Universiteit
Leiden
The Netherlands

Machine learning-based NO2 estimation from seagoing ships using TROPOMI/S5P satellite data

Kurchaba, S.

Citation
Kurchaba, S. (2024, June 11). *Machine learning-based NO2 estimation from seagoing ships using TROPOMI/S5P satellite data*. Retrieved from https://hdl.handle.net/1887/3762166

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/3762166

Note: To cite this publication please use the final published version (if applicable).

Chapter 2

General workflow

The aim of this study is to develop a methodology enabling the analysis of TROPOMI satellite data for the task of ship emission monitoring. In each chapter of the thesis, we address different aspects of this task. However, there is a sequence of steps that will be performed repeatedly in each chapter. Those steps are the integration of several data sources, feature engineering, selection of a machine learning model and optimization of its hyperparameters, the application to a given problem, and comparison of the estimated values of NO_2 with the theoretical measure of ship emission potential. We call it the general workflow (c.f. Figure 2.1). Each step of this general workflow will be introduced to the reader in this Chapter. The order, technical details, or methodology applied in each step will depend on the problem at hand and will be described in each chapter separately.

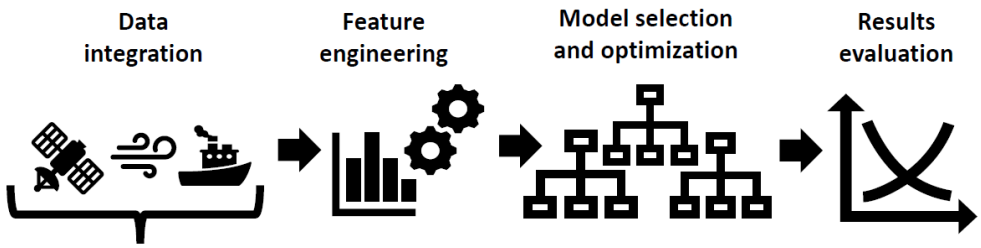

Figure 2.1: Visualization of general workflow.

2.1 Data integration

The first step of the general workflow is data integration. Within this thesis, we understand the process of data integration as follows:

Definition 2.1. Data integration is the process of combining data from several sources into one unified dataset in a way that enables the solution of a particular task at hand.

The main data source of this study is the set of NO_2 measurements coming from the TROPOMI instrument. These data contain among others information of our interest - the amount of NO_2 produced as a result of NO_x emission of individual seagoing ships. However, to enrich the TROPOMI data for our analyses, additional data sources are needed. In this study, we integrate the following data sources: 1) the TROPOMI NO_2 product, 2) data on ship positions and some properties of the ship, and 3) wind information. In the following subsections, we introduce the sources of data used in the study.

2.1.1 TROPOMI data

The Sentinel-5 Precursor (Sentinel-5P) satellite was launched in October 2017 and started its operational phase in April 2018. TROPOMI is a spectrometer on board the Sentinel-5P satellite mission – a sun-synchronous satellite with a local equatorial overpass time at 13:30. The instrument measures the Top Of the Atmosphere (TOA) solar radiation reflected by and radiated from the Earth covering ultra-violet up to the part of the visible spectrum (270-500 nm), near-infrared (675-775 nm), shortwave infrared (2305–2385 nm) spectral bands. The maximal ground pixel resolution of the instrument reaches 3.5×5.5 km^2 at the nadir, while the actual size of the pixel will vary depending on the true distance between the satellite and the captured part of the Earth's surface. We use Level 2 tropospheric NO_2 column data, publicly available via https://dataspace.copernicus.eu/ (previously https://s5phub.copernicus.eu/). In Chapter 3, the analysis is based on the TROPOMI data version 2.4.0. In Chapters 4, 5, the used data version is 1.3.0, and in Chapter 6 the study is conducted using TROPOMI data version 2.3.1.

