Supervised Classification of Aerosol types from POLDER-3 and OMI satellite data.

An investigative study into using microphysical parameters and class labels for aerosol classification

Snigdha Narra

Delft University of Technology

TU Delft

Supervised Classification of Aerosol types from POLDER-3 and OMI satellite data: An investigative study into using microphysical parameters and class labels for aerosol classification

by

Snigdha Narra

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday May 03, 2022 at 09:30 AM.

Student number: 4748158
Project duration: January 1, 2021 – May 3, 2022
Thesis committee: Prof. S. Speretta, TU Delft, supervisor

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Summary

Aerosol is defined as the suspension of solid or liquid in the atmosphere. Its size can range from 0.001 to 200 μm. While some aerosols do not pose any serious threat to humankind, others have devastating effects. Thus, it is important to understand the type and distribution of these aerosols in the atmosphere. While traditional satellite data poses delays due to the huge amount of data that it has to process through the traditional pipeline, machine learning is quickly proving to be a likely winning candidate in providing accurate and efficient models. The advances in machine learning and cloud computing combined with the terabytes of data from the earth observation satellites opens up avenues for creating newer and variant data products of better accuracy in the domain of aerosol classification. There is a recognized need for distinguishing and characterizing different kind of aerosols in the 5.6 billion dollar air quality market. This research focuses on the investigation and designing of machine learning models for aerosol retrieval process. To meet this end we implement supervised learning on satellite data to achieve aerosol classification to distinguish the different types of aerosols. The two satellites whose data will be analysed are *Polarization and Directionality of the Earth's Reflectances* (POLDER)-3 and *Ozone Monitoring Instrument* (OMI). In this study, the three supervised learning algorithms *Support Vector Machine* (SVM), *Random Forest* (RF) and *K Nearest Neighbours* (KNN) were implemented to classify aerosol types for the year 2006 on POLDER-3 and OMI satellite data. We used results from previous studies on POLDER-3 as eight input aerosol class label along with selected microphysical parameters for supervised learning and could achieve a very high reproducibility of the aerosol classes with a reduction in training time. Similarly, we used three aerosol label classes as input along with selected microphysical parameters to generate a high reproducibility. The results showed that SVM performed best on POLDER-3 data while RF was the best performing algorithm on OMI data. Using SVM on POLDER-3 dataset with hyperparameter tuning we reached an overall accuracy of 99%, precision of 99%, recall of 98% and f1-score of 99% on POLDER-3 dataset for the eight classes of aerosols. Using RF on OMI dataset with hyperparameter tuning we reached an overall accuracy of 99%, precision of 99%, recall of 99% and f1-score of 99% on OMI dataset for the three classes of aerosols. It was concluded that while machine learning still has a long way to go, it shows promising results in the field of satellite data processing for aerosol classification.

Keywords: Supervised Learning, OMI, POLDER-3, satellite data, aerosol classification, earth observation, SVM, RF, KNN

Preface

This report represents the culmination of my journey as a Master's student at the Delft University of Technology. In this long master's journey, I've not only learnt technical skills but also life lessons. I have far too many people to be grateful for to see the light of day to defend my thesis. Firstly, my parents Ravindra Narra and Sunitha Kolli. Thank you for providing me with every bit of support and guidance that I needed along the way. The countless number of life lessons you shared when I fell will be with me throughout my life. Secondly, my supervisor, Stefano Speretta, for continuously guiding me and motivating me to finish what seemed like a gargantulean task. The weekly meetings were a great support in finishing the thesis. I would also like to thank Johan de Vries from Airbus for introducing me to the topic and guiding me on the fundamentals in the initial days. I would like to thank all my colleagues at Airbus, who have shined their light. Having relocated to the United States, I spent many a sleepless night contemplating if I would ever be able to finish the thesis. The trick seemed to be able to split it into work packages. A heartfelt thank you to my friend Gauri for introducing me to the idea of pursuing aerospace engineering at Delft. I would like to thank the numerous friends I made in Delft, who have gone out of their way to help me while I was at Delft and when I was away. You all will always hold a special place in my heart. A big thank you to my brother, Surya Narra, for teaching me never to let failure get you down. My heartful thanks to Rama Aunty and family for giving me a nurturing home during the hardest phase of my thesis. Lastly, to my husband, Ravi Kodali and my son Arjun, we made it across the finish line!
Greatest gratitude to TU Delft for making me realise that I could be stronger than I thought I was.

Snigdha Narra
Pune, April 2022

Contents

Summary i

Preface ii

Nomenclature v

List of Figures viii

List of Tables xi

1 Introduction 1
- 1.1 Thesis Research Question . . . 1
- 1.2 Thesis Research Outline . . . 2
- 1.3 Report Layout . . . 2
- 1.4 Contributions of the thesis . . . 2

2 Background 3
- 2.1 Historical Background . . . 5
- 2.2 Market share . . . 6
- 2.3 Atmospheric Aerosol . . . 7
- 2.4 Previous Studies on aerosol classification using satellite data . . . 10

3 Satellite Missions 12
- 3.1 Historical Missions for Aerosol measurements . . . 12
- 3.2 *Polarization and Directionality of the Earth's Reflectances* (POLDER)-3 (PARASOL) . . . 16
- 3.3 *Ozone Monitoring Instrument* (OMI) . . . 18
- 3.4 Chapter Summary . . . 19

4 Supervised Learning Algorithms 20
- 4.1 *K Nearest Neighbours* (KNN) . . . 20
- 4.2 *Random Forest* (RF) . . . 23
- 4.3 *Support Vector Machine* (SVM) . . . 24

5 Methodology 26
- 5.1 Research Work flow . . . 26
- 5.2 Data Collection . . . 29
- 5.3 Machine Learning Algorithms . . . 31
- 5.4 Environment . . . 32
- 5.5 Model Evaluation . . . 37
- 5.6 High Level Requirement Traceability Matrix . . . 39

6 Implementation and Results 40
- 6.1 POLDER Data . . . 40
  - 6.1.1 Validation . . . 55
  - 6.1.2 Discussion on POLDER-3 . . . 59
- 6.2 OMI Data . . . 61
  - 6.2.1 Validation . . . 68
  - 6.2.2 Discussion on OMI . . . 71
- 6.3 Discussion of the Results on POLDER-3 and OMI data . . . 72

7 Conclusion and Recommendations 74
- 7.1 Recommendations and Future Work . . . 75

Bibliography **80**

A Overview of past studies on Machine Learning on Aerosol Classification **81**

B Code repository **83**
B.1 POLDER parameters 83
B.2 R coding for POLDER data. 85
B.3 Motivation for the project 85

C Meeting with Atmospheric Sciences Experts **86**
C.1 Meeting with Dr. Pepijn Veefkind 86
C.2 Meeting with Herman Russchenberg 86

Nomenclature

Acronyms

SRON *Netherlands Institute for Space Research*

KNMI *Royal Netherlands Meteorological Institute Koninklijk Nederlands Meteorologisch Instituut*

NASA *National Aeronautics and Space Administration*

S5P *Sentinel-5 Precursor*

PARASOL *Polarization and Anisotropy of Reflectances for Atmospheric Sciences coupled with Observations from a Lidar*

OMI *Ozone Monitoring Instrument*

S5P *Sentinal 5 Precursor*

POLDER *Polarization and Directionality of the Earth's Reflectances*

TROPOMI *Tropospheric monitoring Instrument*

CIMON *Crew Interactive Mobile Companion*

DLR *German Aerospace Center [Deutsches Zentrum für Luft- und Raumfahrt]*

SVM *Support Vector Machine*

KNN *K Nearest Neighbours*

RF *Random Forest*

AI *Artificial Intelligence*

AI4EO *Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond*

AVHRR *Advanced Very High Resolution Radiometer*

EO *Earth Observation*

EOSDIS *Earth Observing System Data and Information System*

SAM *Stratospheric Aerosol Measurements experiment*

AEM *Applications Explorer Mission*

VOC *volatile organic components*

PA *Primary aerosol*

SA *Secondary aerosol*

IOA *Index of Agreement*

R2 *Coefficient of Determination*

EARLINET *European Aerosol Research Lidar Network*

EAE *Extinction Angstrom Exponent*

AERONET *Aerosol Robotic Network*

ML *Machine Learning*

AOD *Aerosol Optical Depth*

MODIS *Moderate Resolution Imaging Spectroradiometer*

ANN *Artificial Neural Network*

CALIOP *Cloud-Aerosol Lidar with Orthogonal Polarisation*

MISR *Multi-Angle Imaging SpectrRadiometer*

SWIR *Short-Wave Infrared*

NIR *Near Infrared*

AOT *Aerosol Optical Thickness*

AE *Angstrom Exponent*

EPA *Environmental Protection Agency*

XG *Extreme gradient*

ESA *European Space Agency*

GACA *Global aerosol categorization method*

AIRS *Atmospheric Infrared Sounder*

RBF *Radial Basis Function*

Symbols

Symbol	Definition	Unit
m	Meter	-
V	Velocity	[m/s]
ρ	Density	[kg/m^3]
δ	Delta representing change or increment	-
m_r	Real Refractive Index	-
$reff$	Effective radius	-
N	Aerosol Column number Density	-
$sphere_(frac)$	Sphericity	-

List of Figures

2.1 Distribution of satellites across the Earth. Data source from European Space Agency, The Society of Concerned Scientists, Business Insider and Euroconsult [69] . . . 3
2.2 Distribution of satellites across the Earth. Data source from European Space Agency, The Society of Concerned Scientists, Business Insider and Euroconsult bar graph [69] . . . 4
2.3 Aerosol Types [2] . . . 5
2.4 Commercial *Earth Observation* (EO) data market and Value Added Services Market in 2015 [17] 6
2.5 Atmospheric Aerosol Distribution Across the Globe. Goddard Earth Observing System Forward Processing model output for aerosols on August 23, 2018. The visualization depicts huge plumes of smoke drifted over North America and Africa. Also visible are three different tropical cyclones churned in the Pacific Ocean, and large clouds of dust blew over deserts in Africa and Asia. [62] 7
2.6 Report by Greenpeace Center for research on Energy and Clean Air [5] . . . 8
2.7 Aerosol Atmospheric processes [63] . . . 8
2.8 Passive sensing working [3] . . . 10

3.1 Early beginning of dedicated atmospheric study missions beginning with the *Stratospheric Aerosol Measurements experiment* (SAM) instrument. Although it took measurements only during sunrise and sunset, it laid the foundation for the beginning of atmospheric studies from space. [45] . . . 12
3.2 Timeline of POLDER-3,OMI and TROPOMI Instruments . . . 15
3.3 POLDER-3 onboard the PARASOL satellite . . . 16
3.4 Parasol and POLDER-3 instrument in the A train with Aura withOMIinstrument [16] . . . 17
3.5 Parasol lifetime [16] . . . 17
3.6 OMI onboard the Aura satellite . . . 18

4.1 KNN Algorithm flowchart [41] . . . 21
4.2 Example of KNN classification [31] . . . 22
4.3 Random Forest Classifier [13] . . . 23
4.4 Visualizing the hyperplane separating the two feature vectors of classes in 2D and 3D plane [1] . 25

5.1 Timeline of POLDER-3, OMI and TROPOMI Instruments. In this study we explore the data for the year 2006 for POLDER-3 and OMI. TROPOMI is recommended as future work. . . . 27
5.2 Methodology Outline . . . 28
5.3 Splitting the data into test and train sets using the cross validation approach to prevent over-fitting of the data . . . 29
5.4 Relationship between different levels of satellite data [41] . . . 30
5.5 Anaconda Graphical User Interface . . . 32
5.6 Jupyter Notebook . . . 33
5.7 Sample Confusion Matrix format for OMI biomass burning . . . 37
5.8 Bias and Variance Tradeoff [18] . . . 38
5.9 Bias and Variance . . . 39

6.1 Dataframe created from the extracted variables from the POLDER-3 data . . . 41
6.2 Feature correlation matrix for the four features. . . . 42
6.3 Weights for each hyperphysical parameter in POLDER-3 data for the year 2006 [16] . . . 42
6.4 Monthly distribution of the data points . . . 43
6.5 Coordinates for the aerosol reading from the satellite data [21] . . . 43
6.6 Seasonal variation in Single Scattering Albedo . . . 44
6.7 Box plot of SSA distribution vs seasons . . . 45
6.8 Seasonal variation in Real Refractive Index . . . 45

