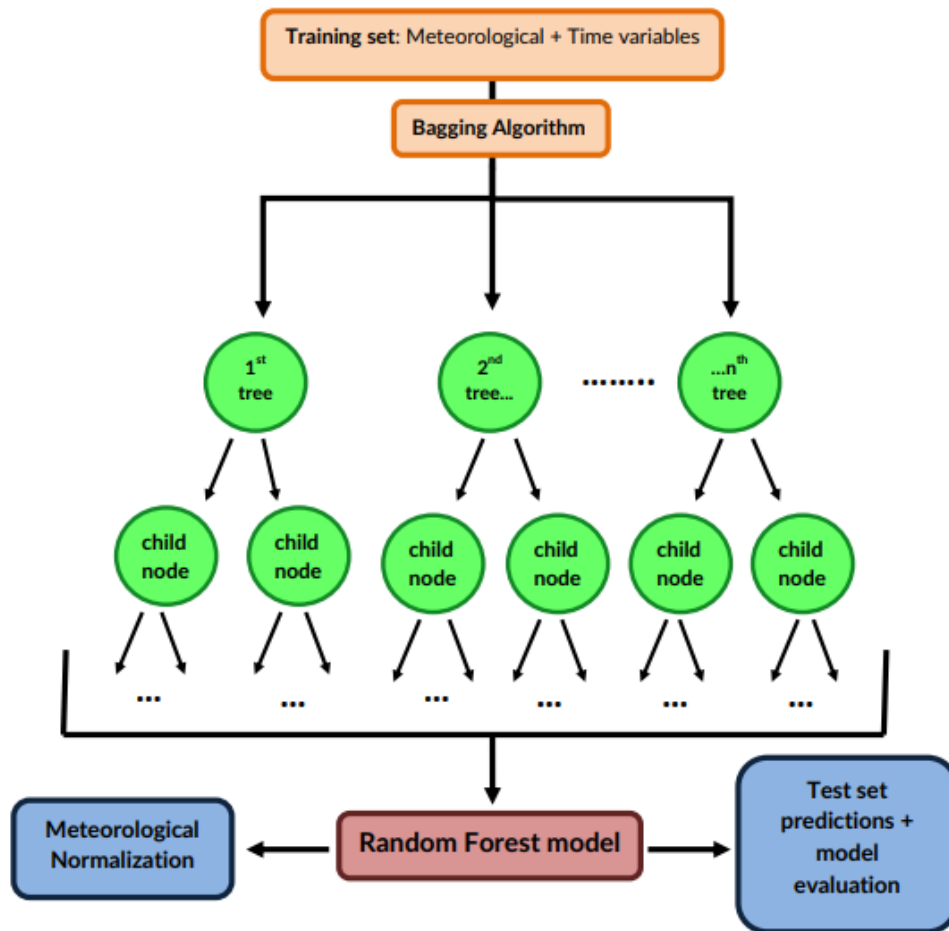



Figure 2.10: Random Forest construction

The next step is to use the Random Forest model to calculate the weather normalized (or deweathered) concentrations. Weather conditions are assigned randomly to a particular timestamp (e.g. 24/8/2019, at 14:00:00) and the model makes predictions for the concentration. This process is repeated 1000 times, with different meteorology each time, acquired from the whole study period. This results in 1000 predictions for this exact timestamp, which are then averaged into one, the meteorologically normalized (or deweathered) value (Table 2.2). The same process is repeated for all the timestamps and the normalized dataset is obtained. Note that sometimes during the normalization step, apart from the meteorological conditions, the temporal (time) variables are randomly resampled as well. This process normalizes both seasonal and meteorological impact, and it is inadequate when the aim of the study is to investigate short-term changes in concentrations or even compare seasonal variations with changes in emissions, because the normalization removes the impact of both the weather and seasonality [8, 9].

Table 2.2: Exact implementation of Meteorological Normalization, for a **single** timestamp. The Random Forest receives randomly selected meteorology and makes predictions that are then averaged to provide the final normalized concentration.

Wind speed (m/s)	Temperature (°C)	Pressure (Pa)		Predicted Concentration (cm ⁻³)	Average Normalized Concentration (cm ⁻³)
9.62	9.88	98389.56		32.63	} 21.18
10.47	11.25	99209.72		29.90	
8.93	14.6	99357.87		20.45	
8.86	14.55	100775.20		12.81	
4.50	15.25	100669.10		10.09	

3. Modeling with R

3.1. Data Sources and Preprocessing

As mentioned above, the main focus area is the station of Finokalia. The data used were meteorological measurements as well as time variables of the study period 1/1/2015 - 21/10/2020. These included the exact date-time stamp of the measurement, the number of seconds passed since the UNIX epoch (1/1/1970 at 00:00:00), the yearly number of the week (1-52), the number of weekday (1-7), the hour value (0-23) and the day of the year (1-365).

Hourly ground measurements from the station were used, specifically for the quantities: Temperature, Wind Speed, Wind Direction and Humidity. The missing values that this dataset contained (174 for Wind Speed and Wind Direction, 175 for Temperature and 10866 for Humidity) were replaced using linear interpolation in Python.

ERA5 (ECMWF Reanalysis, 5th Generation) is a product released by the European Centre for Medium-Range Weather Forecasting (ECMWF), which provides reanalysis data. Reanalysis is a method that combines observations with model measurements, in order to obtain complete datasets that accurately describe global climate in hourly resolution. We acquired the following gridded data from ERA5: boundary layer height (determines the depth of air with regards to the Earth's surface which is most affected by resistance to transfer of momentum, heat or moisture across the surface), surface net solar radiation (solar radiation that reaches a horizontal plane at the surface of the Earth minus the amount reflected by the Earth's surface), surface pressure, total cloud cover and total precipitation.

Apart from ground measurements and the ERA5 data, the cluster of backward trajectories, indicative of air mass history, is needed as well. To obtain this variable we first acquired the data of 72-hour backward trajectories using the Hybrid Single-Particle Lagrangian Integrated Trajectory (HYSPLIT) model, at an hourly resolution and by setting the receptor height at 1000m. The HYSPLIT model is able to simulate air mass trajectories (either forward or backward), as well as transport, dispersion, deposition and chemical transformation events. The most widely used application, utilized in this project as well, is back trajectory analysis, where HYSPLIT is run to determine the origin (previous longitude, latitude and height) of different air parcels arriving at the area of interest at a given time.

The clustering process in the hourly trajectories was impossible to complete due to lack of computational memory, so we used a 4-hour resolution instead. The back trajectories were then clustered in 12 clusters. [9] Each cluster represents common air masses that the location of Finokalia is

exposed to. The clustering was done using the `TrajCluster()` function of the **openair** R package, which groups clusters according to the Euclidean distance method. [2] A visualization of the 12 clusters is depicted in Fig. 3.1, for clusters computed in 2017.

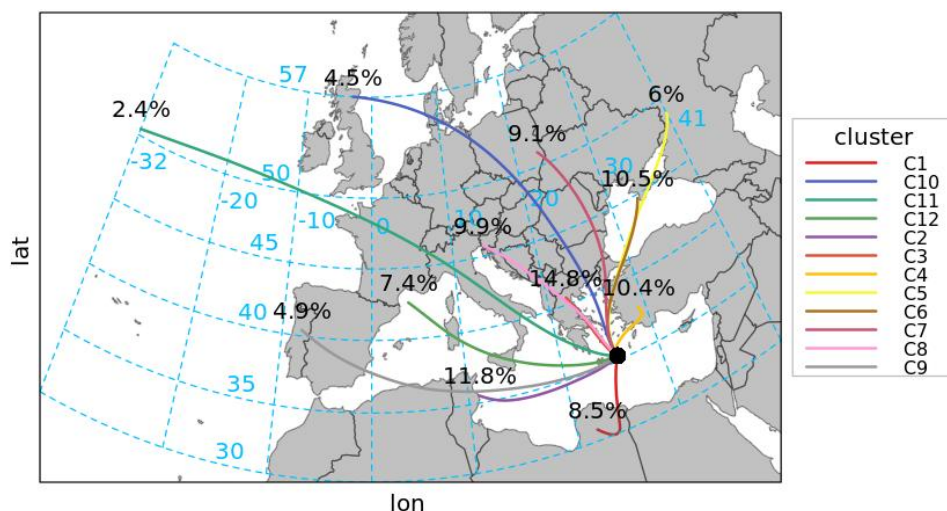


Figure 3.1: The twelve clusters of back trajectories, visualized using the `openair` package. The percentages indicate the frequency of occurrence.