The retrieval of NO_2 columns is performed using a 3-step procedure described in the Algorithm Theoretical Baseline Document [101]. A visual description of the process is presented in Figure 2.2. As a first step, NO_2 slant column densities are defined as the integrated amount of NO_2 along the average photon path from the Sun through the atmosphere back to the sensor [11, 41]. Next, based on the output from the

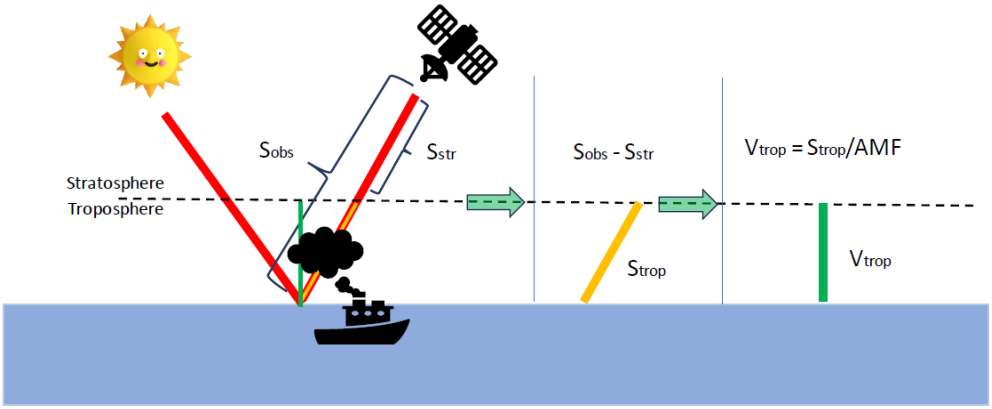

Figure 2.2: Illustration of the retrieval algorithm of NO_2 vertical tropospheric column density. S stands for Slant Column Density, V – Vertical Column Density. Visualization inspired by [105].

data assimilation system, the slant column is split into stratospheric and tropospheric components [29]. As the last step, using a tropospheric air mass factor (AMF), the tropospheric slant columns are transformed into tropospheric vertical column densities. The AMF accounts for the path length that sunlight travels through the atmosphere before reaching the satellite sensor, normalizing it by the amount of sunlight that would reach the surface under direct overhead conditions. Calculation of AMF to a large extent depends on the emission inventories and chemical transport models, which, in turn, rely on information about historical concentrations of emissions, including NO_2 [31]. Starting from Chapter 4, in all chapters of this thesis, we will base our analysis on tropospheric vertical column densities, as this variable ensures the best enhancement of NO_2 plumes. In Chapter 3, however, we study the sensitivity of the TROPOMI data-based detection system with respect to the NO_2 ship plume detection. Therefore, the historical information contained in AMF, and, in the resulting vertical column densities may cause information leakage. To prevent such a situation, the study presented in Chapter 3 will be based on tropospheric slant column density data.

2.1.2 Ship-related data

The second data source used in this study is information on ship positions. To coincide the detected NO_2 plumes with the emitting ships, the information on the positions of the ships at the moment of the satellite overpass is compulsory for this study. The used data on the positions of the ships comes from the Automatic Identification System

(AIS) transponders. Since 2002, all commercial sea-going vessels have the obligation to carry on board an AIS transponder [76] which transmits information about the position, speed, heading (direction), a unique identifier (MMSI), and the type of the ship.

At the moment, there is no open-access AIS with the spatiotemporal coverage and data quality required for this study. The data, however, can be accessed through several commercial providers. For the scope of this study, the AIS data as well as information about the dimensions of ships were provided by ILT, which has access to commercial databases.

In Figure 2.3, we present an example of TROPOMI data with the indicated positions of ships in the area starting from 2 hours before, until the moment of the satellite overpass. We can see that while the beginning of a ship's trajectory often corresponds with the origin of the plume, some significant deviations can be observed for the rest of the trajectory line. This happens because after the plume has been emitted by the ship, it is carried away in the direction of the prevailing winds. Therefore, for the efficient allocation of the plume with the ship emitter, information about the speed and the direction of the wind is required.

2.1.3 Wind data

Throughout the thesis, we use wind data from the European Center for Medium range Weather Forecasts (ECMWF). The wind fields (wind speed and wind direction) are the results of operational model analyses at a spatial resolution of $0.25°$[1], the temporal resolution of 6 hours and altitude of 10 meters. In [41], the wind data at 10 meters altitude was considered sufficient for ship-plume matching. Starting from the TROPOMI product version upgrade from 1.2.2 to 1.3.0 on March 27, 2019, the ECMWF 10-meter wind data for coinciding time is available as a support product in the TROPOMI data file [101].