6.9 Box plot of RRI distribution vs seasons . . . 46
6.10 Seasonal variation in Angstrom Exponent . . . 47
6.11 Box plot of Angstrom exponent distribution vs seasons . . . 47
6.12 Seasonal variation in spherical coarse fraction . . . 48
6.13 Box plot of Spherical Fraction distribution vs seasons . . . 48
6.14 Feature Pair Plot . . . 50
6.15 Root Mean Squared Error results of KNN, RF and SVM . . . 51
6.16 Parameter search . . . 52
6.17 Dependence of C and Gamma with accuracy. The Y axis represents the mean test score of accuracy taken over the five folds of cross-validation. The X axis represents the parameter C varied over 0.1, 1, 10 and 100. The line graphs represent the four parameters of gamma at 0.001, 0.01, 0.1 and 1. . . . 53
6.18 Dependence of C and Gamma with accuracy. The Y axis represents the mean test score of accuracy taken over the five folds of cross-validation. The X axis represents the parameter Gamma varied over 0.001, 0.01, 0.1 and 1. The four line graphs represent different values of parameter C at 0.1, 1, 10 and 100. . . . 53
6.19 Dependence of C and Gamma with rank score. The Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter C. The four line graphs are for parameter gamma at 0.001, 0.01, 0.1 and 1.0. . . . 53
6.20 Dependence of C and Gamma with rank score. The Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter Gamma. The four line graphs are for parameter C at 0.1, 1.0, 10.0 and 100.0. Rank 1 is for C =100 and gamma =0.001. The rank 16 or lowest rank is for C =0.1 and gamma = 0.001. . . . 53
6.21 Dependence of C and Gamma with mean fit time. The Y axis represents the mean fit time in seconds. The X axis contains the parameter Gamma. The four line graphs are for parameter C at 0.1, 1.0, 10.0 and 100.0. . . . 54
6.22 Dependence of C and Gamma with mean fit time. The Y axis represents the mean fit time in seconds. The X axis contains the parameter C. The four line graphs are for parameter Gamma at 0.001, 0.01, 0.1 and 1.0. . . . 54
6.23 Dependence of C and Gamma with mean fit time. The Y axis represents the mean fit time in seconds. The X axis contains the parameter C. The four line graphs are for parameter Gamma at 0.001, 0.01, 0.1 and 1.0. . . . 54
6.24 Dependence of C and Gamma with rank score. The Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter C. The four line graphs are for parameter gamma at 0.001, 0.01, 0.1 and 1.0. Rank 1 is for C =100 and gamma =0.001. . . . 55
6.25 Dependence of C and Gamma with test score. The Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter Gamma. The four line graphs are for parameter C at 0.1, 1.0, 10.0 and 100.0. Rank 1 is for C =100 and gamma =0.001. The rank 16 or lowest rank is for C =0.1 and gamma = 0.001. . . . 55
6.26 Dependence of Variance and Bias on C and Gamma . . . 55
6.27 Confusion Matrix for the 8 cluster classes . . . 57
6.28 Precision, Recall and F-1 Score . . . 57
6.29 Plot of all eight clusters . . . 58
6.30 Aerosol Clusters throughout the year. Here the eight clusters are plotted on the world map. Cluster 1 is smoke, 2:Mixed Smoke, 3:Marine, 4:Urban Industrial, 5:dusty Smoke, 6:Marine Dust, 7:dust and 8:Polluted Dust. . . . 59
6.31 362 daily OMI Level 3 files distribution . . . 62
6.32 Dataframe containing variables extracted for OMI data. Here there are six columns containing Latitude, Longitude, Aerosol Model, Aerosol Optical Thickness, Single Scattering Albedo and UV Aerosol Index for a single day in Level 3(L3) file. . . . 62
6.33 Feature correlation matrix for the three features and Aerosol Model. . . . 63
6.34 Plotting four major aerosol types for OMI from January to April. 1 - WA – Weakly absorbing 2 – BB – Biomass Burning 3 – DD – Desert Dust 4 - VO – Volcanic Aerosols . . . 63
6.35 Plotting four major aerosol types for OMI from May to August. 1 - WA – Weakly absorbing 2 – BB – Biomass Burning 3 – DD – Desert Dust 4 – VO – Volcanic Aerosols . . . 64

6.36 Plotting four major aerosol types for OMI from September to December. 1 - WA – Weakly absorbing 2 – BB – Biomass Burning 3 – DD – Desert Dust 4 – VO – Volcanic Aerosols 64
6.37 Feature Pair Plot 65
6.38 Root Mean Squared Error results of KNN, RF and SVM on OMI data 66
6.39 Mean test score variation with changing max depth parameter for different parameter criterion 'gini' and 'entropy'. The Y axis represents the mean test score accuracy of the five folds of cross-validation. The X axis depicts the parameter maximum depth which is varied from 4,5,6,7,8. The two column represent the parameter criterion and entropy as explained in Chapter 5. Line graphs are drawn for three parameters 'auto', 'sqrt' and 'log2'. 67
6.40 Analyzing the optimization results. The Y axis represents the mean test score accuracy of the five folds of cross-validation. The X axis depicts the parameter maximum depth which is varied from 4,5,6,7,8. The X axis depicts the parameter maximum depth which is varied from 4,5,6,7,8. The three columns represent the three parameters of max_features 'auto', 'sqrt', and 'log2'. In each graph, the two ling plots represent the criterion 'gini' and 'entropy'. 67
6.41 Analyzing the optimization results. The Y axis represents the mean test score accuracy of the five folds of cross-validation. The X axis shows the categorical variation between the parameters 'gini' and 'entropy'. The three columns represent the parameter max features 'auto', 'sqrt', and 'log2'. In each plot the line graphs represent the max depth varied from 4,5,6,7,8. 68
6.42 Analyzing the optimization results - 4. The Y axis contains the mean fit time taken over the five folds of cross-validation. The X axis the parameter max depth varied over 4,5,6,7,8. The three columns represent the parameter max features 'auto', 'sqrt' and 'log2'. In each plot the two line graphs show the two parameters of n estimators as 200 and 500. 68
6.43 Confusion Matrix for Random Forest 71
6.44 Precision, Recall and Support 71

B.1 Github link 83

List of Tables

2.1 History of platforms and sensors used to derive aerosol properties from space [37] 6
2.2 Various Aerosol Types and their distribution in $\mu g/m^3$ 9

3.1 Previous Aerosol Observation Missions [37]. 13
3.2 Three aerosol measuring instruments at a glance: A comparison 14
3.3 POLDER-3 data parameters [16] 17
3.4 OMI instrument specifications [65] 19

4.1 Features of small and large values of k [71] 22

5.1 Data collection for POLDER-3 and OMI satellite data 30
5.2 Machine Learning algorithm merits and demerits 31

6.1 POLDER-3 data parameters [16] 40
6.2 Eight clusters found in POLDER-3 data that used as labels for supervised learning for N = 1, 131, 324 resulting in cluster labels for aerosols on 339,402 datapoints. observations [16] 49
6.3 Centroids of the eight found clusters and their hyperparameter values [16] 49
6.4 Cross-validation Table for POLDER-3 data 56
6.5 OMI instrument specifications [48] 61
6.6 Difference between aerosol products from OMI 61
6.7 Cross-validation Table for OMI data 69
6.8 Requirements 73

A.1 Overview of past studies on Machine Learning on Aerosol Classification 82

B.1 Expected Aerosols from literature over certain regions on the globe [2] 85

Glossary

AERONET The AERONET (AErosol RObotic NETwork) is a network of ground-based Sun photometers that measure atmospheric aerosol properties. [28]. 24

Bias Bias is defined as the difference between the average value of prediction of the model and the correct value of the prediction

. 38

RRI Real Refractive Index is an important parameter to determine the absorption and scattering of the particle along with the information regarding the particles size

. 45

Self Organizing Map A self-organizing map or self-organizing feature map (SOFM) or Kohonen map is an unsupervised machine learning technique used to produce a low-dimensional representation of a higher dimensional data set while preserving the topological structure of the data. For example, a data set with p variables measured in n observations could be represented as clusters of observations with similar values for the variables. [32]

. 27

SSA Single Scattering Albedo is defined as the ratio of scattering efficiency to the total extinction efficiency or the sum of both absorption and scattering

. 24, 43

Variance The variability of model prediction for a specific data point or value, which tells us about the dispersion of our data, is known as variance.

. 38

1 Introduction

> The story of civilization is, in a sense, the story of engineering - that long and arduous struggle to make the forces of nature work for man's good.
>
> *L. Sprague de Camp*

In this chapter, a brief motivation for the research area and an introduction to the research questions are presented. This is followed by the requirements that govern the research project and the software that is subsequently developed. The main aim of the thesis research work is to develop machine learning techniques on satellite data for the purpose of Earth observation. The main content will be exploring the supervised and unsupervised machine learning techniques on past satellite data for the purpose of aerosol classification.

Aerosols come in different size and shapes. While, some aerosols are so minute that they can only be viewed with an electron microscope and are made up of a few molecules, other aerosols are large enough to be seen with the naked eye, but small enough to float in the air. A particle's lifetime in the atmosphere is proportional to its size and weight. Larger particles tend to sink to the ground in a few of hours, but smaller particles take longer. Irrespective of whether we can see them or not, their presence is almost always felt through their interactions with the incident sunlight by reflecting, absorbing or radiating it. Thus, the best measure to gather data on the aerosols is through satellite sensors measuring these light interactions. Through this study we aim to understand the present distribution of aerosols and to have a better understanding of the Air Quality over the world through the satellite data. The goal of the research is to develop a supervised learning algorithm to create an aerosol classification algorithm, wherein, the primary objective of the classification algorithm is to convert abstract aerosol parameters derived from the satellite sensors, like the size distributions, index of refraction, shape, Angstrom exponent and Aerosol Index that sensors provide into distinct Aerosol types. The science output of this study would help to identify the good and bad aerosols with regard to impact on human beings as well as help understand the combination of sources on a global scale.

The scope of the research project is as follows:

- To classify the satellite data especially on atmospheric aerosols using supervised learning method.
- To build an efficient model which performs better on the satellite aerosol retrievals.
- To optimise the classification accuracy of the supervised learning models using hyperparameter tuning.

1.1. Thesis Research Question

Through the thesis study the following questions will be answered.

1. What are the past satellite missions that give information regarding the atmospheric aerosols?
2. Which algorithms have higher accuracy on the satellite aerosol retrievals?
3. What is the classification accuracy of the machine learning algorithms SVM, KNN and RF for aerosol classification?

(a) Is it possible to build a higher accuracy than that presently achieved by the existing algorithms for aerosol classification?
(b) How do the classification results of SVM, KNN and RF compare with each other?

1.2. Thesis Research Outline

Based on the literature study, suitable machine learning algorithms were identified which could be used on satellite data. The theory of underlying machine learning techniques was understood. The advantages and disadvantages of the machine learning techniques were compared. The next step after the literature review, would be to gather the data. Satellite data would be gathered and pre-processed such that it could be used as an input to the machine learning algorithms. The data would be then split into train, test and validation datasets.
Following this, the machine learning algorithms would be implemented and the performance metrics would be analyzed to understand the accuracy of the algorithms. This would be followed by the analysis of the results. The final step of the thesis would be the report detailing all the above-mentioned steps.

1.3. Report Layout

The thesis report in divided into the following chapters:

- Chapter 1 is the introduction to the thesis document.
- Chapter 2 provides all the background required to understand the topic related to the thesis along with a brief description of the motivation behind the thesis project. Further, it provides an overview on the present aerosol distribution.
- Chapter 3 provides the literature review of the thesis document.
- Chapter 4 presents all the basic information behind using machine learning.
- Chapter 5 describes the methodology used in the thesis project.
- Chapter 6 presents the implementation and results of the report. Section 6.3 presents the discussion of results.
- Chapter 7 is the conclusion to the thesis document along with the recommendations for future work.

1.4. Contributions of the thesis

Previous researchers like Vincent de Bakker and Russel et al used unsupervised learning to create aerosols clusters on POLDER-3 dataset. Our work extends the previous work done by various researchers on a single data set to two satellite data. While most of the focus area in the previous works has been on unsupervised learning, this work looks at supervised learning and the application to extend the code across multiple satellite data platforms. The aim of the study is to use machine learning algorithms used in past studies as collected in the literature study to see their applicability for aerosol classification. High accuracy and low error upon training the model while tuning the hyperparameters indicate the applicability of using machine learning models for aerosol classification. In this study we lower the training time to generate aerosol clusters by nearly half that of the 24 hrs needed in the study of de Bakker to classify with 99% accuracy for the same eight aerosol classes on POLDER-3 data. The Cluster 1 is smoke, 2:Mixed Smoke, 3:Marine, 4:Urban Industrial, 5:dusty Smoke, 6:Marine Dust, 7:dust and 8:Polluted Dust. On the OMI dataset, we are able to recreate the three aerosol classes : Weakly absorbing, Biomass burning and Desert Dust with 99% accuracy. Another important conclusion through this work is that there is no one size fits all solution to machine learning models for aerosol classification and infact does depend on the dataset at hand.

2

Background

”Space isn’t remote at all. It’s only an hour’s drive away, if your car could go straight upwards.”

Sir Fred Hoyle, Astronomer

We have come a long way from October 4, 1957 when the first satellite Sputnik was launched. In a little over 63 years, almost 8,900 satellites have been launched from all across the world. From this astounding number to date, approximately 1,900 remain operational, continuously gathering scientific data or providing the infrastructure for communication.

Figure 2.1: Distribution of satellites across the Earth. Data source from European Space Agency, The Society of Concerned Scientists, Business Insider and Euroconsult [69]

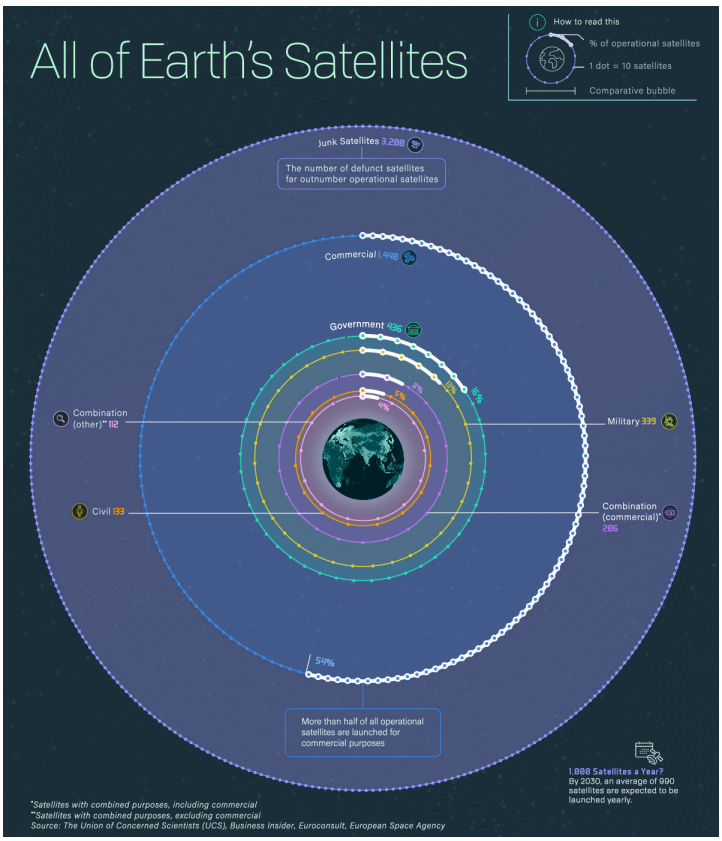
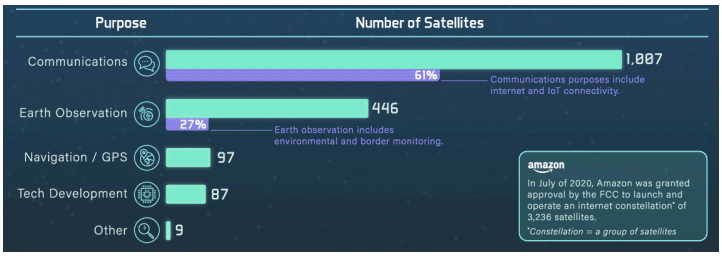


Figure 2.2: Distribution of satellites across the Earth. Data source from European Space Agency, The Society of Concerned Scientists, Business Insider and Euroconsult bar graph [69]



Given the vast amount of operational satellites and an even larger amount of data coming in, maintaining and analyzing the data to provide sound scientific output in a timely manner proves to be a challenge for traditional computing. Here, machine learning proves to be a useful tool. The advances in machine learning and cloud computing combined with the terabytes of data from the Earth observation satellites opens up avenues for creating newer and variant data products of better accuracy, modelled to the requirements of the customer. Google Earth Engine is a successful commercial example of using satellite data coupled with large scale cloud computing services to provide customizable user output in the field of Earth Observation[22][49].*National Aeronautics and Space Administration* (NASA) on its official website also claimed to encourage the adoption of Artificial Intelligence Techniques to space science in collaboration with tech companies like Google, IBM, and Intel[59]. In the year 2018, *Crew Interactive Mobile Companion* (CIMON) was the first *Artificial Intelligence* (AI)-based robot deployed on the International Space Station. As a result of Airbus, IBM, and *German Aerospace Center [Deutsches Zentrum für Luft- und Raumfahrt]* (DLR)'s collaboration, CIMON showed that AI technologies were slowly making their way into the otherwise conservative space industry [14].