Lastly, we acquired number concentration data from a Scanning Mobility Particle Sizer (SMPS) instrument, located at Finokalia Station. The particle diameters considered were 25-100nm (Aitken mode), 100-850nm (Accumulation mode), and 8-25nm (Nucleation mode). CO (measured with a Picarro) and Black Carbon (measured with an Aethalometer AE33) concentrations were also chosen as pollutants of interest. The Black Carbon was investigated both as an individual quantity, and also separately regarding contribution to total concentrations from fossil fuel combustion and wood burning. This led to a dataset consisting of 8 target values in total.

After the collection of data, preprocessing was done in order to prepare the final dataset for the Deweathering stage. This preparation was conducted in R and it included filtering all the available data to achieve a 4-hour resolution, in order to match the trajectory cluster variable.

3.2. Random Forest model and Meteorological Normalization

The data preprocessing stage results in a dataset which consists of the time variables (date, UNIX time, Julian day, weekday, hour), the meteorological variables (wind speed, wind direction, temperature, pressure, relative humidity, cluster of back trajectory, boundary layer height, surface net solar radiation,

total cloud cover, total precipitation, surface pressure), Particle Number and CO concentration data. Next, the implementation of the Random Forest and the Meteorological Normalization is conducted, using the **rmweather** package in R. [5] Two periods were investigated, October 2018 – March 2019 and October 2019 – March 2020.

The first **rmweather** function required is the `rmw_prepare_data()` which prepares the dataset for the Random Forest modeling and splits the available data into a training and a test set, at a percentage of 70% and 30% accordingly. The next step is to use the `rmw_do_all()` function to build and train the Random Forest model and immediately perform the Meteorological Normalization afterwards. Following previous works on the subject ([5], [9]), the hyperparameters for the Random Forest were set to `n_trees=300` and `min_node_size=3`. The 300 Decision Trees that made up the Random Forest were trained using the bagging algorithm (resampling with replacement). All the available features were used for training, whereas the normalization was conducted using only the weather variables, excluding the time variables. As mentioned in the previous chapter, this was done to enable investigation of the seasonality of weather normalized data.

Evaluation of the Random Forest was conducted with the `rmw_predict_the_test_set()` function. The predictions made on the test set were compared with the real ones to calculate various performance metrics, using the `modStats` function of the **openair** package. The **rmweather** package also offers functions which plot various quantities and are useful for assessing our methodology. The `rmw_plot_test_prediction()` function returns a plot, where the y-axis represents the values predicted by the Random Forest model and the x-axis represents the true values of the label. This scatter plot is a very convenient way to visualize the model's prediction accuracy. The codes for this section are presented in the Appendix.

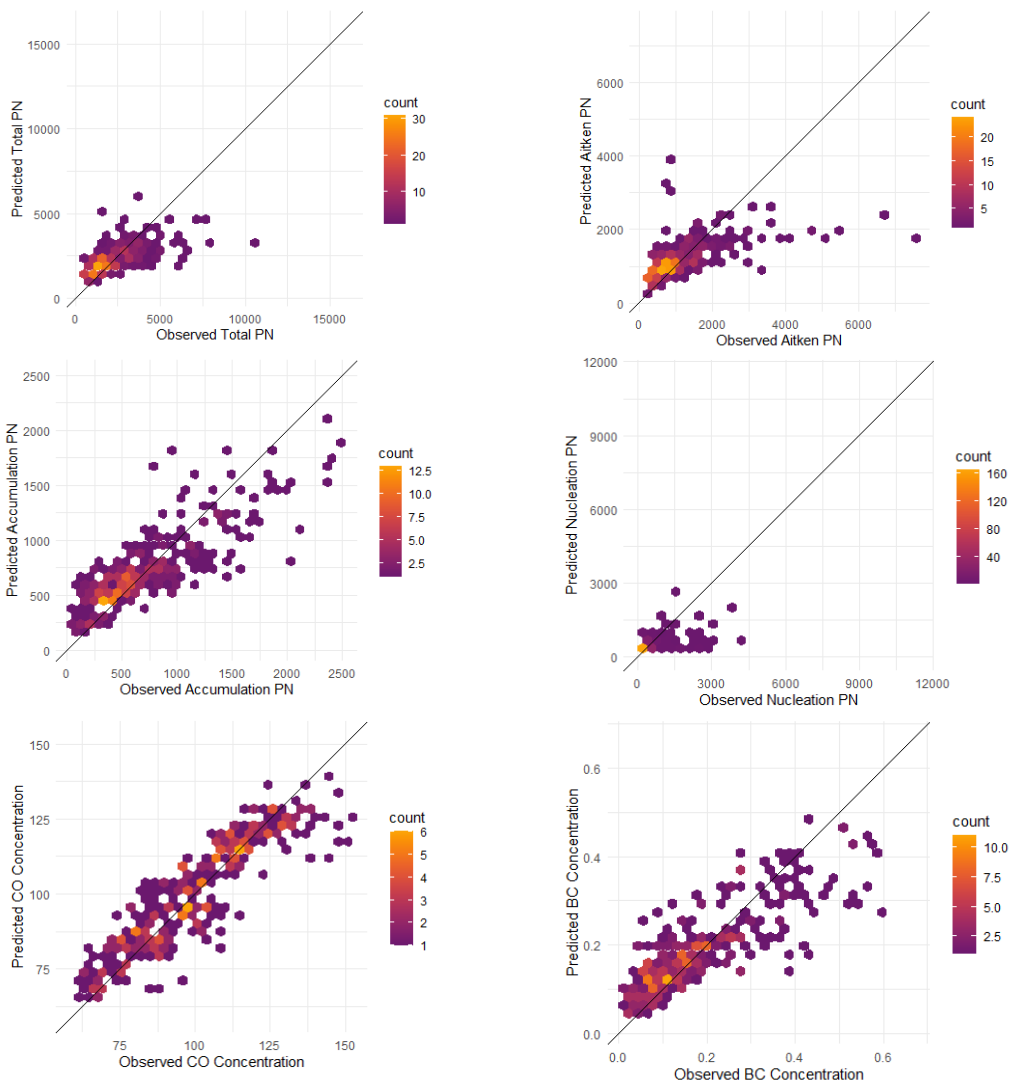
4. Results and Discussion

4.1. October 2019 – March 2020

The dates of the first run were 1/10/19 – 1/4/20. Training was conducted using 70% of the original dataset, and the remaining 30% which corresponded to the test set was used to calculate metrics and help evaluate the RF models. Table 4.1 presents all of the metrics, where n denotes the number of the test set instances for each target value, BCff denotes the Black Carbon percentage coming from fossil fuel combustion, while BCwb denotes the percentage from wood burning. In general the scores are high: the Pearson Coefficient of all the models remains constantly above 0.5 and in some cases it even approaches 1, which is the ideal value and indicates linear relationships between the variables. FAC2 metric is also calculated and it is over 0.7 in all cases, indicating agreement between the observed and the predicted values. IOA is also satisfying, achieving scores close to 0.5, with 1 being the ideal case. Nearly half of the models exhibit negative COE values. Generally, a negative COE value indicates that the mean of the observations is a better predictor than the model. However, as the rest of the metrics exhibit good accuracy of the models, the values of COE would not be significant to take into account. There is great difference between the RMSE, MGE, MB scores of the models and this is understandable since these metrics are scale dependent, and each target value has various scales. For example, the Total Particle Number ranges from 2000-12000 cm^{-3} while Black Carbon concentrations coming from fossil fuel combustion range from 0.1 – 0.6 $\mu\text{g m}^{-3}$.

To further investigate the accuracy of the models, plots of True and Predicted values are produced (Fig. 4.1). The density of the values near the $y=x$ line is high and this shows that the majority of the predicted values are the same (or close) to the observed (true) ones. This argument further justifies the conclusion that our models are good. In some plots there are some outliers, as in the Aitken mode or in the Accumulation mode, but these are all isolated special cases. In general, the model has difficulties predicting values of bigger ranges, and this is exactly why in the pollutant concentrations (CO, BC, BCff and Bwb) the scores and accuracy are higher.