2.2 Feature engineering

After data integration, the second step of the general workflow is feature engineering. We define feature engineering as follows:

Definition 2.2. Feature engineering is the process of extracting features from raw data.

[1]For the analyzed area the spatial resolution of $0.25° \times 0.25°$ translates to $\approx 23.4 \times 27.6$ km^2.

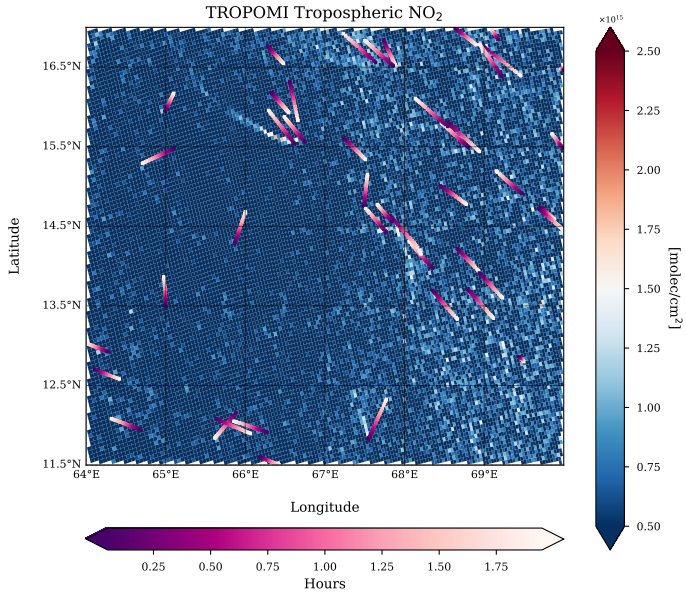

Figure 2.3: The NO_2 vertical tropospheric column density. Date: April 11th, 2020. Region: the Arabian Sea. Magenta lines indicate ship tracks based on information from AIS data. On the right-hand side of the map, an outflow effect from the variety of land bases NO_2 sources can be noticed.

Definition 2.3. Features are the set of characteristics associated with the data [75]. Other names of features are variables or attributes.

The aim of feature engineering is to transform the integrated dataset to be used by a machine-learning method for a particular task. Since in each chapter, we address different tasks, the applied methods of feature engineering will differ as well. The examples of feature engineering techniques that were used throughout the thesis include an assignation and further geometric transformation of the Region of Interest, data aggregation (through calculation of various statistics), encoding of spatial information, and categorical data encoding. They will be explained in the respective chapters.

2.3 Model selection and optimization

The next step of the workflow is the selection of a machine-learning model suitable for a given problem. This process includes the selection of the best-performing algorithm and the optimization of its hyperparameters. We define hyperparameters as follows:

Definition 2.4. Hyperparameters of a machine-learning algorithm are the parameters that steer the behavior of the learning process. The hyperparameters cannot be learned by the algorithm from its experience E (c.f. Definition 1.1) and need to be set by a researcher [55].

In the next subsections, we explain how we perform model selection and optimization in this thesis.

2.3.1 Machine-learning metrics

When performing model selection, we define beforehand evaluation metrics suited to the problem at hand. Several machine-learning metrics are used in this thesis and are defined below depending on whether we perform a binary classification or a regression task.

Binary classification metrics

In the context of binary classification, each classification result is assigned to one of the four categories:

- True positives (TP): the output data points with a positive class label correctly identified by the classifier.

- True negatives (TN): the output data points with a negative class label correctly identified by the classifier.
- False positives (FP): the output data points incorrectly identified by the classifier as positive.
- False negatives (FN): the output data points incorrectly identified by the classifier as negative.

The assignment to one category depends on the output of the model, that is the computed probability and a probability threshold. For instance, if the model gives a probability of 0.78 for a given data point and the threshold is set to 0.5, the data point will be labeled as positive. If the original label of the data point was positive, it is then correctly classified and is considered as TP. Using these categories, we can define performance metrics for a binary classifier.