Although history has shown us that the space industry has been slow in adopting AI and much less to on-board applications on the spacecraft, the trends have been changing due to the continued efforts by the industry and the governmental agencies. A recent 2019 report by Cosine Remote Sensing and ESA/ESTEC, The Netherlands, demonstrated the application of Artificial Intelligence for the purpose of cloud screening onboard a spacecraft[19]. The AI algorithm was applied to data collected from the hyperspectral imager and a thermal infrared radiometric spectrometer integrated into a CubeSat platform. The payload is called HyperScout-2 and is capable of acquiring the data, processing the data from L0 to L2 levels, and deploying the neural network algorithm [19]. The neural network pipeline deployed on HyperScout-2 is said to cater to a wide range of applications ranging from disaster mitigation, fire or fire hazard notification to agriculture monitoring.

The applications of Artificial Intelligence in the domain of Earth Observation is seeing new found growth with investments like the 5 million euros that were made to the AI labs in TU Munich by the federal government [1]. An example of this growth is the "Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond *Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond* (AI4EO)" laboratory which is a collaboration of TUM and DLR. Led by Prof. Xiaoxiang Zhu, one of the application areas is to ascertain the change in the level of the layer below the Earth's surface, which could be an indicator of the potential collapse of urban infrastructure like dams, tunnels, and the like. Based on a report generated by the United Nations, it was estimated that by 2050 almost 2/3rds of the world population would be living in cities, which further drove in the need for the research in the area [46].

Based on an internal report at Airbus Defence and Space Netherlands, the Air Quality Market was estimated to grow to 5.6 Billion Dollars in 2021[26]. Enhancing the aerosol remote sensing data for the air quality market was predicted by the report to usher new policies that could bring overall economic benefits that outweigh costs 20:1. That same report mentions that the annual welfare loss worldwide was 5.1 Trillion dollars[26], almost more than 600 dollars per person.

[1] `https://www.tum.de/nc/die-tum/aktuelles/pressemitteilungen/details/36026/`
[2] https://www-sciencedirect-com.tudelft.idm.oclc.org/science/article/pii/S0034425717302900

2.1. Historical Background

”There’s so much pollution in the air now that if it weren’t for our lungs there’d be no place to put it all. Robert Orben”

Carl Sagan

Figure 2.3: Aerosol Types [2]

Sea salt, dust, and volcanic ash are three common types of aerosols. (Photograph by Katherine Mann.)

Soot, Ash, Smoke, Sulphates, Organic Carbon, Nitrates, Mineral Dust and Sea Salt. What these have in common are that these particulates in the atmosphere are termed as Aerosols. Over the past few decades, atmospheric instruments like POLDER, OMI and *Tropospheric monitoring Instrument* (TROPOMI) have been giving major insights into understanding the aerosols and trace gases in ways not imaginable in the earlier years.

The history of measuring the Atmospheric Aerosols began with the *Advanced Very High Resolution Radiometer* (AVHRR) in the 1970s. While the Aerosol detection is not limited to the satellite remote sensing and satellite imagery, but also includes ground-based aerosol detection along with aerial detection, the literature study in Chapter 3 will focus on the Aerosol Classification using the satellites. In the last 40 years, atmospheric aerosol detection has seen the application of advanced satellite remote sensing technologies.

Table 2.1: History of platforms and sensors used to derive aerosol properties from space [37]

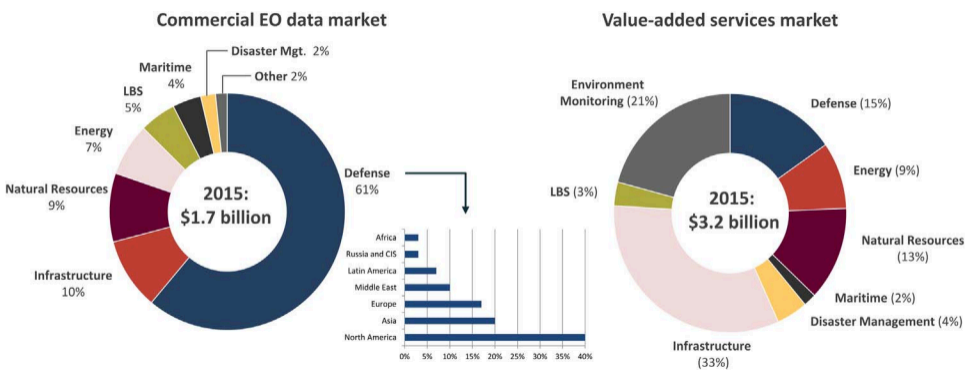
Satellite	Instrument	Launch	End	of bands (wavelength (μm))	Accuracy
Landsat	MSS	1972	1978	4 (0.5 - 1.1)	τ 10 %
SMS-1,2	VISSR	1974	1981	5 (0.65 - 12.5)	-
GOES-1	VISSR	1975	Present	5(0.65 -12.5)	τ 18 34 %
Apollo-Soyuz	SAM	1975	1975	0.83	-
GMS-1	VISSR	1977	2005	4(0.45 - 12.5)	-
TIROS-N	AVHRR	1978	1980	4(0.58 - 11.5)	-
Nimbus-7	SAM-2, CZCS, TOMS	1978	1993	1, 6(0.441-11.5), 6(0.385 - 0.380)	$\sigma_e xt$ (10%)
AEM-B	SAGE	1979	1981	4(0.45 -12.5)	-
NOAA 6-16	AVHRR	1979	Present	5(0.58 -12)	τ 10 % , τ 3.6 %
ERBS	SAGE-2	1984	2005	4(0.386 -1.02)	$\sigma_e xt$ (10%)
TRMM	VIRS	1997	Present	5(0.63 - 12)	τ 35 %, α (+- 0.5)
SPOT-3	POAM-2	1991	1996	9(0.353-1.060)	$\sigma_e xt$ (20%)
ERS-1	ATSR, GOME	1991	1999	4(1.6, 3.7, 11,12) , 4 (0.24-0.79)	-
UARS	HALOE	1992	2005	8(2.45 - 10.01)	$r_e ff$ (+- 15 %), $\sigma_e xt$ (5%)
SSD	LITE	1994	1994	3(0.355, 0.532, 1.064)	$\beta(\lambda_1)$ / $\beta(\lambda_2)$ (<5%)
ERS-2	ATSR-2, GOME	1995	Present	7(0.55 -12), 0.24-0.79	τ (<0.03), τ (30)%
Earth Probe	TOMS	1996	Present	6(0.309-0.360)	τ (20-30%)
ADEOS	POLDER, ILAS , OCTS	1996	1997	9(0.443-0.910), 2(0.75-0.78, 6.21-11.77), 7(0.412-0.865)	τ (20 -30 %)
Orb View -2	SeaWiFS	1997	Present	8(0.412-0.865)	T(5-10%)
SPOT -4	POAM-3	1998	Present	9(0.354-1.018)	$\sigma_e xt$ (+- 30%)
TERRA	MODIS,MISR	1999	Present	36(0.4-14.4), 4(0.35 0.87)	τ (5 - 15 %) , τ (10 - 20 %)
METEOR-3M	SAGE-3	2001	2005	9(0.385-1.545)	$\sigma_e xt$ (5%) ,τ (5 %)
PROBA	CHRIS	2001	Present	62(0.4-1.05)	-
Odin	OSIRIS	2001	Present	0.274-0.810	$\sigma_e xt$ (15%)
AQUA	MODIS	2002	Present	-	-
ENVISAT	AATSR, MERIS , SCIAMACHY	2002	Present	7(0.55-12.0), 15(0.4-1.05), 0.24-2.4	τ (0.16), τ (0.2), AI(0.4)
ADEOS -2	POLDER-2 , ILAS-2 , GLI	2002	2003	9(0.441-0.910), 4(0.75-12.85), 36(0.38-12)	τ (0.1)
MSG – 1	SEVIRI	2002	Present	12(0.6-13.4)	τ (0.08)
ICEsat	GLAS	2003	2003-	2(0.532, 1.064)	$\sigma_e xt$ (10%), τ (20%)
AURA	OMI , HIRDLS	2004	Present	3(0.27-0.5), 21(6-18)	τ (30%), $\sigma_e xt$ (5 25%)
PARASOL	POLDER-3	2004	Present	8(0.44-0.91)	-
CALIPSO	CALIOP	2006	Present	2(0.532,1.064)	-

2.2. Market share

The investment and the market share of the air quality market shows a continually increasing trend over the years and market research across multiple business analyst firms indicate a strong prevalence to such improving trends in the future. In such a scenario, investigating in the sources and types of aerosols and to derive more insights into the satellite data to understand the prevalence of aerosols around us in the air is a positive investment not only from an environmental, policy and health perspective but also from the monetary perspective.

"EO is the gathering data about planet Earth's chemical, physical and biological systems through remote sensing technologies supplemented via encompassing the collection, Earth surveying approaches, analysis and presentation of data" [8] . Generally, research is prevalent in the field of science and technology, like study of ecosystems, hydrology, climate and meteorology, forests area and marine life. EO helps to improve the lifestyle and protect populations from disasters e.g. forest fire detection, tsunami warning etc.

Figure 2.4: Commercial EO data market and Value Added Services Market in 2015 [17]



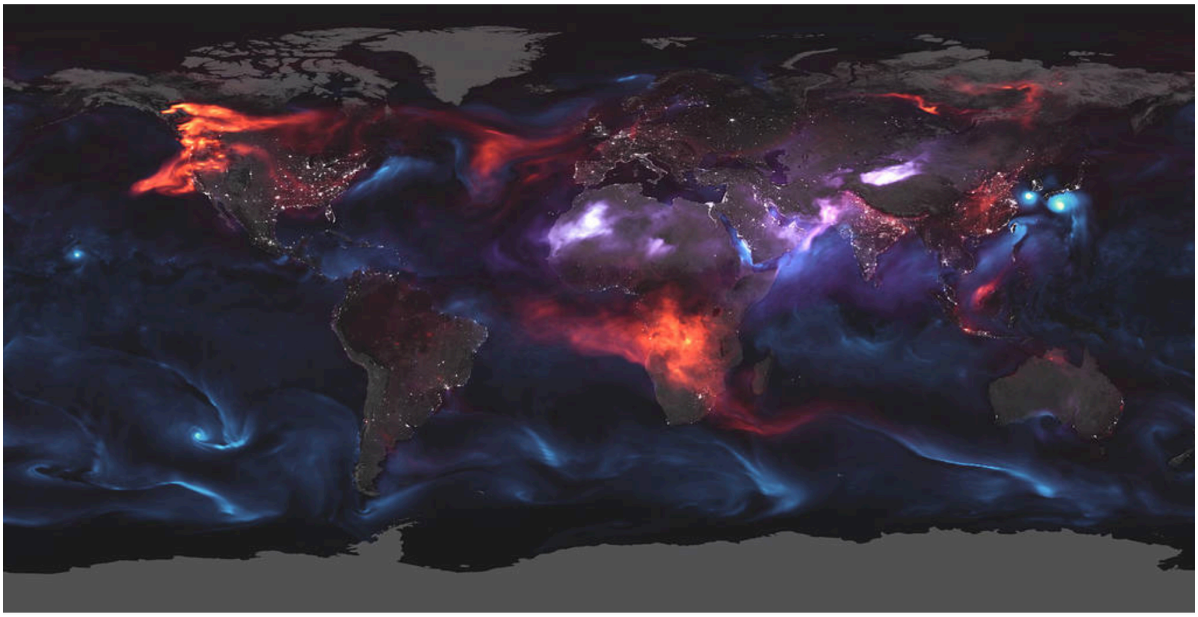
Even as far back as 2015, the Commercial EO data market was worth 1.7 billion dollars with Value-added services market valued at 3.2 billion dollars. In 2025, this evaluation is estimated to increase to 5.3 billion dollars [17]. Of this nearly 21 percent as of 2015 was attributed to environmental monitoring. From this evaluation, it is evident that there is increased interest in value-added services market in Earth Observation. To meet that end, Machine Learning can help add value to the satellite data by processing enormous data and detecting changing trends and classifying events on a global scale.[35]

2.3. Atmospheric Aerosol

> Man must rise above the Earth—to the top of the atmosphere and beyond—for only thus will he fully understand the world in which he lives.
>
> *Socrates*

The following section outlines the need for Air Quality monitoring, followed by an introduction to the science behind various aerosol types. Further, various studies conducted in the past on aerosol monitoring are highlighted.