Target Value	n	FAC2	MB	MGE	NMB	NMGE	RMSE	r	COE	IOA
Total (cm ⁻³)	315	0.89	-10	779	-0.004	0.340	1251	0.65	-0.319	0.340
Aitken (cm ⁻³)	315	0.81	-37	460	-0.031	0.390	773	0.60	-0.329	0.335
Accumulation (cm ⁻³)	315	0.88	-18	208	-0.024	0.284	278	0.83	0.157	0.578
Nucleation (cm ⁻³)	315	0.44	30	317	0.074	0.804	731	0.6	-0.582	0.209
CO (ppb)	318	1	-0.37	7.052	-0.004	0.07	9.217	0.89	0.551	0.775
BC (µg/m ³)	318	0.84	-0.003	0.057	-0.018	0.303	0.080	0.80	0.192	0.596
BCff (µg/m ³)	318	0.83	-0.0009	0.049	-0.006	0.316	0.073	0.82	0.238	0.619
BCwb(µg/m ³)	318	0.70	-0.002	0.014	-0.047	0.450	0.02	0.61	-0.303	0.348



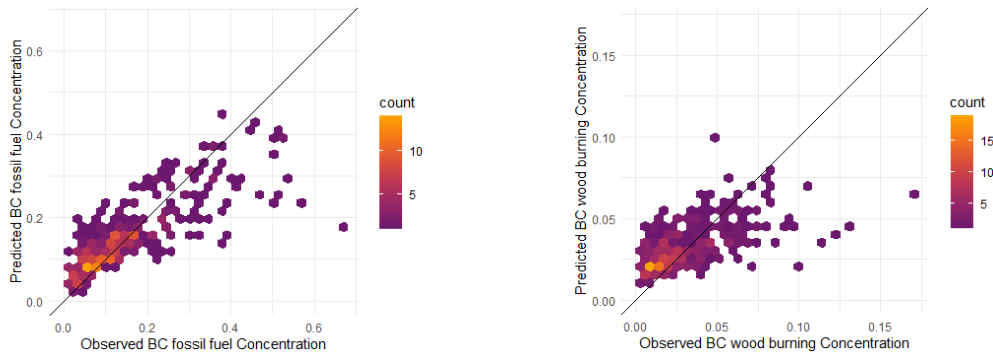


Fig. 4.1: True vs. Predicted values (1/10/19 – 1/4/20)

By taking into account scores and metrics reported in previous works [4, 9] we conclude that the scores of our models are very satisfying. Unfortunately, since this is a first study of Deweathering at Finokalia, we are unable to compare our scores to studies conducted in this particular area, in the past. It should be noted however that our scores are by no means low and were achieved while having few data in our disposal, compared to other studies.

Having completed training and evaluation of the Random Forest model, the next step is to perform the Meteorological Normalization. In order to examine its effect, we plot the observed and deweathered quantities in the same figures (Fig 4.2). The difference in magnitude of the two quantities is remarkable. To better interpret the results, average means of the study period of each target value were calculated (Table 4.2).

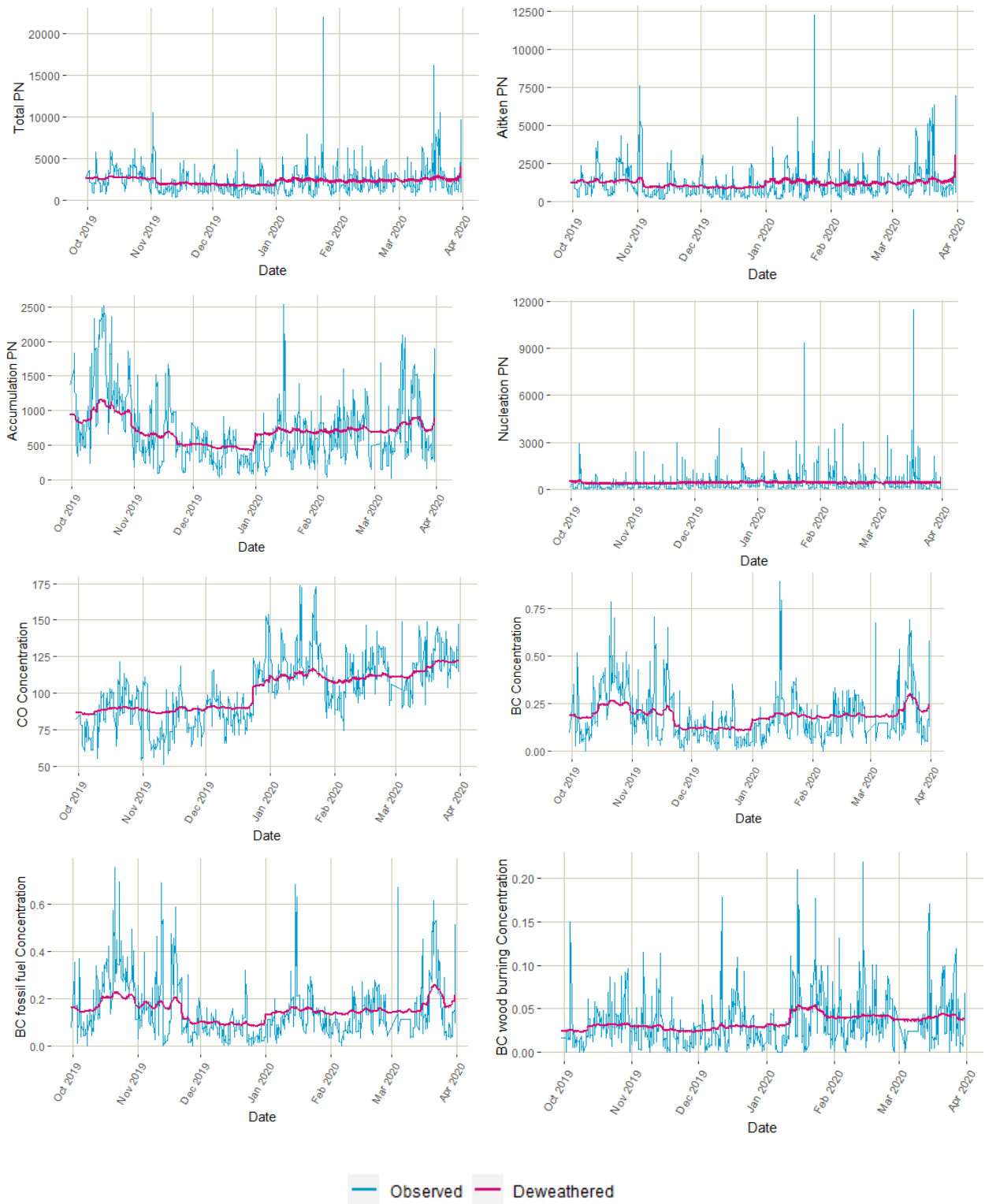


Fig. 4.2 : Observed and Deweathered values of all the target variables (1/10/19 – 1/4/20)

Table 4.2.: Comparison of observed and deweathered values (1/10/19 – 1/4/20)			
Target Value	Average Deweathered	Average Observed	Dew-Obs
Total (cm ⁻³)	2255	2203	52
Aitken (cm ⁻³)	1178	1114	64
Accumulation (cm ⁻³)	706	700	6
Nucleation (cm ⁻³)	412	388	24
CO (ppb)	101.4172435	101.1070489	0.310194598
BC (µg/m ³)	0.182718211	0.181461494	0.001256718
BCff (µg/m ³)	0.15005908	0.149857711	0.000201369
BCwb(µg/m ³)	0.033868827	0.031252399	0.002616428

The deweathered patterns that emerge are plotted separately to be thoroughly investigated (Fig. 4.3). They represent the ‘true’ values of Particle Number and pollutant concentrations, produced solely by emissions and chemical processes in the atmosphere (the weather effect is excluded through deweathering). We observe small fluctuations in all of the graphs, which correspond to weekly or even daily variations. Patterns that may be similar between the target values indicate common sources or sinks.

The Total PN pattern is governed mostly by the Aitken and Accumulation particles, as reported in previous studies [11, 20]. The Nucleation PN does not exhibit patterns as significant as the rest of the quantities, and it does not seem to contribute much to the Total PN compared to the Aitken and Accumulation concentrations. There is an increase in the Aitken particle number at the end of March 2020, and due to its non-recurring nature, this behavior could be attributed to new particle formation.

By investigating the deweathered quantities of CO and BC a common increase around the mid-end of March can be noted, followed by a smooth decrease in all cases. That pattern is also seen in the particle number concentration plots. It could be associated with the lockdown, as similar patterns regarding lockdowns in other locations have been reported. [9] The concentrations of these pollutants during a lockdown would be expected to decrease, due to less frequent use of vehicles. The fact that in Finokalia this decrease is not very sharp could be attributed to the area not being urban, but rather positioned a long way from the big cities of Crete.

A common pattern in both the pollutant concentrations and the particle number concentration graphs is a sharp increase in January of 2020. A possible explanation for that is the olive branch burning that takes place in the area after the olive harvest in November and December each year. Only the CO and BCwb concentrations differ in that aspect, as CO increases one week earlier, while BCwb increases one week after January 2020. This deviation cannot be explained, but it is not significant, as it may have to do with other

sources in the area, like heating and fireplace emissions.