First, we introduce a *precision-recall curve* – a graphical evaluation technique depicting precision as a function of recall where:

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN}, \tag{2.2}$$

Such a curve is obtained by using multiple probability thresholds to obtain multiple precision and recall points. When comparing the precision-recall curves of two models, if the precision and recall points of one curve are all above the points of the other, then the corresponding model is considered better than the other.

In the case of intersecting curves, to rank the models, we calculate an area under the precision-recall curve. We call this metric *average precision*. For a classifier that classifies all the data points correctly, the value of the *average precision* will be equal to 1. For a random guess classifier, the value of the *average precision* is equal to the ratio of positive samples in the dataset. Throughout the thesis, the average precision will be used as an evaluation metric for performing model selection and hyperparameter optimization.

The next method that we use in this thesis for the evaluation of the binary classifier is the *Receiver Operating Characteristic (ROC) curve* [35, 36]. The *ROC curve* is a graphical evaluation technique depicting all possible thresholds between the true positive rate (TPR), which is another name for *Recall*, and the false positive rate (FPR) [109], which we define as follows:

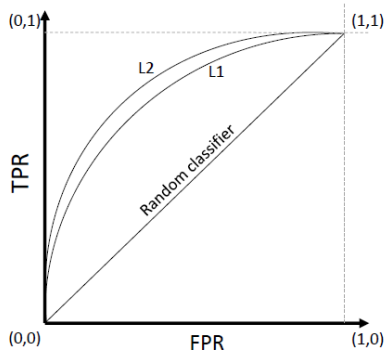

Figure 2.4: ROC curve - schematic example

$$TPR = \frac{TP}{TP + FN}, \tag{2.3}$$

$$FPR = \frac{FP}{FP + TN}, \tag{2.4}$$

An example of the *ROC curve* is presented in Figure 2.4. The classification results are perfect when $TPR = 1$ and $FPR = 0$. The classification results are completely wrong when $TPR = 0$ and $FPR = 1$. A diagonal line from the bottom left to the top right ($TPR = FPR$) corresponds to the results of a random-guess classifier.

Based on the *ROC curve*, we can compute the *Area Under the ROC Curve (ROC-AUC)* metric. The highest achievable score of *ROC-AUC* is equal to 1. In the case of random guessing, the *ROC-AUC* score will be equal to 0.5.

Regression model

For the evaluation of regression model performances, we use two metrics: *Pearson correlation coefficient* and *coefficient of determination* R^2. *Pearson correlation coefficient* ρ is defined as:

$$\rho = \frac{Cov(Y, \hat{Y})}{\sigma(Y)\sigma(\hat{Y})} \tag{2.5}$$

where Cov is the covariance, $\sigma(Y)$, and $\sigma(\hat{Y})$ are standard deviations of real and predicted values of a target variable respectively. The value $\rho = 1$ indicates a perfect linear correlation, value $\rho = -1$ indicates perfect linear anti-correlation and $\rho = 0$ is

the total absence of linear correlation. The second metric, R^2, is defined as:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{2.6}$$

$R^2 \in [0; 1]$, and is a measure of the goodness of fit of a model and is interpreted as the part of the variation of the predicted variable that is explained by the regression model. The $R^2 = 1$ suggests that the predictions obtained with a regression model fit the data perfectly well. We use R^2 as the quality metric for the process of model selection and optimization.

2.3.2 Hyperparameter optimization strategies

There are two strategies of algorithm selection and optimization of its hyperparameters used in this thesis. The first is a selection of the optimal algorithm among the list of pre-selected candidates while performing a randomized search [10] for the hyperparameters optimization. The benefit of this strategy is that we can explore the performance of pre-selected candidates and quantify the gain achieved from the usage of more complex techniques. The second strategy is to directly solve the so-called CASH problem (Combined Algorithm Selection and Hyperparameter optimization [57]) using automated machine learning (AutoML) [49]. AutoML deals with the automation of the application of machine learning to real-world problems [55]. The CASH problem is the task of selecting a suitable machine-learning algorithm (which can be a combination of several algorithms) for the analyzed dataset, together with the proper pre-processing methods and set of hyperparameters of all components involved, without requiring human intervention [55]. The advantage of using this strategy is that such a technique enables an efficient selection of a machine-learning algorithm and feature preprocessor from a more extensive list of candidates within a limited time frame. This is particularly useful when performance benchmarks are unavailable. The disadvantage of such a technique, however, is that the comparison of the performance of several models cannot be done directly (as weaker candidates are discarded during the process of optimization). In this thesis, we address the CASH problem using TPOT (Tree-based Pipeline Optimization Tool) [77] – a Python package for automatic selection of machine learning pipelines based on genetic programming (GP) [58].