Figure 2.5: Atmospheric Aerosol Distribution Across the Globe. Goddard Earth Observing System Forward Processing model output for aerosols on August 23, 2018. The visualization depicts huge plumes of smoke drifted over North America and Africa. Also visible are three different tropical cyclones churned in the Pacific Ocean, and large clouds of dust blew over deserts in Africa and Asia. [62]

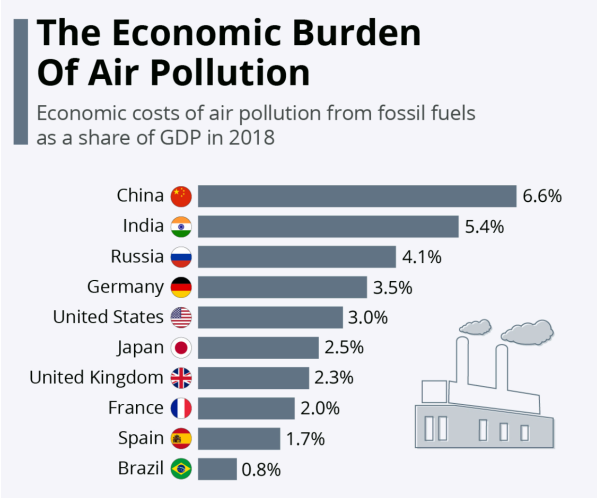


A report Air Quality Monitoring System Market by Product, Sampling, Pollutant, End-user, by Geography Global Opportunity Analysis and Industry Forecast up to 2026 reported that the Air Quality monitoring market would be worth 6.5 Billion dollars by 2025. [3]

Furthermore, a research by Greenpeace Center for Research on Energy and Clean Air [47] reported that the a country like China or India could end up spending almost 5 -6 % of the GDP due to Air pollution. Furthermore, it stated that 4.5 million deaths with aerosols with Particulate Matter 2.5 pollution are also responsible for 1.8 billion days of work absence, 4 million new cases of child asthma and 2 million preterm births. With Aerosols having such a profound effect on human lives and the global economy, a further study to understand what aerosols are and how they can be classified is necessary.
The atmospheric aerosol is described as a combination of liquid particles and/or solid particles suspended in air. Generally, the aerosol denotes the particulate component in atmospheric science. In most of the cases the

[3] https://www.prnewswire.com/news-releases/air-quality-monitoring-system-market-size-worth-6-5-billion-by-2025-grand-view-research-inc-300961673.html

Figure 2.6: Report by Greenpeace Center for research on Energy and Clean Air [5]



atmospheric aerosols is primarily formed in the lowest layers of the atmosphere, namely stratosphere, troposphere. Also it can occur in different sizes varying from nanometer scale to micro meter scale [5].

Figure 2.7: Aerosol Atmospheric processes [63]

Figure 2.7: Aerosol Atmospheric processes [47]

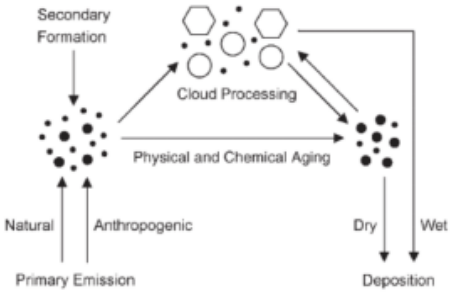


Table 2.2: Various Aerosol Types and their distribution in $\mu g/m^3$

	Source	Low	High	Best
Natural Primary	Soil Dust	1000	3000	1500
Natural Primary	Sea Salt	1000	10000	1300
Natural Primary	Volcanic Dust	4	10000	30
Natural Primary	Biological Debris	26	80	50
Natural Secondary	Sulfates from biogenic gases	80	150	130
Natural Secondary	Sulfates from volcanic SO2	5	60	20
Natural Secondary	Organic Matter from biogenic VOC(Volatile Organic Compounds)	40	200	60
Natural Secondary	Nitrates	15	50	30
Total Natural	2200	2200	23500	3100
Anthropogenic Primary	Industrial Particulates	40	130	100
	Dust	300	1000	600
	Soot	5	20	10
Anthropogenic Secondary	Sulfates from SO2	170	250	190
	Biomass Burning	60	150	90
	Nitrates from NOx	25	65	50
	Organics from anthropogenic VOC	5	25	10
Total Anthropogenic		600	1640	1050
Total		2800	26780	4150

Atmospheric aerosols can originate from anthropogenic sources, natural sources or it may form through atmospheric chemical processes. There are two types of aerosol formation in terms of formation mechanism namely *Primary aerosol* (PA) and *Secondary aerosol* (SA). In the atmosphere, SA is formed by gas-to particle formation and phase transitions. Here, the phase transition are either condensation or nucleation process while the formation of heterogeneous SA results from two or more (multi)- phase chemical reactions [71]. This type of SA may occur far from the source. Based on the mechanical breakup of parent materials, fragmentation, and some incomplete parent materials, PA can be formed. The primary particles are directly emitted into the atmosphere and PA inherit the chemical assets of parent material. Naturally, PA are surface-sourced. The aerosol particles that are directly inserted into the atmosphere are known as PA in Earth science. For example, natural PA are mineral dust, animal debris, sea spray, volcanic ash, plant debris, which are formed through mechanical resources. A huge volume of particles can be released into the atmosphere during a volcanic eruption, where few of these particles may reach very high altitudes (of above 10 km into the stratosphere). From different surface types, the dispersion and disintegration of animal and vegetal fragments and microbes are blown off, forming the biogenic component of PA [6]. Similarly, humans pollute the atmosphere and release primary anthropogenic aerosols from automobiles, biomass burning, industry, and agriculture. SAs may be formed with the presence of precursor gases that is originated from anthropogenic and natural aerosols like NO2, SO2, and *volatile organic components* (VOC) that account for gas to particle conversion. Specific to aerosol mass fluxes, the strength of various aerosol sources are discussed in table 1[6].

Studies over the previous years have shown that aerosol particles are related with well being problems including cardiovascular and respiratory diseases, cardiovascular, neurological, and respiratory sicknesses [70], particularly aerosol particulate matter 2.5 μm (PM 2.5) because of its capacity to infiltrate further and store in the lower respiratory tract [43].

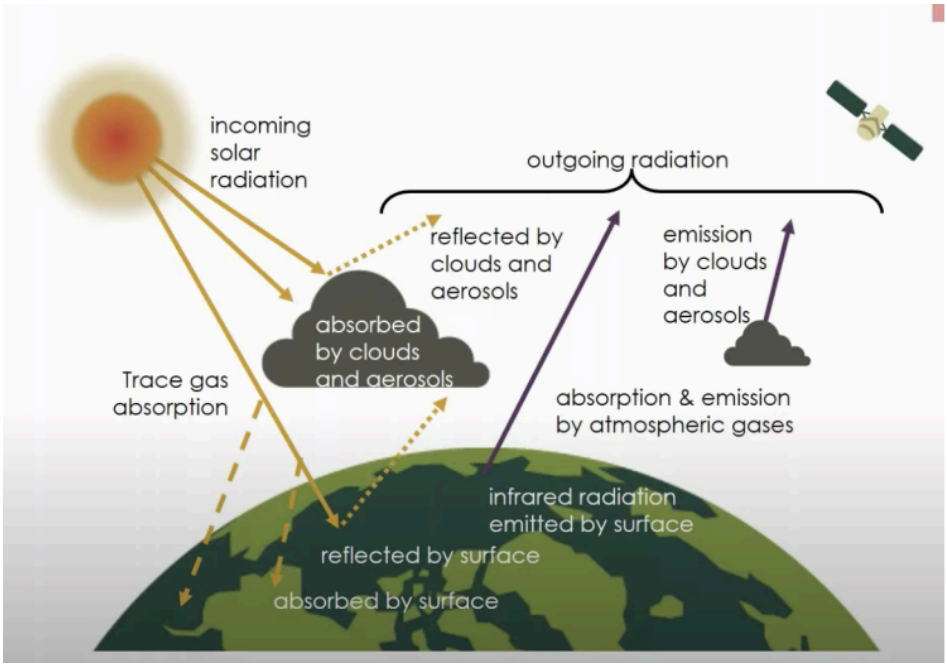
There is a recognized need for detection and characterization of aerosols for various domains including that of public health. Satellite aerosol retrieval provides a suitable method to detect and characterize these aerosols on a global scale. A study by Zheng et al. [73] has predicted the multi-aerosol mixing state metrics using machine learning technique that allows to improve the fundamental representation of aerosol mixing state that may lead to huge errors at a global scale. At the end of the simulation, they have concluded that the presented ensemble method, a combination of the deep learning and *Extreme gradient* (XG) boost methods gave the best results.The XG boost method has been dominating the machine learning methods for being a high-speed and high-performance implementation of gradient boosted decision trees. This ensemble method is able to predict the multi-aerosol mixing state metrics with acceptable predictive power (*Index of Agreement* (IOA) = 0.99, *Coefficient of Determination* (R2) = 0.99). Furthermore, it was stated that the ensemble technique gives better performance when

compared to pure deep learning and non-ensemble XG boost methods. In closing, it was highlighted that when selecting the predictive models, the trade-off among accuracy and run-time needs to be considered. In the future, they have planned to enhance the study by integrating effective learning models and increasing the data size which may be combined with an enhanced distribution of data.

Previous studies [16] on aerosol classification were performed on older satellite data (2008,[16]) on now defunct satellite. The aim of the research was also to apply machine learning algorithms to newer satellite data retrieval's that could give an insight into the present day classification of aerosol distribution on Earth.

How exactly the satellites measure these aerosols presence over the globe is evident in the figure below. An example to demonstrate this working is through the example of OMI. OMI is a passive sensing instrument, implying that it needs the sunlight to take measurements. As can be seen in Figure 2.8, the incident solar radiation is either absorbed by clouds and aerosols, reflected by the surface, emitted by clouds and aerosols or absorbed by trace gases. The satellite measures this outgoing radiation along with the direct incoming sunlight to analyse further constituent aerosols or presence of trace gases.

Figure 2.8: Passive sensing working [3]



The figure explains the working of passive remote sensing. Passive remote sensing does not use active power from the satellite, instead it uses the incoming solar radiation from various sources to detect energy. The strength of the reflected and emitted radiation is dependent on many surface and atmospheric conditions. The incoming solar radiation from the sun is either absorbed by the clouds and aerosols or absorbed by trace gases. It could also be reflected by the clouds and aerosols or the surface. As for the outgoing radiation, the emission from the clouds and aerosols along with that from the surface combined with the reflectance from the clouds, aerosols and surface are measured by the satellite. By measuring these incoming and outgoing radiation, the satellite can provide a description of the constitution of the atmospheric conditions, i.e. cloud cover, aerosols, trace gases and the surface conditions. It is this working principle of passive sensor that enable us to obtain data that later can be used to create better models of aerosol and distribution.

2.4. Previous Studies on aerosol classification using satellite data

Numerous satellite aerosol classification methods based on the threshold approach have been developed. Higurashi and Nakajima [27] devised a four-channel algorithm for detecting aerosol types using data from four-channel Sea-viewing Wide Field of View Sensors (SeaWiFS). In northern Asia, four aerosol species have been discovered above the ocean, including soil dust, carbonaceous, sulfate, and sea salt. Jeong and Li [30] proposed

a method for classifying aerosols based on data from the AVHRR and the Total Ozone Mapping Spectrometer (TOMS). The *Aerosol Optical Thickness* (AOT) and *Angstrom Exponent* (AE) from AVHRR were used in conjunction with the aerosol index from TOMS to classify aerosols into seven categories (biomass burning, dust, sea salt, pollution/sulfate). On a global basis, two of the twenty-one mixtures (biomass/dust mixtures, sulfate/sea salt mixtures, and unexplained mixtures) were identified. Lee et al.[38] classified aerosols into four major types (dust, sea salt, smoke, and sulfate) and two mixtures of major types over northeastern Asia using AOT and AE values from the *Moderate Resolution Imaging Spectroradiometer* (MODIS) and an aerosol index value from the OMI data. Additionally, those authors examined the geographical and temporal distributions of aerosol types across northeastern Asia and compared them to those predicted by a global aerosol climate model. Kim et al. proposed a MODIS–OMI algorithm to classify aerosol types using the aerosol index from OMI and the fine mode fraction from MODIS. The MODIS–OMI categorization results were compared to those of the 4CA over northeastern Asia, and their agreement ranged from 32% to 81%. Along with channel data and aerosol optical characteristics, Torres et al. pioneered the use of carbon monoxide, a tracer of carbonaceous aerosols, rather than AE, to discriminate between carbonaceous and dust aerosols. CO column values from the *Atmospheric Infrared Sounder* (AIRS) and an aerosol index were utilized to identify aerosols in an operational OMI near-UV aerosol algorithm. Penning de Vries et al. [52] developed a novel global aerosol categorization method *Global aerosol categorization method* (GACA) based on monthly average aerosol properties (*Aerosol Optical Depth* (AOD) and aerosol optical depth) and trace gas column densities (NO2, HCHO, SO2, and CO). On a monthly basis, the GACA categorized aerosol kinds, and the findings were compared to aerosol compositions produced from the global monitoring atmospheric composition and climate model. Mao et al. [42] classified aerosol types over eight study locations, including significant aerosol source regions and downwind of the source regions, using AOD and aerosol relative optical depth from MODIS. The consistency between satellite-based and ground-based data in Mao et al. ranges from 36% to 91% over the studied regions. Indeed, of the aforementioned satellite aerosol classification investigations, only Mao et al. attempted such a satellite and ground-based validation. The majority of the classification results were compared solely to those from an aerosol climate model and to those from older aerosol classification methods. Although accuracy assessments of aerosol categorization methods are uncommon, uncertainty has been documented in satellite aerosol optical characteristics and trace gas products. According to Chu et al.[12], the uncertainty of MODIS-derived AOD was (0.05 + 0.15 AOD), and that of MODIS AE was up to 30%. According to Thrastarson et al [66]., the uncertainty in AIRS CO products is 15%. These uncertainties in satellite input data can result in aerosol type misclassifications using threshold-based classification algorithms.

3

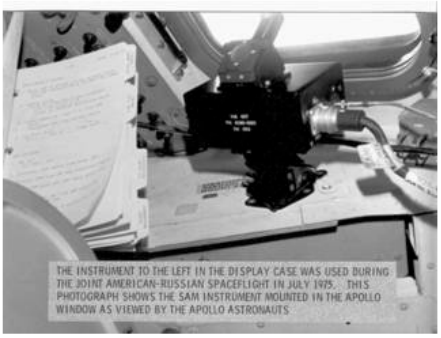
Satellite Missions

The present chapter will look at the historical missions that led to the global aerosol measurements from space in Section 3.1. From there on, we look at the three mission/satellites that will be studied in the thesis. Section **??** looks at the TROPOMI satellite on board the Sentinel 5 Precursor satellite followed with an introduction to the dataset. Section 3.3 looks at OMI onboard Aura satellite while Section 3.2 details POLDER-3 onboard *Polarization and Anisotropy of Reflectances for Atmospheric Sciences coupled with Observations from a Lidar* (PARASOL). Finally, Section 3.4 wraps up this chapter with an overview.