Another universal characteristic that was observed was low values during November, which remained low until the occurrence of the sharp increase in January. In November and December every year there are great amounts of rain compared to the rest of the months, so the drop in concentrations could be associated with wet deposition. This scenario, however, is ambiguous as precipitation as a meteorological variable was used during the normalization step, so its effect should have been removed. Nevertheless, rainfalls in southern Greece during that period as a regional pattern may result to such a decrease, despite the rain occurrence at Finokalia or not.

Some of the values, namely the Accumulation mode, the CO, and the BC concentrations, display increases in October. This may stem from the fact that October comes after summer, which is a dry period in Crete with very little rainfall. Due to the lack of rain during that time, particles tend to build up ('accumulate') until they are scavenged or deposited, so that could explain the high values that emerge in October. Also, the absence of rain is conducive to the occurrence of transport phenomena, meaning that pollutants from large cities like Heraklion or Athens could reach Finokalia and contribute to the concentrations measured there. [20, 23] It seems, however, that the Random Forest model has limited capacity regarding the deciphering of transport which is strongly correlated with the air mass backward trajectories. It is not clear whether it is able to normalize the impact of transports, even though the back-trajectory cluster variable was included in the training process. This limitation has been reported in a previous study as well [9]. It may originate from the fact that the model does not have enough training instances to learn from, given the fact that the study period is relatively short (October to April, 7 months) and there are 12 clusters of back trajectories in total. This complication could be resolved if less clusters were used, or if a longer study period was investigated.

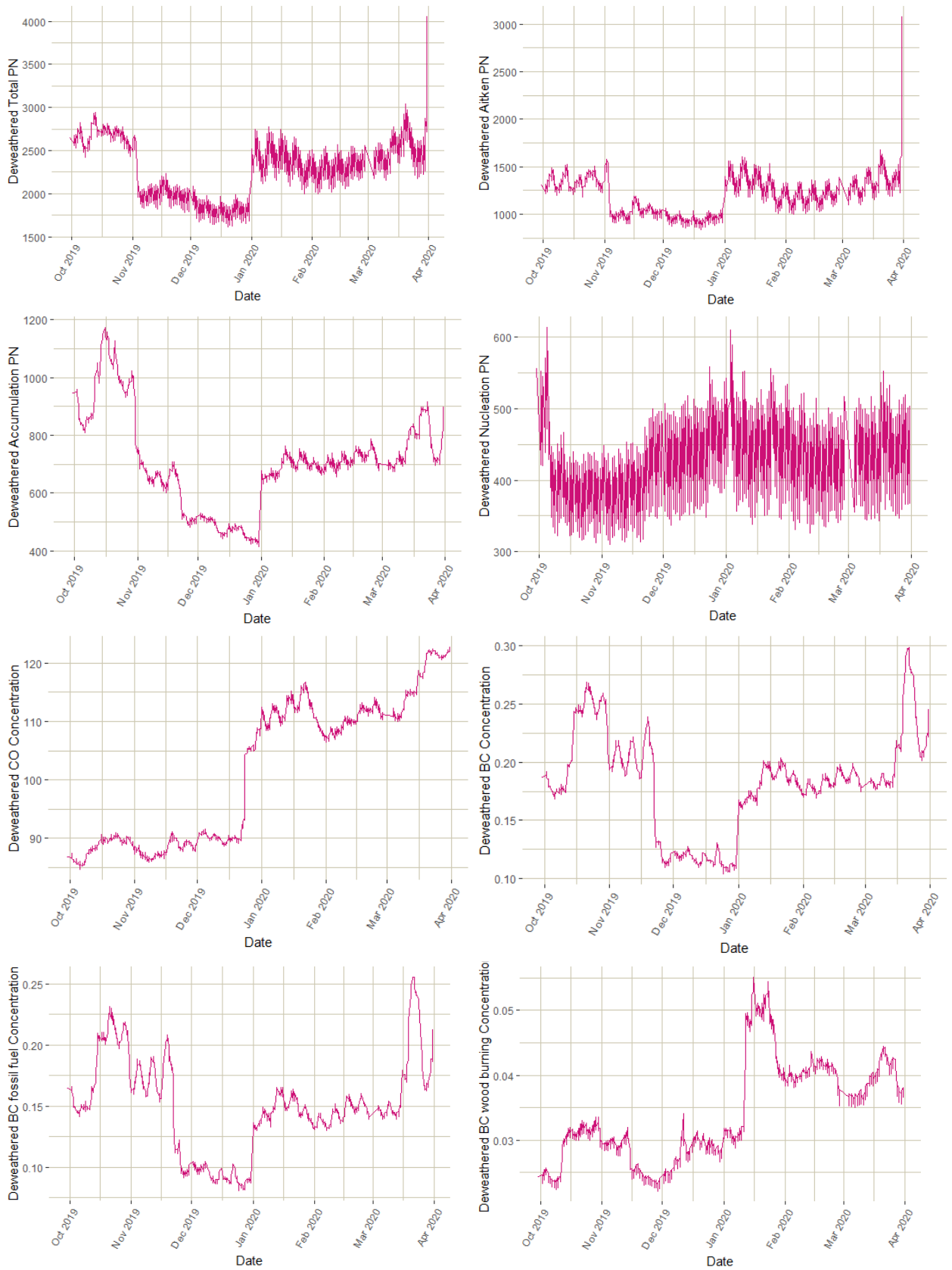


Fig. 4.3 : Deweathered values of all the target variables (1/10/19 - 1/4/20)

4.2. October 2018 – March 2019

The second run was for the dates 1/10/18 – 1/4/19. The metrics used were the same as in the previous run and they are presented in Table 4.3. There is not great difference when compared to the scores of the first run. The test instances (n variable) of the Particle Number are less in this case, but it does not affect the model's performance significantly. Again we observe the scale dependency of the RMSE, MB and MGE scores, and how their values change depending on range of the target values. The rest of the metrics remain fine. There is no increase or decrease compared to the previous section. Again, the True vs. Predicted plots are presented (Fig.4.4) to better justify the model's evaluation results. The majority of values is close to the $y=x$ line, indicating that the models are accurate to the same level as before.

Table 4.3 : Metrics used for the evaluation of the model, for the period 10/18 – 03/19										
Target Value	n	FAC2	MB	MGE	NMB	NMGE	RMSE	r	COE	IOA
Total (cm ⁻³)	261	0.854	4	947	0.001	0.365	1790	0.663	-0.173	0.413
Aitken (cm ⁻³)	261	0.808	19	529	0.015	0.423	1040	0.675	-0.265	0.367
Accumulation (cm ⁻³)	261	0.919	10	191	0.012	0.228	268	0.866	0.423	0.711
Nucleation (cm ⁻³)	261	0.325	-47	445	-0.095	0.880	1131	0.529	-0.303	0.348
CO (ppb)	313	1	0.643	9.9618	0.005	0.089	13.8	0.745	-0.023	0.488
BC (µg/m ³)	324	0.892	-0.007	0.0428	-0.033	0.213	0.0616	0.878	0.390	0.695
BCff (µg/m ³)	324	0.879	-0.007	0.0357	-0.04	0.216	0.0525	0.884	0.428	0.714
BCwb(µg/m ³)	324	0.756	-0.0006	0.0127	-0.017	0.366	0.0202	0.624	-0.197	0.401