In order to combine the process of model performance evaluation with the process of algorithm selection and optimization of its hyperparameters, we apply a *nested cross-validation* scheme [96, 18]. The general setup of *nested cross-validation* is as

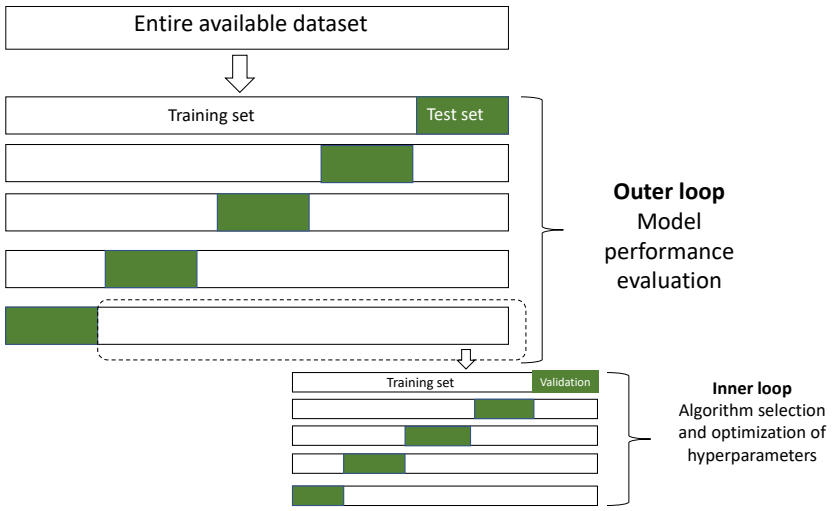

Figure 2.5: Schematic representation of nested cross-validation. In the inner loop, the generated training and validation sets are used to find an optimal set of the hyperparameters of the model. In the outer loop, we generated a series of test sets that are used for the model performance evaluation.

follows: In the outer loop of cross-validation, the entire dataset is split into K subsets (folds). The model is trained on K-1 subsets, while the remaining subset is used for the model evaluation. This procedure is repeated K times. Within each iteration of the outer loop, an inner cross-validation loop is performed. The training data from the outer loop is further split into K-1 subsets for training and one subset for validation. Different model hyperparameters are tested using the training and validation sets in the inner loop. The model with the best performance on the inner loop validation set is selected. The selected model from the inner loop is then evaluated on the test set from the outer loop. For a visual explanation, see Figure 2.5. Note that, in the field of statistics, another naming convention for resulting splits of the data is used (validation is then called test set, and vice-versa). The main advantage of the nested scheme of cross-validation is the prevention of information leakage coming from using the same data for the evaluation of model performance and tuning the hyperparameters, which takes place in case straightforward cross-validation is applied [18].

2.4 Results evaluation

The last step of the general workflow is a comparison of estimated values of NO_2 with some kind of independent measurement. However, as mentioned earlier, for the task of ship emission monitoring, TROPOMI is the only way of measurement above the open sea. The "ground truth" data for this task is not available. To overcome this, we will use a theoretical ship emission proxy E_s as a reference value defined as follows:

$$E_s = L_s^2 \cdot U_s^3, \tag{2.7}$$

where L_s is the length of the ship in meters (m), and U_s is its speed in meters per second (m/s). The details of the derivation of the given measure can be found in [41], where the proxy was introduced. As it is noted in [41], the advantage of E_s in comparison to other ship emission proxies (e.g. [33]) is that it can be calculated based on AIS data only, while other existing emission proxies require ship information that is not in the AIS data and is not available publicly. This, however, will result in some loss of the quality of emission approximation.

To sum up, in this Chapter, we described general process steps that will be used throughout the thesis. We call it the general workflow. The applied methodological details of each of the described steps may differ depending on the task at hand and will be described in each chapter separately.