3.1. Historical Missions for Aerosol measurements

Back in July 1975, although atmospheric measurements had been done from balloons, they had never been done from space. SAM was the first experiment to be conducted in space to measure aerosols. Onboard the Apollo-Soyuz test flight, marking the end of the Apollo missions would mark the beginning of atmospheric studies onboard the spacecraft. SAM demonstrated that utilizing a method known as occultation from an orbiting vantage point may provide scientists with a far better lens on the stratosphere, which is a layer of the atmosphere that is 10-31 miles (16-50 kilometers) above the surface. While the instrument had only four orbits, it was a proof of concept of taking good quality atmospheric measurements from space, paving the way to several revisions on the instruments on follow on missions like the SAGE instrument that flew onboard *Applications Explorer Mission* (AEM)-B [45].

Figure 3.1: Early beginning of dedicated atmospheric study missions beginning with the SAM instrument. Although it took measurements only during sunrise and sunset, it laid the foundation for the beginning of atmospheric studies from space. [45]



The SAM instrument was manually pointed toward the sun during sunrise and sunset so that it could measure the properties of sunlight passing through the edges of Earth's atmosphere.
Credits: NASA

In the last 40 years , atmospheric aerosol detection has seen the application of advanced satellite remote sensing technologies. These advances have led to the development of new aerosol products of which are included in the table listed below[37].

Table 3.1: Previous Aerosol Observation Missions [37].

Satellite	Instrument	Launch	End	No. of bands (wavelength (μm))	Accuracy
Landsat	MSS	1972	1978	4 (0.5 - 1.1)	τ 10 %
SMS-1,2	VISSR	1974	1981	5 (0.65 - 12.5)	-
GOES-1	VISSR	1975	Present	5(0.65 -12.5)	τ 18 34 %
Apollo-Soyuz	SAM	1975	1975	0.83	-
GMS-1	VISSR	1977	2005	4(0.45 - 12.5)	-
TIROS-N	AVHRR	1978	1980	4(0.58 - 11.5)	-
Nimbus-7	SAM-2 CZCS TOMS	1978	1993	1 6(0.441-11.5) 6(0.385 - 0.380)	$\sigma_e xt$ (10%)
AEM-B	SAGE	1979	1981	4(0.45 -12.5)	-
NOAA 6-16	AVHRR	1979	Present	5(0.58 -12)	τ 10 % , τ 3.6 %
ERBS	SAGE-2	1984	2005	4(0.386 -1.02)	$\sigma_e xt$ (10%)
TRMM	VIRS	1997	Present	5(0.63 - 12)	τ 35 %, α (+- 0.5)
SPOT-3	POAM-2	1991	1996	9(0.353-1.060)	$\sigma_e xt$ (20%)
ERS-1	ATSR GOME	1991	1999	4(1.6, 3.7, 11,12) 4 (0.24-0.79)	-
UARS	HALOE	1992	2005	8(2.45 - 10.01)	$r_e ff$ (+- 15 %), $\sigma_e xt$ (5%)
SSD	LITE	1994	1994	3(0.355, 0.532, 1.064)	$\beta(\lambda_1)$ / $\beta(\lambda_2)$ (<5%)
ERS-2	ATSR-2 GOME	1995	Present	7(0.55 -12) 0.24-0.79	τ (<0.03), τ (30)%
Earth Probe	TOMS	1996	Present	6(0.309-0.360)	τ (20-30%)
ADEOS	POLDER ILAS OCTS	1996	1997	9(0.443-0.910) 6(0.441-11.5) 2(0.75-0.78 6.21-11.77) 7(0.412-0.865)	τ (20 -30 %)
Orb View -2	SeaWiFS	1997	Present	8(0.412-0.865)	T(5-10%)
SPOT -4	POAM-3	1998	Present	9(0.354-1.018)	$\sigma_e xt$ (+- 30%)
TERRA	MODIS MISR	1999	Present	36(0.4-14.4) 4(0.35 0.87)	τ (5 - 15 %) , τ (10 - 20 %)
METEOR-3M	SAGE-3	2001	2005	9(0.385-1.545)	$\sigma_e xt$ (5%) ,τ (5 %)
PROBA	CHRIS	2001	Present	62(0.4-1.05)	-
Odin	OSIRIS	2001	Present	0.274-0.810	$\sigma_e xt$ (15%)
AQUA	MODIS	2002	Present	-	-
ENVISAT	AATSR MERIS SCIAMACHY	2002	Present	7(0.55-12.0) 15(0.4-1.05) 0.24-2.4	τ (0.16), τ (0.2), AI(0.4)
ADEOS -2	POLDER-2 ILAS-2 GLI	2002	2003	9(0.441-0.910) 4(0.75-12.85) 36(0.38-12)	τ (0.1)
MSG – 1	SEVIRI	2002	Present	12(0.6-13.4)	τ (0.08)
ICEsat	GLAS	2003	2003-	2(0.532, 1.064)	$\sigma_e xt$ (10%), τ (20%)
AURA	OMI HIRDLS	2004	Present	3(0.27-0.5), 21(6-18)	τ (30%), $\sigma_e xt$ (5 25%)
PARASOL	POLDER-3	2004	Present	8(0.44-0.91)	-
CALIPSO	CALIOP	2006	Present	2(0.532,1.064)	-

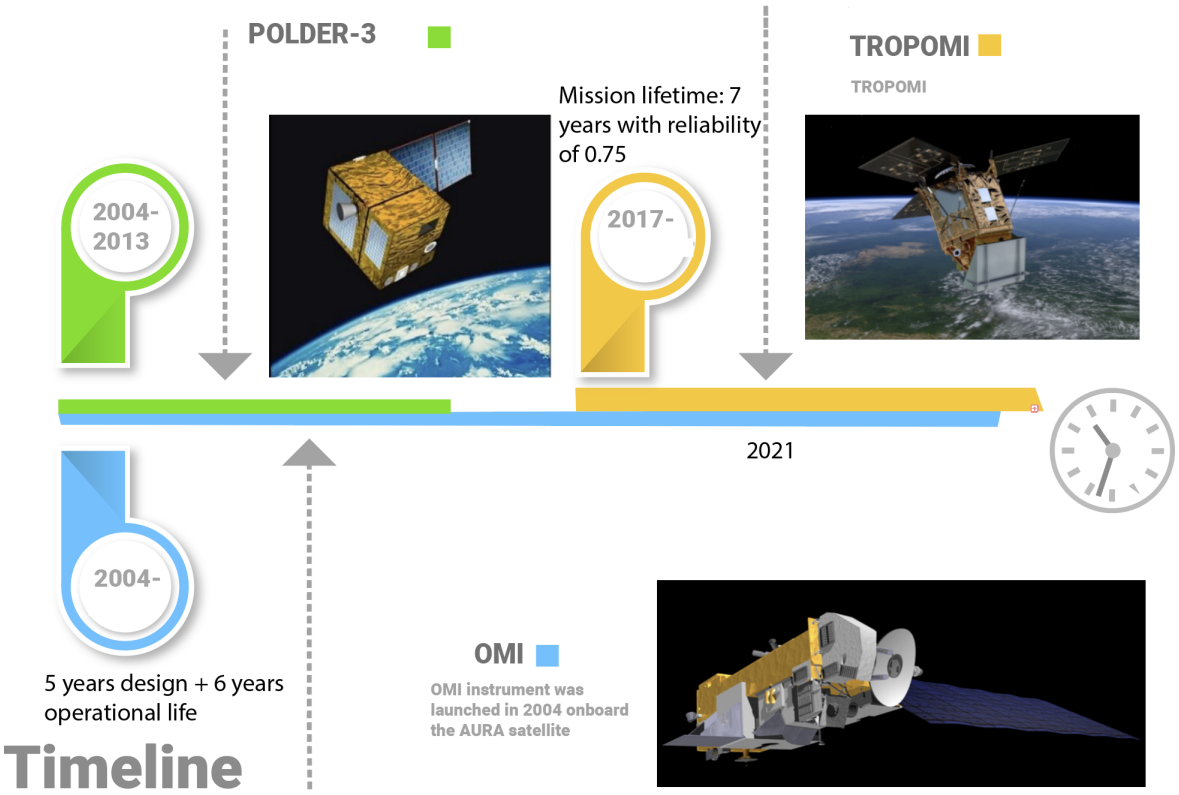
While there have been many missions in the past dedicated to the purpose of aerosol observation, only three missions shall be studied for the purpose of this thesis. The three being, POLDER-3, OMI, and TROPOMI. The reasoning being that we needed three missions from various timelines giving us a sense of past, present and future

missions. While POLDER-3 gives us a sense of past missions, TROPOMI gives use a picture of the atmospheric composition of the present and future. OMI has been chosen since it has an overlapping timeline with both POLDER-3 andTROPOMI thus giving us an opportunity to compare results between the various missions.

Table 3.2: Three aerosol measuring instruments at a glance: A comparison

Instrument	POLDER-3	OMI	TROPOMI
Satellite	PARASOL	AURA	SENTINEL-5-P
Launch Date	December 18, 2004	July 15, 2004	October 13, 2017
Instrument Type	Passive optical imaging radiometer polarimeter instrument	Nadir-viewing visible ultraviolet spectrometer	Spectrometer sensing ultraviolet(UV) visible (VIS) near (NIR) short-wavelength infrared (SWIR)
Resolution	6 km over 2400 km swath	2600 km	7 x 7 km best resolution
Swath	2400 km	2600 km	2600 km
Wavelength	443 and 910 nm FWHM	270 to 500 nm	UV 270-320 nm visible 310-500 nm NIR 675-775 nm SWIR 2305-2385 nm.[6]
Agency	CNES	NASA satellite Dutch Space instrument KNMI science output	ESA

Figure 3.2: Timeline of POLDER-3,OMI and TROPOMI Instruments



3.2. *Polarization and Directionality of the Earth's Reflectances* (POLDER)-3 (PARASOL)

POLDER-3 is an instrument onboard the PARASOL. Launched in December 2004, POLDER 3 has by far the most detailed sensor information regarding aerosols. It was the first of its kind sensor capable of making multi-angle polarization imaging observations in the world. It has a push broom scanner with a telecentric lens [64]. In section B.1, the detailed parameters that can be used to derive information regarding the aerosols are noted. Almost 55 variables present make POLDER-3 a highly suitable candidate to analyse atmospheric aerosol composition.

Furthermore, both OMI and PARASOL were on the same A train as can be seen from Figure 3.4. Thus, both were measuring the same area, making comparisons stronger. However, in December 2009, as is evident in Figure 3.5 PARASOL was moved to a lower orbit, and the syncing with the A train sensors only happened in 2 to 3 months. By 2011, PARASOL was out of view of the other A-train sensors at the equator. Unfortunately, PARASOL was decommissioned 9 years after launch on 18 December 2013, making the necessity to continue the present study with newer and functioning satellites like OMI and TROPOMI necessary. Nevertheless, POLDER 3 provides a detailed benchmark to study the global atmospheric composition of aerosols which proves to be a good starting point for the thesis.

It is interesting to note that POLDER has flown thrice on satellite missions. The first was with ADEOS-1 in 1996-1997 mission and the second with ADEOS-2 in 2002. Although, they could only gather a few months of data due to malfunction of the spacecraft solar panels [68]. POLDER-3 in contrast gave nearly 9 years worth of scientific output.

Figure 3.3: POLDER-3 onboard the PARASOL satellite

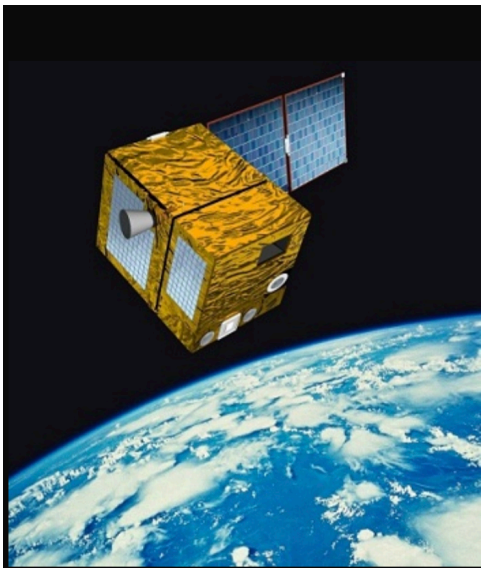


Table 3.3: POLDER-3 data parameters [16]

Abbreviation	Full name	Wavelengths / Modes / Variants
AOT	Aerosol Optical Thickness	440, 490, 563, 670, 865, 1020 nm
SSA	Single Scattering Albedo	440, 490, 563, 670, 865, 1020 nm
reff	Effective Radius	Fine, Coarse Mode Fraction
veff	Effective Variance	Fine, Coarse Mode Fraction
m_r	Real Refractive Index	Fine, Coarse Mode Fraction
m_i	Imaginary Refractive Index	Fine, Coarse Mode Fraction
sphere_frac	Sphericity	Fine, Coarse Mode Fraction
lat	Latitude	Coordinates of Centers and Corners
long	Longitide	Coordinates of Centers and Corners
psurf	Surface Pressure	
N	Aerosol Column Number Density	Fine, Coarse Mode Action
number_of_points	Number of Data Points	Over Ocean, Land
error	Parameter Retreival Uncertainity / Error	AOT, SSA, reff, veff, m_r, m_i, N, sphere_frac

Figure 3.4: Parasol and POLDER-3 instrument in the A train with Aura withOMIinstrument [16]

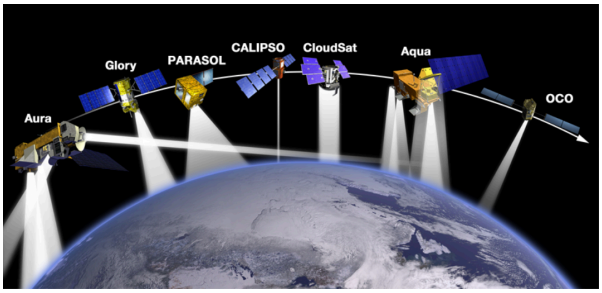
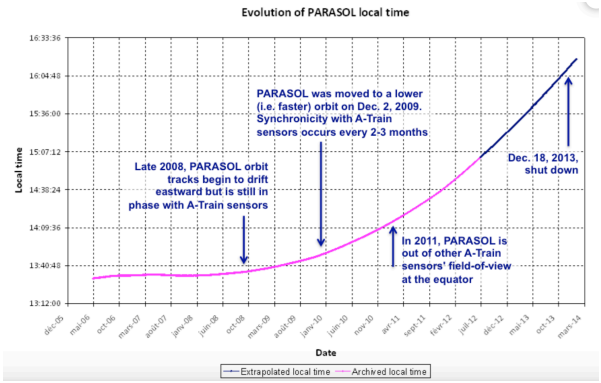


Figure 3.5: Parasol lifetime [16]



The POLDER-3 sensor comprises of a digital camera with a CCD detector array of 274 x 242 pixels. It is capable of measuring in nine spectral channels ranging from blue (0.443 m) to near-infrared (1.020 m), as well as polarization at 0.490 m, 0.670 m, and 0.865 m and at up to 16 distinct angles (51° along track, 43° across track). At nadir, the pixel size is 5.3 km 6.2 km. The POLDER-3 overpass occurs at approximately 1:30 p.m. local time, and one scene of photos covers an area of 2100 x 1600 km2. POLDER-3 enables two-day global coverage. The equipment measures polarized light in many directions, which enables the extraction of more precise aerosol optical and physical properties [64][20].