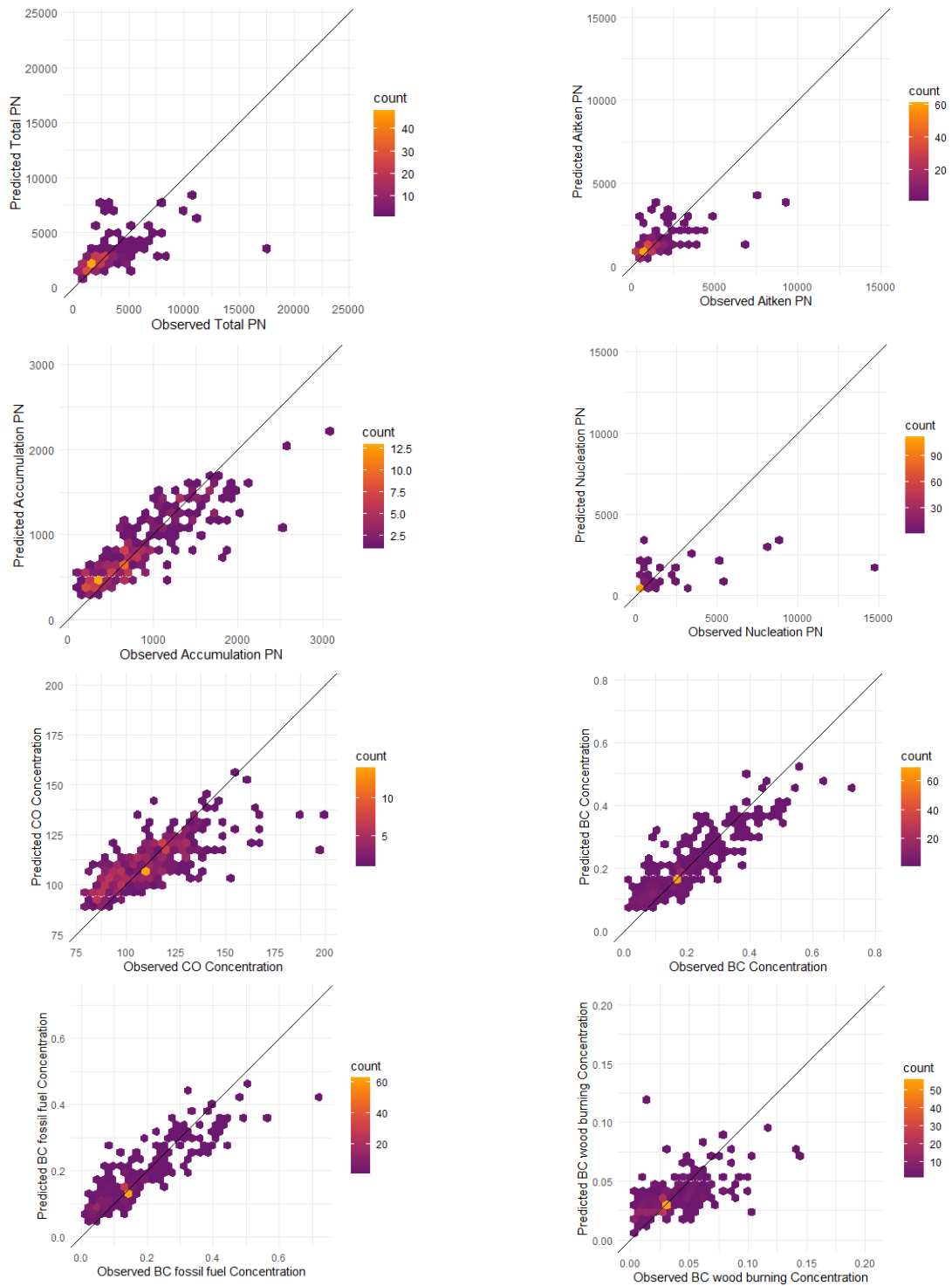


Fig. 4.4 : True vs. Predicted values (1/10/18 – 1/4/19)

The next step is to plot once again the observed and deweathered quantities to examine the effect of Meteorological Normalization (Fig. 4.5). These plots, unlike the ones that depict only deweathered quantities, can be used to detect missing values in the dataset. For example, in the BC measurements it can be seen from the plots that in the period January 2019 –

February 2019 the measurements are scarce. Again, there are plenty of times where there is great difference in magnitude between the observed and deweathered values. The average values are presented in Table 4.4.

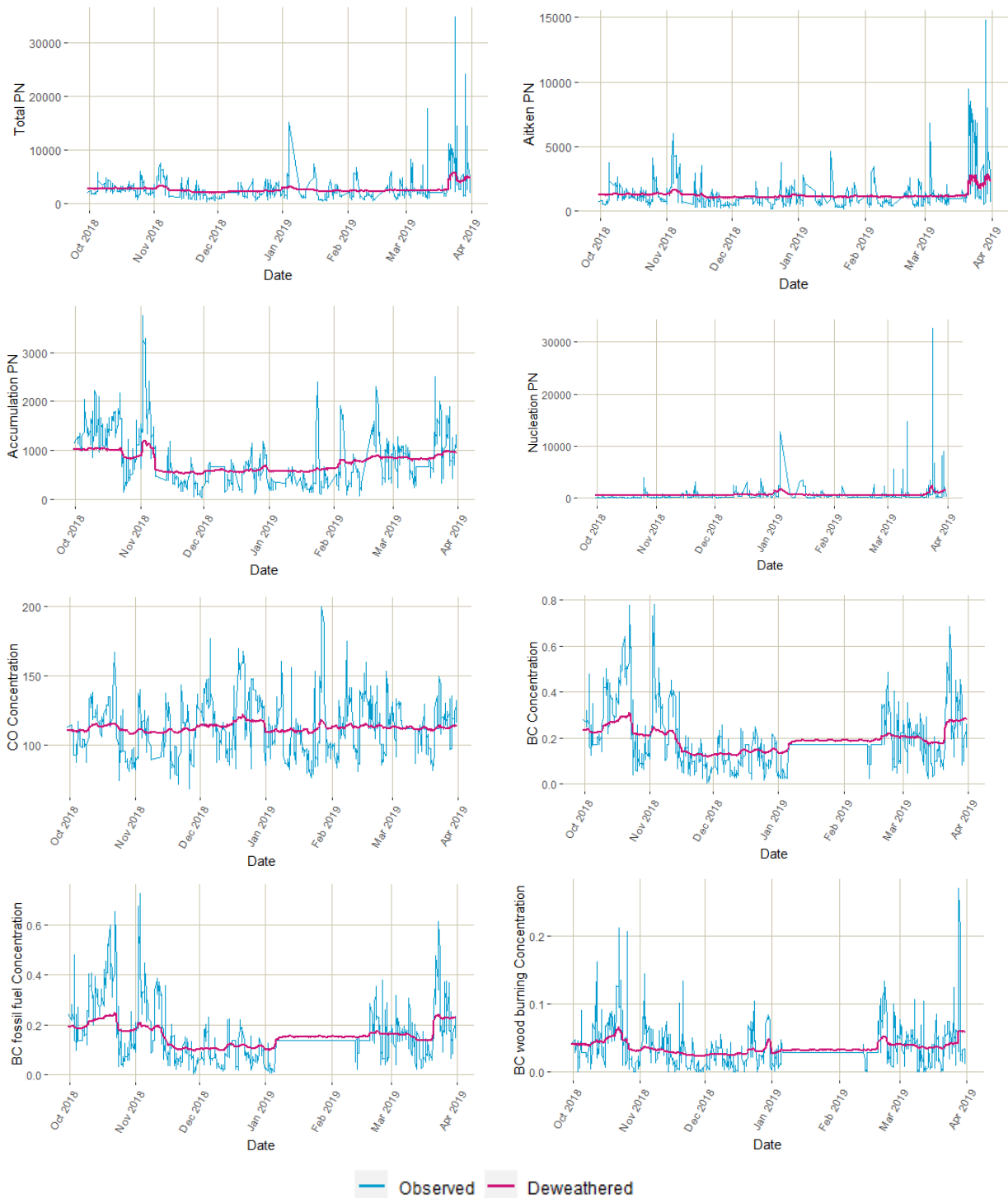


Fig. 4.5 : Observed and Deweathered values of all the target variables (1/10/18 – 1/4/19)

Table 4.4.: Comparison of observed and deweathered values (1/10/18 – 1/4/19)			
Target Value	Average Deweathered	Average Observed	Dew-Obs
Total (cm ⁻³)	2547	2489	58
Aitken (cm ⁻³)	1209	1183	26
Accumulation (cm ⁻³)	767	807	40
Nucleation (cm ⁻³)	567	478	89
CO (ppb)	112.6322891	112.7073428	0.075053742
BC (µg/m ³)	0.193048636	0.194802951	0.001754315
BCff (µg/m ³)	0.156932436	0.159559502	0.002627066
BCwb(µg/m ³)	0.034745937	0.033752477	0.00099346

In the deweathered plots that follow, we can observe the patterns and the impact of the Meteorological Normalization, and compare the plots with the year before. Like in the previous case, the daily and weekly variations can be observed and the Total particle number is governed mostly by the Aitken and the Accumulation modes. On the other hand, there is a peak in the Nucleation mode in January which is present only for a few days, something that was not observed in the year before. The particle number drops immediately and returns to the level before the peak. This event could be attributed to new particle formation. Also at the start of November there is a sudden increase in the Total Particle Number (associated with increases in Aitken and Accumulation, but not with Nucleation) and peaks around that period appear in the concentrations of CO and BC as well. As discussed in the previous section, this behavior could be attributed to particle accumulation due to lack of rain, right before the wet season which starts around November each year. Then during and after November, some values exhibit a constant decrease (in BC, Accumulation and Total particle number the drop is more visible), most likely due to precipitation.

At the end of March we observe once again an increase and then a smooth decrease, which in the previous section was identified as probable COVID-19 lockdown effects. However, in 2019 there was no lockdown so that conclusion is most likely false. It could be associated with dust events, but as explained above, it is unclear how the model handles these events.