3.3. *Ozone Monitoring Instrument* (OMI)

The Total Ozone Mapping Spectrometer (TOMS) instrument of NASA and the Global Ozone Monitoring Experiment (GOME) instrument of the European Space Agency (ESA) are the forerunners ofOMI(on the ERS-2 satellite). It has a considerably higher ground resolution than GOME and can measure a significantly more atmospheric components than TOMS (13 km x 25 km forOMI vs. 40 km x 320 km for GOME).OMI is a major instrument aboard Aura for monitoring the recovery of the ozone layer in response to the phase-out of substances such as CFCs, which was agreed upon by the world's governments in the Montreal Protocol and subsequent revisions in Copenhagen and London.

Criteria pollutants including O3, NO2, SO2, and aerosols are measured by OMI. The US *Environmental Protection Agency* (EPA) has identified these areas as hazardous to the environment. OMI is 100 times more sensitive than TOMS at detecting volcanic ash and sulfur dioxide generated by volcanic eruptions. The accuracy of these measures is critical for aviation safety.OMI monitors ozone profiles (in the UV) in addition to TES and HIRDLS (in the IR) and MLS (in the IR) (in the microwave).OMI measures BrO, formaldehyde, and OClO, all of which are involved in the chemistry of the atmosphere.OMI onboard the Aura satellite platform was launched on July 15, 2004 [65].

Figure 3.6: OMI onboard the Aura satellite

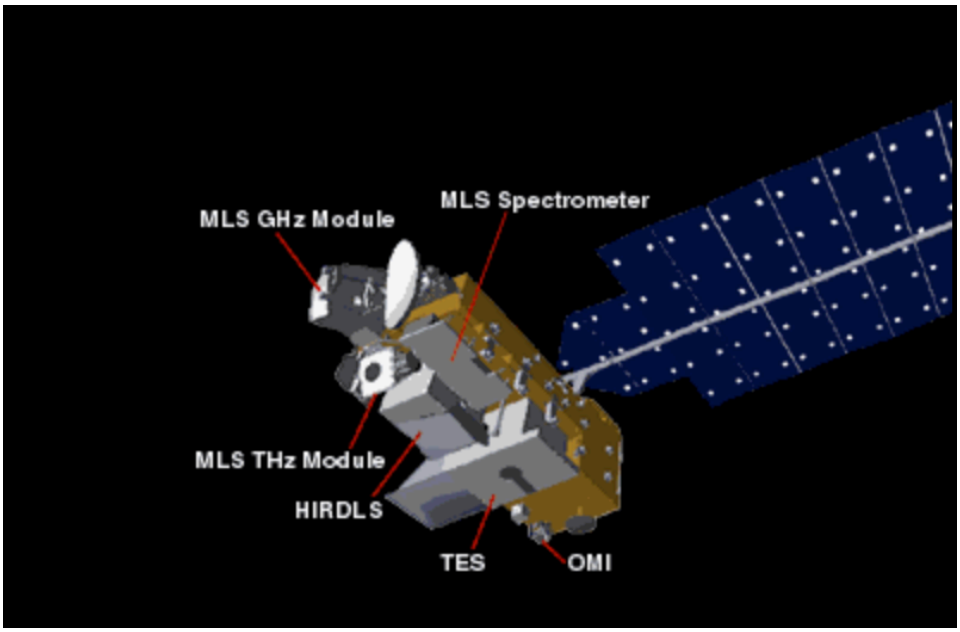


Table 3.4: OMI instrument specifications [65]

Item	Parameter
Visible:	350 - 500 nm
UV:	UV-1, 270 to 314 nm, UV-2 , 306 to 380 nm
Spectral resolution:	1.0 - 0.45 nm FWHM
Spectral sampling:	2-3 for FWHM
Telescope FOV:	114 (2600 km on ground)
IFOV:	3 km, binned to 13 x 24 km
Detector:	CCD: 780 x 576 (spectral x spatial) pixels
Mass:	65kg
Duty cycle:	60 minutes on daylight side
Power:	66 watts
Data rate:	0.8 Mbps (average)
Pointing requirements (arcseconds)	(Platform+instrument, pitch:roll: yaw, 3s):
Accuracy:	866:866:866
Knowledge:	87:87:87
Stability (6 sec):	87:87:87
Physical Size:	50 x 40 x 35 cm

With a wide-field telescope feeding two image grating spectrometers, the equipment examines Earth's backscattered radiation. A CCD detector is used in each spectrometer.
A white light source, LEDs, and a multi-surface solar-calibration diffuser are all included in the onboard calibration. The polarization of backscattered radiation is removed using a depolarizer.

3.4. Chapter Summary

A brief historical context to the satellite missions dedicated either fully or partially to the study of atmospheric aerosols is presented. Out of the various missions flown, three missions TROPOMI, OMI and POLDER-3 are selected due to their differing timelines that allow overlap to measure and compare the science output. Furthermore, the quality of the generated output also played a role in selecting these mission. In additions, all of the three satellite missions have a connection with the Dutch space industry, either through data processing organizations like *Netherlands Institute for Space Research* (SRON) / *Royal Netherlands Meteorological Institute Koninklijk Nederlands Meteorologisch Instituut* (KNMI) or through satellite manufacturing companies like Airbus Defence and Space Netherlands B.V. (previously Dutch Space).

4

Supervised Learning Algorithms

In this Chapter, we look at the theory and the studies related to the three supervised learning algorithms KNN, RF and SVM implemented in this thesis. Supervised learning is a subset of *Machine Learning* (ML). ML techniques are used to automatically build an empirical model from available information [36]. Typically, we may perform either supervised or unsupervised learning when using ML methods. Supervised Learning requires ground truth labels or data toward training capable prediction-making models, while unsupervised learning approach seeks to determine emergent patterns within input data [15] [34]. ML as we know it today was conceived back in 1959 by Arthur Samuel [56], two years after the first satellite Sputnik was launched. Although the application of ML was intended for pattern recognition for the computer industry, in the recent years ML is proving to be ubiquitous across multiple fields including satellite data retrievals which is the focus area of this thesis.

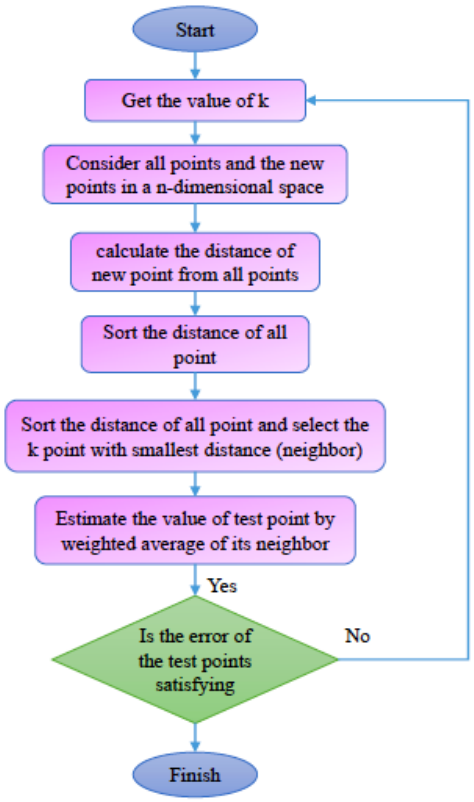
4.1. *K Nearest Neighbours* (KNN)

The basic principle of k nearest neighbour or KNN method is to cluster the input data on the basis of distance metric (e.g. Eucleadian, Manhattan, Minkowski, Cosine, Jaccard and Hamming with Eucleadian being the most popular in Scikit-learn package in python). Then, the new data is classified based on its proximity to existing groupings. Here, k represents the determination of parameter like how many adjacent data, i.e. neighbors are required for classification before assigning the clusters label to input data. As noted in the research conducted by D. Crowe et al [15], the primary advantages of KNN are that the model training and prediction are completed at a rapid pace. This ensures that KNN can be used as a near-real time and resource-efficient algorithm.

KNN is split into two different phases namely training and classification. In training phase, the number of objects are classified manually by human beings from the training set. Here the class labels and feature vectors are stored then the computer reads this subset of objects. For these objects the correct classification is known. During the process of classification phase, the input data are categorized by majority of nearest neighbors' value (by measuring distance among variable). Usually, the training phase is executed once, and the classification phase is executed any number of times afterwards.

From Figure4.1, we can see the flowchart for the KNN algorithm. First, we get the value for the parameter k which defines the number of nearest neighbours. Next, we take all the datapoints and the new points in a n-dimensional space. This is followed by calculating the distance of the new point from all the datapoints. Subsequently, we sort the distance of all the datapoints and select the k point with the smallest distance. We then estimate the value of the test point by weighted average of its neighbour. The algorithm ends when the error of the test points is within the desirable range.

Figure 4.1: KNN Algorithm flowchart [41]



In Figure 4.2, we can clearly demonstrate the difference the selection of the K parameter has on the classification of the input into output class. In the figure, the input sample is the green circle that has to be classified into either the class (red triangle) or class (blue square). For k =3, There are two triangles and only one square inside the inner circle, and thus, the green circle is classified to red triangle. If k=5 it is classified as blue square, due to the larger number of samples in the area specified by the dotted lines. K is commonly taken to be an odd number if the number of output classes is two to avoid overlapping. The selection of small and large k value are used to classify the data in an efficient manner. Each of these value specifications are summarized in Table 4.1. In a nutshell, the best value of k depends on the actual data, as there is no one size fits all solution. Every training sample is traversed and the distance d between the training set sample and the new sample is computed.

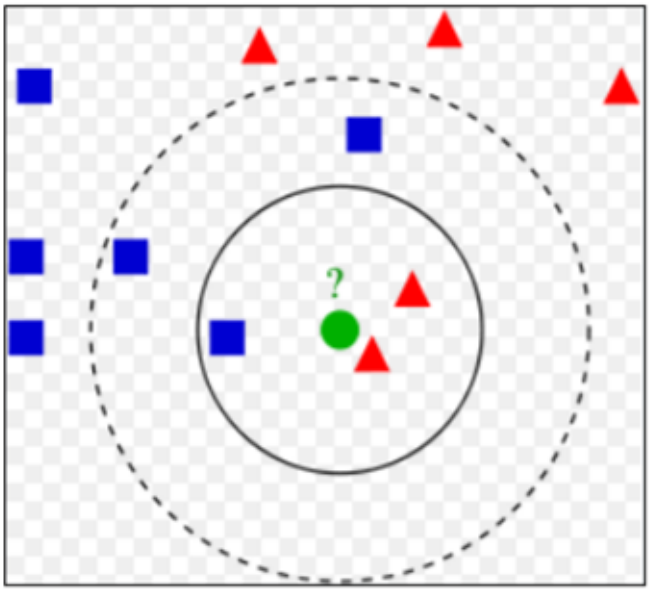


Figure 4.2: Example of KNN classification [31]

Training time Complexity is equal to:

$$O(n_{features} * n_{samples} * k) \tag{4.1}$$

where n refers to the number of training samples and K is classifier parameter referring to the number of nearest neighbours to include in the majority of the voting process.

Table 4.1: Features of small and large values of k [71]

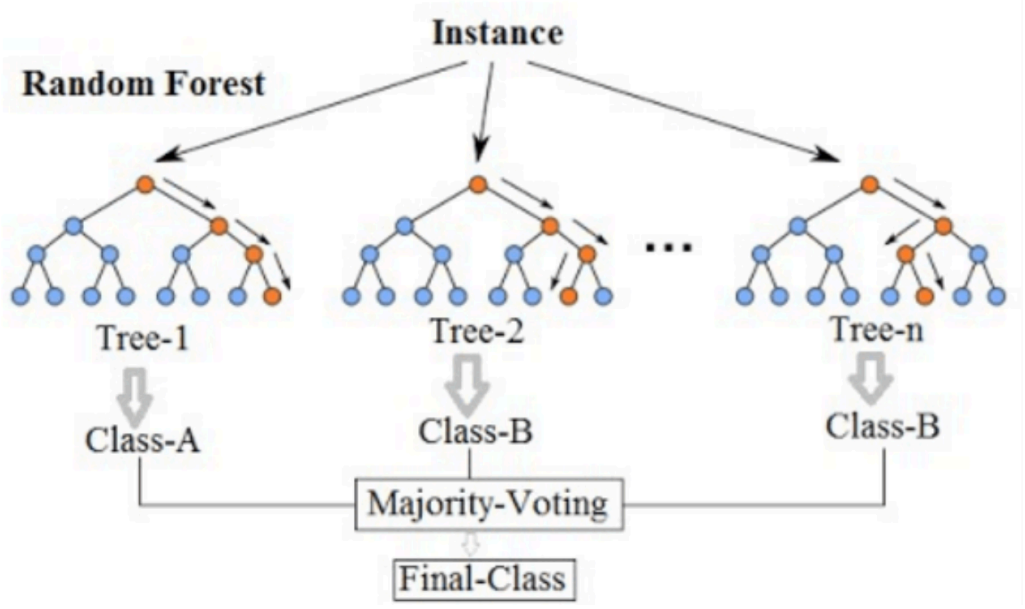
Small values of k	**Large values of k**
Cause over-fit	Cause over-generalization
Increase negative effect of noise	Reduce negative effect of noise
Create distinct class boundaries	Create indistinct class boundaries

A work by Nikolaos Papagiannopoulos et al [50] introduced a programmed airborne order technique dependent on the *European Aerosol Research Lidar Network* (EARLINET) escalated optical boundaries with the point of building an organization wide arrangement apparatus that could give close continuous vaporized composing data. The introduced strategy relies upon a directed learning method and utilizes the Mahalanobis separation work that relates each unclassified estimation to a predefined airborne sort. As an initial step (preparing stage), a reference dataset is set up comprising of previously ordered EARLINET information. Utilizing this dataset, they characterized 8 airborne classes: clean mainland, contaminated mainland, dust, blended residue, dirtied dust, blended marine, smoke, and volcanic debris. The impact of the quantity of vaporized classes has been investigated, just as the ideal arrangement of concentrated boundaries to isolate distinctive airborne sorts. Besides, the calculation is prepared with writing molecule straight depolarization proportion esteems. As a subsequent advance (testing stage), they applied the strategy to a previously arranged EARLINET dataset and examine the consequences of the correlation with this grouped dataset. The prescient precision of the programmed characterization shifts between 59 % (least) and 90 % (greatest) from 8 to 4 vaporized classes, separately, when assessed against pre-grouped EARLINET lidar. This demonstrates the expected utilization of the programmed order to all organize lidar information. Moreover, the preparation of the calculation with molecule straight depolarization esteems found in the writing further improves the precision with values for all the vaporized classes around 80 %. Also, the calculation has demonstrated to be exceptionally flexible as it adjusts to changes in the size of the preparation dataset and the quantity of airborne classes and arranging boundaries. Finally, the low computational time and interest for assets make the calculation incredibly reasonable for the execution inside the single analytics chain, the EARLINET brought together preparing suite.

4.2. *Random Forest* (RF)

As shown in Figure4.3, RF classifier is a kind of a machine learning based ensemble approach which is a combination of output and collection of decision tree of individual trees. The random forest is a type of decision tree method that minimizes the variance through averaging the the unbiased decision trees [25]. The bagging approach is comparable to the boosting method which combine different weaker rules rather than using single decision rule. However, bagging takes an average value and boosting method produces a sequence of weak decision rules [4] . In order to predict aerosol type, trees within the ensemble method we can describe as vote on classification label. The spectrum is allocated to majority choices whereas each tree has equal weight ratio. Commonly voted label implies higher certainty since the ensemble approach is independent to the subset of training data. by providing alternative hypotheses we can add ensemble method that increases the robustness of classification approach so preferable to unique classifiers [11].

Figure 4.3: Random Forest Classifier [13]



Structure of Random Forest Classification

Training time Complexity is equal to:

$$O(n_{samples} * log(n_{samples}) * k) \tag{4.2}$$

where k is the number of trees in the Random Forest.
Run time Complexity is equal to:

$$O(depthoftree * k) \tag{4.3}$$

Space Complexity is equal to:

$$O(depthoftree * k) \tag{4.4}$$

Random Forest is supposedly faster than other algorithms.
Run time Complexity is equal to

$$O(n_{features} * n_{samples}) \tag{4.5}$$

Testing takes longer since each test instance is compared to the entire training data. According to a study by Wonei Choi et al. [10], a new method for categorizing aerosol types was created using satellite data and a machine-learning methodology. In a RF model, an aerosol type dataset from the *Aerosol Robotic Network* (AERONET)

was employed as a target variable. To find the best collection of input variables for the RF-based model, the contributions of satellite input variables were quantified. Based on inputs from satellite variables, the new technique can classify seven different types of aerosols: pure dust, dust-dominant mixed, pollution-dominant mixed aerosols, and pollution aerosols (strongly, moderately, weakly, and non-absorbing). The model's performance was statistically tested using AERONET data that was not included in the model training dataset. The accuracy of the model in classifying the seven categories of aerosol was 59 percent, with 72 percent for four types (pure dust, dust-dominant mixed, strongly absorbing, and non-absorbing). The model's performance was compared to that of a previous aerosol classification approach based on the wavelength dependence of SSA and AERONET fine-mode-fraction data. SSA wavelength dependencies for specific aerosol types are consistent with those found using the new approach for the same aerosol types. This research shows that an RF-based model can classify satellite aerosols while being sensitive to non-spherical particle contributions.

A work by Christopoulos et al [11] developed a model using random forest method on single particle mass spectrometry data. They have specifically focused on reducing the dimensionality and evaluated the performance with the help of confusion matrices. Additionally, they have identified the significance of chemical markers among sources of contamination or arbitrary groups of aerosols using chemical feature selection method. The ranking was done using dimensionality reduction method and frequently identified the contaminant via subset of ranked features.

4.3. *Support Vector Machine* (SVM)

In order to perform classification and regression analysis, the SVM method utilizes kernel function through transfer of data into high dimensional space with the help of non-linear transformation approach. It may also find the linear space among variables by separating input data into two classes. Hence the hyper-plane is the greatest margin among two classes. The Figure illustrates SVM based hyperplane among two classes whereas the bold lines denote the hyper-plane that split the data into two classes. This separation process is used to obtain maximum separation between two classes with minimal error. In SVM, the inputs are represented in the form of attribute vectors [39].

Due to the non-linear and higher dimensionality characteristics, the process of hyperspectral remote sensing data becomes more complex SVM. For addressing this issue, a research by Kolluru in 2013 [33] recommended ML based SVMs. This was mainly preferred to perform classification of high dimensional data. In order to perform classification and regression analysis, the SVM method utilizes kernel function through transfer of data into high dimensional space with the help of non-linear transformation approach. It may also find the linear space among variables by separating input data into two classes. Hence the hyper-plane is the greatest margin among two classes. Figure4.4 illustrates SVM based hyperplane among two classes whereas the bold lines denote the hyper-plane that split the data into two classes. This separation process is used to obtain maximum separation between two classes with minimal error. In SVM, the inputs are represented in the form of attribute vectors
.

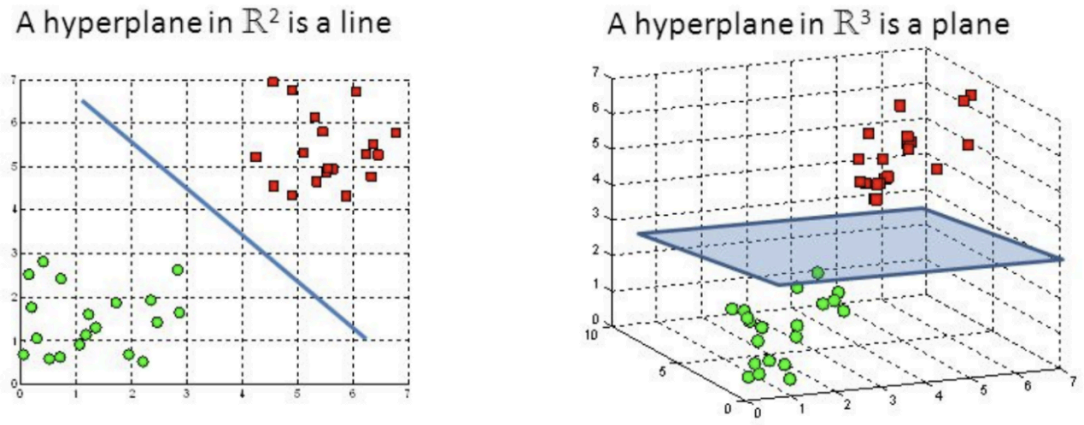


Figure 4.4: Visualizing the hyperplane separating the two feature vectors of classes in 2D and 3D plane [1]

The two planes parallel to the classifier and which passes through one or more points in the data are called 'bounding planes. The distance between these bounding planes is called 'margin'. By the process of learning hyperplane which maximizes, this margin is evaluated. The points of the corresponding class, which falls on the bounding planes, are called 'support vectors. These points are crucial in forming a hyperplane hence the name support vector machine [57].The support vector machines predicts the target values of test attributes based on the training data [29].

In the following paragraphs, studies using SVM classifier for aerosol classification are discussed. For instance, a study by Ma and Gong, (2012) [40] presented an SVM classification for aerosol and cloud. With the help of the SVM classifier, they have confirmed the suitable parameter that includes the attenuated backscatter, depolarization ratio, latitude of a base layer, colour ratio, and layer top in each layer, then avoided the cloud attenuated backscatter coefficients based multi-modal distribution during the simulation process. Also, they have validated the system performance by considering different sample sizes and various feature space/vectors, where the feature space includes depolarization ratio which provides stable results, then frequently increases the number of samples with better accuracy. Research by Lary et al., 2018 [36] has discussed the relation of AOD measured by the AERONET and retrieved through MODIS. But the bias between these two sets of data might be an imperfect understanding of root causes. In order to overfit the training examples, the resulted outcome of various learning techniques may accurately classify the training dataset however, it fails to simplify new examples. To minimize the statistical bound on the generalization error, SVM is specifically made, resulting in outcome as efficiently extrapolate a new example. When compared to the neural network, SVMgets trained faster by adopting a randomized procedure which also follows the deterministic quadratic optimization process. In order to lead to a slow evaluation of new data, the downside of SVM models can be large. To overcome these issues, different optimization methods must be developed for making SVMs faster, especially for operational use [44].

5

Methodology

This chapter goes into the details of the methodology and implementation that will be followed in this thesis project. In section 5.1, the research work plan is discussed. The next section describes how the data was collected. This is followed by discussion in section 5.2 on the machine learning algorithms and motivation for the choice of using SVM, RF and KNN algorithms. Following this, we discuss the implementation considerations for these algorithms in Scikit-learn[51] and discuss the various parameters that need to be selected. Subsequently, the parameters and metrics required to evaluate these models are discussed in section 5.3. Section 5.4 contains the high level requirements guiding the implementation of this thesis project. Thereafter, the environment used for implementation of the code is discussed in section 5.5. Section 5.6 discusses the implementation of thesis work first for POLDER-3 data and then followed by OMI data.

5.1. Research Work flow

The research was split into four phases. The first part of the research phase delved into the literature study of the presently available algorithms, briefly documented in Chapter 4. This phase was completed with the selection of the three algorithms for aerosol classification, i.e. SVM, KNN and RF based on their merits and demerits compared to other algorithms as mentioned in Table 5.2. The literature study gave a background into the available machine learning algorithms to be implemented in the thesis. Furthermore, the satellite data concerning aerosol retrievals were studied and three satellite missions were selected. Out of these, POLDER-3 and OMI for the year 2006 were studied in detail.

The second phase consisted of obtaining the satellite data from SRON and NASA sources for the reference year 2006. The year 2006 was chosen since that was the dataset shared by Otto Hasekamp and this is the year on which the clustering of aerosols was performed by Vincent de Bakker [16]. This was followed by exploring the data and understanding the format and extracting the data in a suitable format such that further analysis could be made. The next step, was to understand the variables and clean the data to remove NaNs.

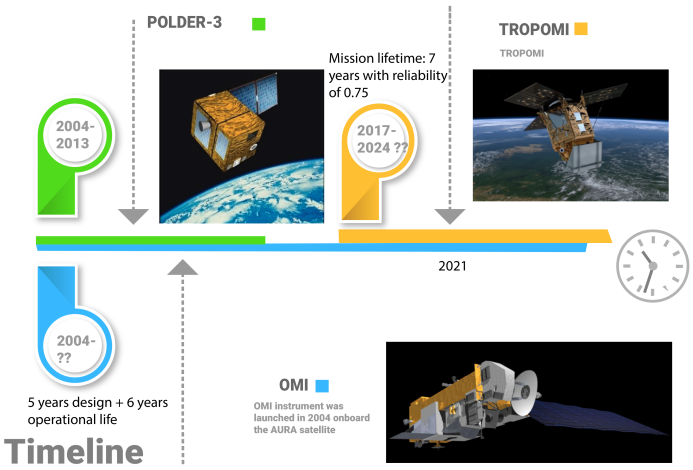
The third phase of implementation began with obtaining the unsupervised learning cluster labels from previous researchers. To obtain the class label for POLDER-3, the research of Vincent de Bakker [16] was studied, understood and recreated. Unsupervised learning techniques SOM and K means were run in R studio to get these labels and the data was stored in .RDA format. This .RDA format was ported to python to implement the supervised learning techniques. Statistical software R was used to run all the simulations for unsupervised learning techniques to create the cluster labels for POLDER-3 to be used as a starting point for running supervised learning algorithms selected in the second phase. Python 3 machine learning environment (SCI-Py) was selected to implement these algorithms. The data was split in 70 % training and 30 % testing format. Further, seven fold cross-validation scheme was used to ensure that the model did not overfit the data. Once the best performing algorithm was selected based on the root mean square error across the class labels, the next step was to tune the hyperparameters to increase the efficiency of the algorithm. Here, we use bias and variance trade-off to find the optimal hyperparameters and then use the confusion matrix to derive insights into these results. The same steps were repeated for OMI accounting for the changes compared to POLDER. Firstly, OMI data has daily files while POLDER-3 has monthly files. Secondly, OMI has inherent aerosol classes present with the data products and

thus we did not need to recreate unsupervised learning to create these class labels. However, OMI had only three class labels compared to eights clusters of POLDER-3. Furthermore, POLDER-3 has more variables describing aerosols than OMI, however it is no longer active and thus necessitates looking at other satellites as an alternative for present and future data like TROPOMI.

In the fourth phase, the results obtained in the third phase are analysed and compiled together while noting the similarities and differences.