In the period around January 2019, a lot of peaks are observed in all the values which is a different case compared to the previous section, where only one sharp increase was present in January 2020 for the majority of the values. Again, the source of these increases would be the olive branch burning after the harvest period. Harvesting may vary over the years, but it should occur in the months between November and February. The increase in BC from fossil fuels is particularly interesting, and it could perhaps result from region wide extended use of fossil fuels for household heating.

The problem of the model not being able to decouple the transport events remains, and it is not clear whether pollution events (i.e. peaks in concentrations) have local sources or not.

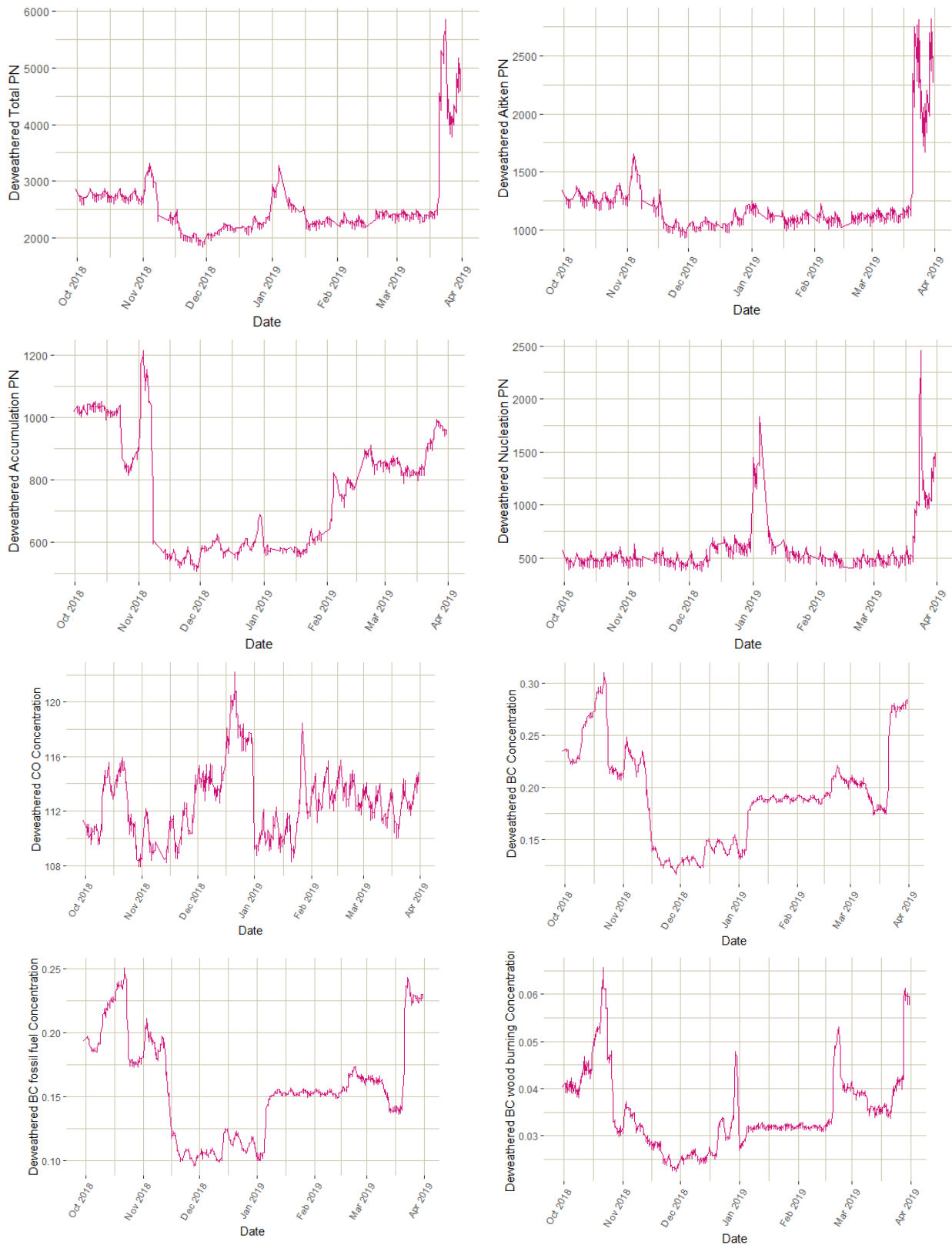


Fig. 4.6 : Deweathered values of all the target variables (1/10/18 - 1/4/19)

4.3. Evaluation of the IMO2020 regulation

The focus of this work so far was the periods October 2019 – March 2020 and October 2018 – March 2019, in order to study the deweathering effects in Finokalia, an approach that has not been done before for this area. A second aim is to evaluate the IMO2020 regulation, as Finokalia is located in a key position with regards to shipping pathways. In order to reach conclusions about whether the limitations were followed, we will investigate changes in the deweathered patterns of the pollutants. As mentioned above, in theory the Deweathering should extract values that are dependent only on emissions and chemical production.

By looking at the plots of the previous two sections, no notable decrease trend can be seen as of 1/1/20 in any of the target values. To better support this argument, we will compare directly the average deweathered values of the two study periods. As it can be noted from Table 4.5 concentrations in January 2019 – March 2019 are greater than January 2020 – March 2020. However, as it was analyzed in the previous sections, the sources of the decreases (or increases) are not completely clear. The model has difficulties decoupling transport which would normally affect the shipping emissions measured in the station. Also, it might take a few months for the imposed limitations to actually take effect, so when available, data from longer periods should be taken into account as well.

Table 4.5: Comparison of deweathered values of the two periods of interest			
Target Value	Average Deweathered (1/1/20-1/4/20)	Average Deweathered (1/1/19-1/4/19)	Dew ₂₀₂₀ - Dew ₂₀₁₉
Total PN	2357	2701	344
Aitken PN	1255	1261	6
Accumulation PN	724	772	48
Nucleation PN	423	655	232
CO	113	112	1
BC	0.193145194	0.198981881	0.005836687
BCff	0.155714228	0.160004023	0.004289795
BCwb	0.04027432	0.035653465	0.004620855

4.4. Conclusions and Future Perspectives

The purpose of this thesis was to apply the Machine Learning – based method of Deweathering to investigate the effects of meteorology on pollutant measurements conducted at Finokalia Monitoring station. The data used were meteorological variables collected at the station, as well as ERA5 reanalysis data and time variables. Random Forest models were built and each model corresponded to one of the following target values: Total Particle Number, Aitken, Accumulation and Nucleation Particle Number, Carbon Monoxide concentrations, Black Carbon Concentration, Black Carbon originating from fossil fuel and finally Black Carbon produced by biomass burning. The study periods were the 6 month period ranging from October to March, for the years 2019 and 2020. All the Random Forest models that were trained exhibited fine performance and were subsequently used to conduct the Meteorological Normalization. The plots that resulted from the deweathering of the target values would be used to reach conclusions regarding the emissions and probable sources of pollution in the area. However, some challenges emerged in explaining these deweathered plots and patterns. Biomass burning was confirmed as a source that contributed to the rise of pollutant concentrations but the models were found to have difficulties in deciphering the effect of the precipitation and backward trajectory cluster variable. As a result in the deweathered plots emerged patterns that were attributed to signals from mixed sources, for example biomass burning combined with transport events. Thus it was unclear to determine which increases and decreases stemmed from local emissions and which were the result of pollutants reaching the station from neighboring places with more dense population.

Many results presented in this work carry great uncertainty, since this method is used for the first time in Finokalia. Most of the previous studies used plain pollutant concentrations as the deweathering targets, not size distribution data, as we did. Further improvements can be made, especially with regards to the back trajectory cluster variable and how it is interpreted by the model. Polar plots of wind speed, wind direction and concentrations could be useful to decouple air mass effects on deweathered local pollution. Also, the partial dependencies information that Random Forest models provide could be used to explore the different correlations of the features used. Finally, there are two more focus areas that would complement our research: the use of other study periods, e.g. the summer months of the last 3 years, and the selection of more compounds as the target values, like CO₂, CH₄ or O₃.

As a conclusion, it can be recognized that once the uncertainties regarding arguments made in this study are dealt with, the normalization step will bear more straightforward results and then the true potential of the

Deweathering method will be uncovered.

5. References

1. Carslaw, D. C. & Taylor, P. J. Analysis of air pollution data at a mixed source location using boosted regression trees. *Atmos. Environ.* **43**, 3563–3570 (2009)
2. Carslaw, D. C. & Ropkins, K. openair – An R package for air quality data analysis. *Environ. Model. Softw.* **27–28**, 52–61 (2012)
3. Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. (O'Reilly Media, 2017)
4. Grange, S. K. & Carslaw, D. C. Using meteorological normalisation to detect interventions in air quality time series. *Sci. Total Environ.* **653**, 578–588 (2019)
5. Grange, S. K., Carslaw, D. C., Lewis, A. C., Boleti, E. & Hueglin, C. Random forest meteorological normalisation models for Swiss PM₁₀ trend analysis. *Atmospheric Chem. Phys.* **18**, 6223–6239 (2018)
6. Mallet, M. D. Meteorological normalisation of PM10 using machine learning reveals distinct increases of nearby source emissions in the Australian mining town of Moranbah. *Atmospheric Pollut. Res.* **12**, 23–35 (2021)
7. Pang, Z., Feng, F. & O'Neill, Z. Investigation of the Impacts of COVID-19 on the Electricity Consumption of a University Dormitory Using Weather Normalization (2020)
8. Vu, T. V. *et al.* Assessing the impact of clean air action on air quality trends in Beijing using a machine learning technique. *Atmospheric Chem. Phys.* **19**, 11303–11314 (2019)

9. Shi, Z. *et al.* Abrupt but smaller than expected changes in surface air quality attributable to COVID-19 lockdowns. *Sci. Adv.* **7**(2021)
10. Breiman, L. Random Forests. *Mach. Learn.* **45**, 5–32 (2001)
11. Kalivitis, N. *et al.* Particle size distributions in the Eastern Mediterranean troposphere. *Atmos Chem Phys* **10** (2008)
12. Draxler, R. & Hess, G. An overview of the HYSPLIT_4 modeling system for trajectories, dispersion, and deposition. *Aust. Meteorol. Mag.* **47**, 295–308 (1998)
13. Draxler, R. & Hess, G. Description of the HYSPLIT_4 modeling system, NOAA Air Resources Laboratory, Silver Spring, MD, 24 pp (1997)
14. Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., Thépaut, J-N.: ERA5 hourly data on single levels from 1979 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS) (2018)
15. Seinfeld and Pandis, *Atmospheric Chemistry and Physics, From air pollution to Global Change*, John Wiley & Sons, 1998
16. Whitby, K. T. The physical characteristics of sulfur aerosols. *Atmospheric Environ.* **1967** **12**, 135–159 (1978)
17. World Health Organisation, *Ambient Air Pollution* [Online], Available from: <https://www.who.int/teams/environment-climate-change-and-health/air-quality-and-health/ambient-air-pollution> [Accessed 25th May 2021]
18. Schraufnagel, D. E. *et al.* Air Pollution and Noncommunicable Diseases. *Chest* **155**, 417–426 (2019)

19. International Maritime Organization, IMO2020: Cleaner Shipping for Cleaner Air, Available from: <https://imo.org/en/MediaCentre/PressBriefings/Pages/34-IMO-2020-sulphur-limit-.aspx> [accessed 8th June 2021]
20. Kalivitis N. Φυσικές Ιδιότητες Αιωρούμενων Σωματιδίων με κλιματικό ενδιαφέρον στην ατμόσφαιρα της Ανατολικής Μεσογείου, PhD Thesis, University of Crete (2008)
21. S. Kang, Y. Zhang, Y. Qian, and H. Wang, "A review of black carbon in snow and ice and its impact on the cryosphere," *Earth-Sci. Rev.*, (2020)
22. Kalivitis, N., Kerminen, V.-M., Kouvarakis, G., Stavroulas, I., Tzitzikalaki, E., Kalkavouras, P., Daskalakis, N., Myriokefalitakis, S., Bougiatioti, A., Manninen, H.E., Roldin, P., Petäjä, T., Boy, M., Kulmala, M., Kanakidou, M., Mihalopoulos, N., 2019. Formation and growth of atmospheric nanoparticles in the eastern Mediterranean: results from long-term measurements and process simulations. *Atmospheric Chem. Phys.* 19, 2671–2686 (2019)
23. Solomos, S., Kalivitis, N., Mihalopoulos, N., Amiridis, V., Kouvarakis, G., Gkikas, A., Biniotoglou, I., Tsekeri, A., Kazadzis, S., Kottas, M., Pradhan, Y., Proestakis, E., Nastos, P.T., Marenco, F. From Tropospheric Folding to Khamsin and Foehn Winds: How Atmospheric Dynamics Advanced a Record-Breaking Dust Episode in Crete. *Atmosphere* 9, 240.(2018)

Appendix A: Codes and Methods

A1. Trajectory Clustering

```
1  startdata=read.csv("Finokalia_traj.csv",stringsAsFactors=FALSE)
2  startdata$date=as.POSIXct(startdata$date)
3  hours=strftime(startdata$date, format="%H")
4  class(hours)
5  hours=as.numeric(hours)
6  end=cbind(startdata, hours)
7  #per 4 wres
8  end1=filter(end, hours==4 | hours==8 | hours==12 | hours==16 |
9  hours==20 | hours== 0)
10 str(end1)
11 write.csv(end1, "Fin_trajper4.csv")
12 #memory error
13 end2=filter(end, hours==0 | hours==2 | hours==4 | hours==6 | hours==8
14 | hours==10 | hours==12 | hours==14 | hours==16 | hours==18 | hours==20 |
15 hours==22 | hours== 0)
16 str(end2)
17 write.csv(end2, "Fin_trajper2.csv")
18 df=read.csv("Fin_trajper4.csv", stringsAsFactors=FALSE)
19 getwd()
20 dfready=df[4:19]
21 str(dfready)
22 options() #check default variables for jupyter
23 options(jupyter.plot_scale=1) #change default variables
24 options(repr.plot.width = 7, repr.plot.height =4, repr.plot.res =
25 120, repr.plot.pointsize=12)
26 library(openair)
27 library(lubridate)
28 library(latticeExtra)
29 library(ggplot2)
30 require(devtools)
31 #install_github('davidcarslaw/worldmet')
32 library(worldmet) ## download_met_data
33 library(mapdata)
34 #setwd("C:/Users/user/Documents/sxoli/ptyxiaki/trajclustering")
35 dfready$date<-as.POSIXct(dfready$date, format="%Y-%m-%d
36 %H:%M:%S", tz="GMT")
37 ficluster=trajCluster(dfready,
38 method="Euclid", n.cluster=12, lwd=2, npoints=NA)
39 cdata=ficluster$data
40 #trajPlot(selectByDate(dfready, start = "15/4/2015", end =
41 "16/4/2015"), map.cols = openColours("hue", 10), col = "grey30")
42 write.csv(cdata, "Finokalia_trajper4clustered12.csv")
43 #for hourly clusters
44 library(dplyr)
45 dfc=read.csv('Finokalia_trajper4clustered12.csv')
46 dfc[!duplicated(dfc$cluster), ]
47 hourly=write.csv(dfc[!duplicated(dfc$date), ], "per4hour.csv")
48 trajPlot(selectByDate(dfready, start = "15/4/2015", end =
49 "16/4/2015"), map.cols = openColours("hue", 10), col =
50 "grey30", origin=FALSE)
```


A2. Final Preprocessing

```
1 print(Sys.time())
2 Sys.timezone()
3 df=read.csv('Finokalia_groundinter.csv',header=TRUE,stringsAsFactors=FALSE)
4 df$date=as.POSIXct(df$date, format="%m/%d/%Y %H:%M")
5 str(df)
6 library(lubridate)
7 df$date=df$date +hours(3)
8 str(df)
9 write.csv(df, 'Finokalia_local.csv')
10 ### Time variables : date_unix,week,weekday,hour,month,day_julian
11 date_unix=as.numeric(df$date)
12 df1=cbind(df,date_unix)
13 str(df1)
14 week=strftime(df$date, format = "%V")
15 df2=cbind(df1,week)
16 dateonly=as.Date(df$date)
17 weekday=as.numeric(strftime(dateonly, "%u"))
18 head(weekday)
19 df3=cbind(df2,weekday)
20 #monday=1,sunday=7
21 df3
22 hour=strftime(df$date, format="%H")
23 df4=cbind(df3,hour)
24 month=strftime(df$date,format="%m")
25 df5=cbind(df4,month)
26 day_julian=yday(dateonly)
27 df6=cbind(df5,day_julian)
28 head(df6)
29 write.csv(df6, 'Finokalia_readyfordw.csv')
```

```
1 import pandas as pd
2 df=pd.read_csv('Finokalia_full.csv',header=0,index_col='date')
3 df.head()
4 ground=df[['temp','RH','ws','wd']]
5 df1=df[['cluster','blh','tp','ssr','sp','tcc','Ntotal','Nnucleation','Naitken','Naccumulation']]
6 df2=ground.interpolate(method='linear')
7 final=pd.concat([df2,df1],axis=1)
8 final.isna().sum()
9 final.to_csv('Finokalia_groundinter.csv')
```

A3. Meteorological Normalization

```
1 rm(list=ls(all=TRUE))
2 library(openair)
3 library(plyr)
4 library(dplyr)
5 library(rmweather)
6 library(ranger)
7 library(readxl)
8 library(randomForest)
9 library(knitr)
10 library(ggplot2)
11 #please install the required (above) packages first
```

```

12 setwd("C:/Users/User/Documents/sxoli/ptyxiaki/DeweatherStuff/Fin
okalia")
13 filename="Finokalia_cl_groundinterpolated_ERA_LOCALTIME_timevar_
co" #model inputs file
14 polllist<-list("co") #run each pollutant one by one
15 ncal=1 #ncal: modeling using different seeds and select a model
with highest model performance
16 #1/3/2015 4:00
17 Dataraw1<-import(paste(filename, ".csv", sep=""), date="date",
date.format = "%Y-%m-%d %H:%M")
18 #Dataraw1: all dataset
19 setwd("C:/Users/User/Documents/sxoli/ptyxiaki/DeweatherStuff/Fin
okalia/1_10_19-1_4_20")
20 Dataraw1$cluster<-as.factor(Dataraw1$cluster) #set back
trajectory as category
21 Dataraw1$weekday<-as.factor(Dataraw1$weekday) #set weekday as
category
22 Dataraw1 <- Dataraw1 %>% filter(!is.na(cluster))
23 Dataraw <- Dataraw1 %>% filter(date>="2019-10-01"& date <=
"2020-04-01") #Dataraw: selected dataset for model training and
weather normalisation
24 for (poll in polllist){
25 r.min <- 0.1
26 perform<-matrix(data=NA, ncol=11, nrow=1)
27 colnames(perform)<-c("default", "n", "FAC2", "MB", "MGE", "NMB",
"NMGE", "RMSE", "r", "COE", "IOA")
28 for (i in as.numeric(1:ncal)){
29 set.seed(i)
30 data_prepared <- Dataraw %>%
31 filter(!is.na(ws)) %>%
32 dplyr::rename(value = poll) %>%
33 rmw_prepare_data(na.rm = FALSE, fraction = 0.7)
34 set.seed(i)
35 RF_model <- rmw_do_all(
36 data_prepared,
37 variables = c(
38 "date_unix", "day_julian", "weekday", "hour", "temp", "rh",
"wd", "ws", "sp", "cluster", "tp", "blh", "tcc", "ssr"), #factors for
random forest modeling
39 variables_sample=c("temp", "rh", "wd",
"ws", "sp", "cluster", "tp", "blh", "tcc", "ssr"), #factors for
weather replacement
40 n_trees = 300,
41 min_node_size = 3, n_samples = 1000,
42 verbose = TRUE
43 )
44 str(data_prepared)
45 testing_model <- rmw_predict_the_test_set(model =
RF_model$model, df = RF_model$observations)
46 model_performance<-modStats(testing_model, mod = "value", obs =
"value_predict",
47 statistic = c("n", "FAC2", "MB", "MGE", "NMB", "NMGE", "RMSE",
"r", "COE", "IOA"),
48 type = "default",
rank.name = NULL)
49 perform<-rbind(perform, model_performance)
50 if (model_performance$r > r.min){
51 r.min <- model_performance$r
52 RF_modelo <- RF_model}
53 }
54 save.image(file= paste(filename, "_", poll, "_RW", ".RData", sep=""))

```

```

55 write.table(perform,
file=paste(filename, "_", poll, "_RWPerformance", ".csv", sep=""),
sep=";", row.names=FALSE)
56 }
57 }
58 setwd("C:/Users/User/Documents/sxoli/ptyxiaki/DeweatherStuff/Fin
okalia/")
59 filename="Finokalia_cl_groundinterpolated_ERA_LOCALTIME_timevar_
co"
60 filenamelist<-
list('Finokalia_cl_groundinterpolated_ERA_LOCALTIME_timevar_co')
61 setwd("C:/Users/User/Documents/sxoli/ptyxiaki/DeweatherStuff/Fin
okalia/1_10_19-1_4_20")
62 for (filename in filenamelist){
63 polllist<-list('co')
64 for (poll in polllist){
65 a=paste(filename, "_", poll, "_RW.RData", sep='')
66 load(a)
67 testing_model <- rmw_predict_the_test_set(model =
RF_modelo$model, df = RF_modelo$observations)
68 print(class(testing_model))
69 model_performance<-modStats(testing_model, mod = "value", obs =
"value_predict",
70 statistic = c("n", "FAC2", "MB", "MGE", "NMB", "NMGE", "RMSE",
"r", "COE", "IOA"),
71 type = "default",
rank.name = NULL)
72 normli<-cbind(RF_modelo$normalised, RF_modelo$observations$value)
73 write.table(testing_model,
file=paste(filename, "_", poll, "_testing_model.csv", sep=""),
sep=";", row.names=FALSE)
74 write.table(normli,
file=paste(filename, "_", poll, "_normalised.csv", sep=""), sep=";",
row.names=FALSE)
75 }
76 }
77 # some scores
78 scores=rmw_model_statistics(RF_modelo$model)
79 scores$r_squared
80
81 # ----PLOTS----
82 date = as.Date(RF_modelo$normalised$date, format = "%Y-%m-%d
%H:%M")
83 value_predict=RF_modelo$normalised$value_predict
84 value=RF_modelo$observations$value
85 df=cbind(as.data.frame(date), value, value_predict)
86 # both
87 ggplot(df, aes(x = date)) +
88   geom_line(aes(y = value, color = "value"), size = 0.7) +
89   geom_line(aes(y = value_predict, color = "value_predict"),
size = 0.8) +
90   labs(x = "Date",
91        y = "CO Concentration",
92        color = "Legend") +
93   scale_color_manual(labels = c("Observed", "Deweathered"),
values = c("deepskyblue3", "deeppink3")) +
94   theme(legend.title=element_blank())+
95   scale_x_date(date_labels = "%b %Y", date_breaks = '1 month')+
96   theme(axis.text.x=element_text(angle=60, hjust=1))+
97   theme(legend.position="bottom")+
98   theme(panel.grid.minor = element_line(colour = "white", size =

```

```

0.5)))+
99   theme(panel.grid.major = element_line(colour = "cornsilk3",
size = 0.5),panel.background = element_rect(fill = "white",
colour="white"))
100 #normalised values
101 ggplot()+
102   geom_line(data =df, aes(x= date,
y=value_predict), color='deeppink3', size=0.6)+
103   theme(axis.text.x=element_text(angle=60, hjust=1))+
104   theme(panel.grid.minor = element_line(colour = "grey", size =
0.5))+
105   theme(panel.grid.major = element_line(colour = "grey", size =
0.5),panel.background = element_rect(fill = "white",
colour="white")) +
106   theme(legend.title=element_blank()+
107     labs(x = "Date",
108           y = "Deweathered CO Concentration")+
109     scale_x_date(date_labels = "%b %Y",date_breaks = '1 month')+
110     theme(axis.text.x=element_text(angle=60, hjust=1))+
111     theme(panel.grid.minor = element_line(colour = "cornsilk3",
size = 0.5))+
112     theme(panel.grid.major = element_line(colour = "cornsilk3",
size = 0.5),panel.background = element_rect(fill = "white",
colour="white"))
113
114 # TRUE PRED
115 rmw_plot_test_prediction <- function(df, bins = 30, coord_equal
= TRUE) {
116
117   # Plot
118   plot <- ggplot2::ggplot(df, ggplot2::aes(value,
value_predict)) +
119     ggplot2::geom_hex(bins = bins) +
120     ggplot2::geom_abline(slope = 1, intercept = 0) +
121     ggplot2::theme_minimal() +
122     viridis::scale_fill_viridis(
123       option = "inferno",
124       begin = 0.3,
125       end = 0.8
126     ) +
127     ggplot2::xlab("Observed CO Concentration") +
128     ggplot2::ylab("Predicted CO Concentration")
129   # Fix axes
130   if (coord_equal) {
131
132     # Get axes limits
133     min_values <- min(c(df$value, df$value_predict), na.rm =
TRUE)
134     max_values <- max(c(df$value, df$value_predict), na.rm =
TRUE)
135
136     plot <- plot +
137       ggplot2::ylim(min_values, max_values) +
138       ggplot2::xlim(min_values, max_values) +
139       ggplot2::coord_equal()
140
141   }
142   return(plot)
143 }
144 rmw_predict_the_test_set(
145   model = RF_modelo$model,

```

```
146 df = RF_modelo$observations
147 ) %>%
148 rmw_plot_test_prediction()
```