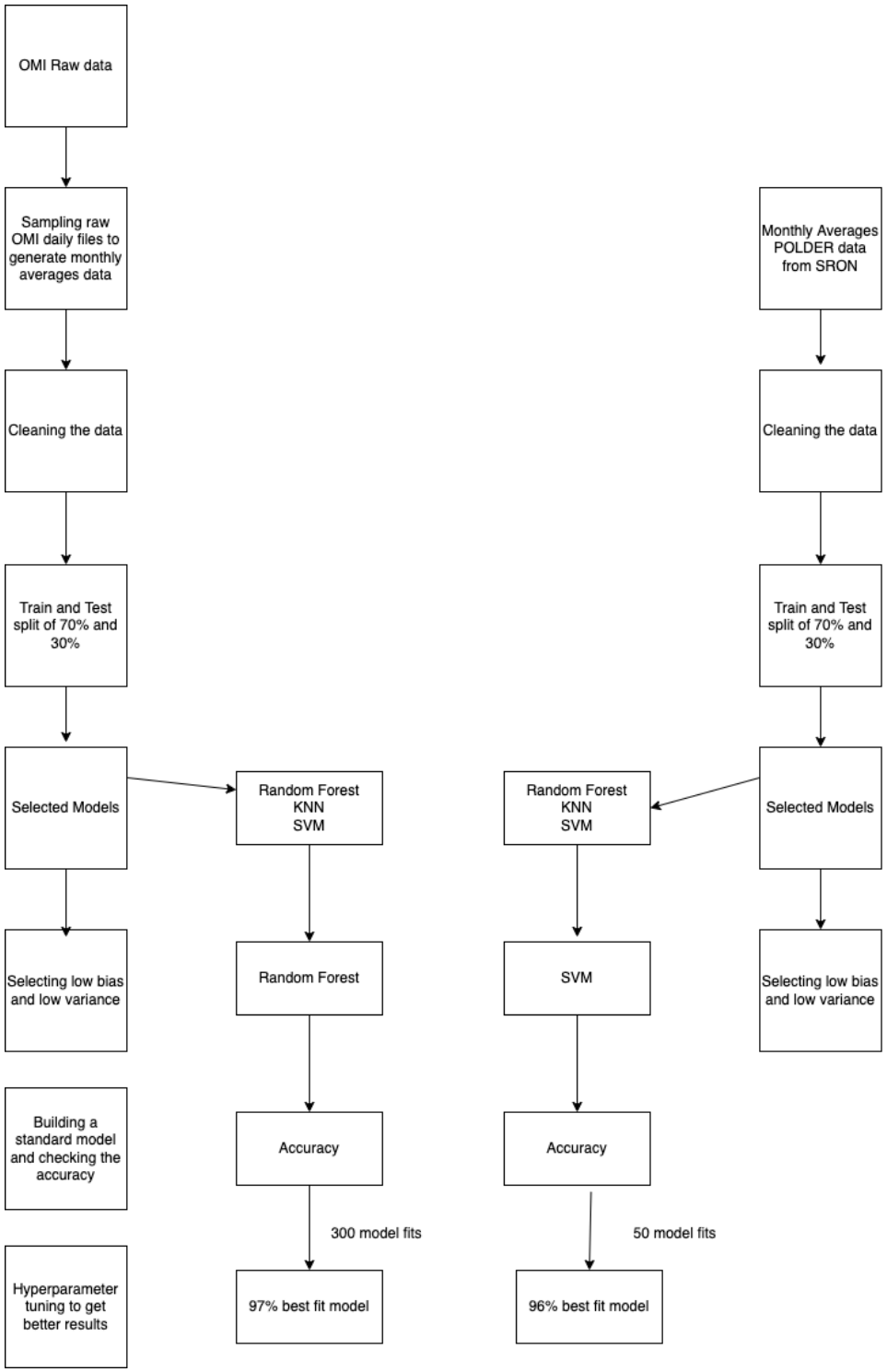
As mentioned in Chapter 3, two satellite missions i.e. POLDER-3 and OMI were chosen to be studied and analysed using machine learning for this thesis as seen in Figure 5.1. Figure 5.2 shows the outline of the methodology used in this thesis. On the right hand side the pipeline for POLDER-3 is detailed and on the left hand side that for OMI. Firstly, these were chosen for the microphysical parameter information contained in the satellite data that could help effectively identify the atmospheric aerosol composition. Secondly, these three missions contain overlapping information that could help in providing a continuous stream of satellite information. POLDER-3 is the oldest instrument out of the three missions, other being OMI and TROPOMI, but also the one with the most detailed microphysical parameters regarding the aerosols. POLDER-3 data was obtained from SRON and was presented in a monthly average fashion while that for OMI was downloaded from the NASA repository. Even though the location and the period were the same, the features were different between the OMI and POLDER-3 satellite and thus it results in no direct similarity between the data. TROPOMI data was obtained from ESA copernicus programme and the data was plotted. Further work on TROPOMI has been proposed in future work keeping in mind the time constraints and the fact that aerosol related parameters like the aerosol layer height variables will be made available in the future.

Figure 5.1: Timeline of POLDER-3, OMI and TROPOMI Instruments. In this study we explore the data for the year 2006 for POLDER-3 and OMI. TROPOMI is recommended as future work.



One of the first steps in the thesis was to recreate the research done by Vincent de Bakker [16] to perform the clustering on the POLDER-3 2006 data to generate the labels of aerosols. Research by Vincent de Bakker [16] created a database of clusters on the POLDER-3 2006 dataset on the world maps, where each cluster corresponded to an aerosol type. He classified these into eight aerosol types: dust, marine, urban-industrial, smoke and the remaining four types are mixtures of these above mentioned classes. The study used unsupervised learning, in the form of Self Organizing Map and K-means clustering algorithm to create labels for each geographical latitude, longitude combination. These results are the starting point in our thesis. We use these labels as ground truth labels to train our supervised learning algorithm. The focus of our study is to see how the three supervised learning algorithms RF, KNN and SVM perform against each other on the two satellite data POLDER-3 and OMI.

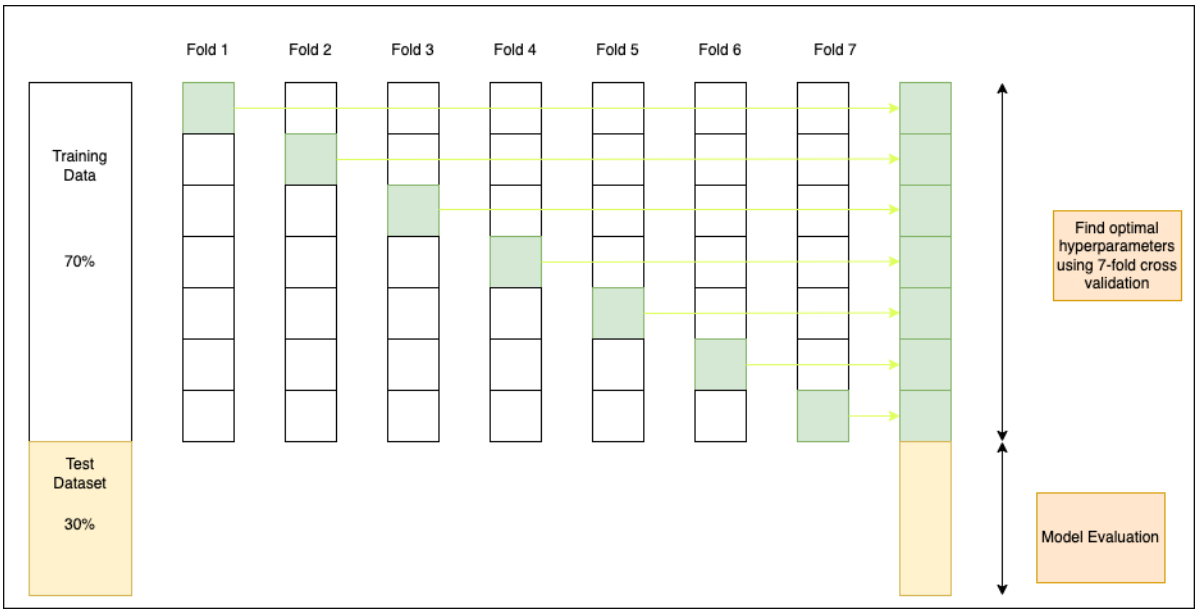
Figure 5.2: Methodology Outline



The next step is to look at the obtained data. First, we look at the POLDER-3 data for the year 2006. We extract the data frame with micro-physical properties for atmospheric columns with a seasonal label. The year is split into four seasons: Winter, Spring, Summer and Autumn. Next, we plot the microphysical property distribution across the four seasons to view the seasonality and the data distribution. Subsequently, we perform data exploration on the data to see the four variables and their cross correlation with other variables to check how much each variable influences each other. With the eight class labels as mentioned in Table 6.2 as the target label, we use cross-validation mentioned in 5.3, to train the model and measure the algorithm performance.

Second, we begin by looking at OMI data for the year 2006 as well. We extract the required micro-physical parameter data from the he5 format along with the timestamp, the latitudinal and longitudinal data. Furthermore, we have the four class labels classifying the aerosol type which we use in training the data. Again, we follow the same methodology as POLDER-3 to find the best performing algorithm.
With the labelled data as our input set, we make a split of 70 % training data and 30 % test data to measure the performance of our model. Cross-validation is a resampling process used to evaluate machine learning models on a limited data sample. The process has a single parameter called k that refers to the number of groups that a given data sample is to be split into. This procedure is often called k-fold cross-validation. We use seven fold cross-validation scheme to ensure that there is no high bias in the data. It ensures that every data point from the dataset has the chance of appearing in the training and test set. It is one of the recommended approaches for a limited input data since it creates a large training data by iterating through the data. In Figure 5.3, we see k-fold validation where one fold is used for testing and k-1 fold is used for model training. This process is repeated k times. The value of k is recommended to be between 5 to 10. In this study we employ k as 7. K is generally chosen that is a divisor of the sample size in our case, or the size of the groups in the sample that should be stratified.

Figure 5.3: Splitting the data into test and train sets using the cross validation approach to prevent over-fitting of the data



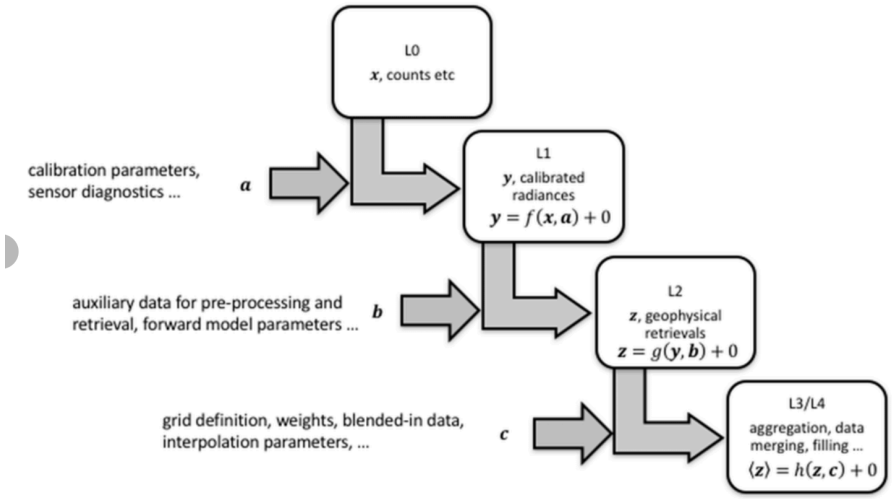
Using this cross validation scheme we train three models. To measure the performance of the three algorithms (RF, KNN, SVM) against one another and to select the best performing algorithm for each use case we use the performance metric of Root Mean Square Error.
The importance and the relevance of understanding aerosols and their classification was mentioned in Chapter 2. In a nutshell, through the study we can comment on how the differing satellite missions measure the aerosols differently and measure how the three machine learning algorithms stack up against each other when the data sets are changed. We can then finally conclude the study with a list of future recommendations for steps that might make the algorithm perform better but were out of scope of this study.

For future work, we look at the TROPOMI data. We extracted the required micro-physical parameter data from the netcdf format along with the timestamp, the latitudinal and longitudinal data. Next, the extracted data was plotted on a world map. Due to time limitations further work on TROPOMI is recommended for future work.

5.2. Data Collection

For the thesis, two satellite data had to be acquired. These included POLDER-3 and OMI. The data collected from the two satellites are at different processing levels. The Figure 5.4 shows the various available data processing levels and the relationship between them.

Figure 5.4: Relationship between different levels of satellite data [41]



NASA states on their website [1] that their *Earth Observing System Data and Information System* (EOSDIS) data products have four levels. Level 0 denotes the raw data from the instrument. Level 1 corresponds to reconstructed, time-referenced level 0 data. Level 2 includes further processing and includes derived geophysical variables at the same resolution. Level 3 data includes more consistency and completeness. The variables from the satellite data are mapped on a uniform space-time grid scales. Level 4 is highly processed data. It might also include results from multiple measurements resulting in better derived analyses. OMI is a NASA satellite and was provided in Level 3 daily gridded format while POLDER-3. *European Space Agency* (ESA) on their website also states that it have four levels of data processing [2]. Level 0 data is the raw data from the satellite. In level 1 data, the data is converted to physical units. In level 2 data the data is processed as results of the experiment of the instrument. An example of level 4 data are the weather maps that are presented on news.
In Table 5.1, we can see how the POLDER-3 and OMI data were sourced from different sources. POLDER-3 was sourced from scientist Otto Hasekamp from SRON while OMI was sourced from the open source satellite data website of NASA. It is worth noting that POLDER-3 was provided in level 3 monthly averaged gridded format while OMI was provided in level 3 daily gridded format which later had to be averaged to create monthly averages. Upon speaking with Dr. Pepijn Veefkind who was the principal investigator of TROPOMI and OMI, HARP `https://stcorp.github.io/harp/doc/html/ingestions/index.html` package was recommended to create these monthly averages.

Table 5.1: Data collection for POLDER-3 and OMI satellite data

Satellite Data	Organization	Link to data	Frequency	Year
POLDER-3	SRON	Obtained from Otto Hasekamp from SRON in the folder PARASOLDATA_GRIDDED	L3: Monthly Averaged Data	2006
OMI	NASA	OMI Science Investigator-led Processing System	L3: Daily averaged readings	2006

The OMI data has been obtained from NASA's Goddard Earth Sciences Data and Information Services Center. Since the instrument was a Dutch - Finish instrument which flew on an American satellite, the data is available both through the Goddard Earth Sciences Data Information Services Center site as well as from the KNMI website which is a Dutch Meteorological website. The OMI data was downloaded from `https://disc.gsfc.nasa.gov/data-access`. However, to access the data from this site a user account must be created on Earthdata website `https://disc.gsfc.nasa.gov/data-access`. Using the wget function on the Mac/linux PC the required L3

[1] `https://earthdata.nasa.gov/collaborate/open-data-services-and-software/data-information-policy/data-levels`
[2] `https://pmtp.hb.se/space-data/space-project-phasing-data-levels-and-data-use/processing-levels/`

files can be downloaded. The region of interest and the target dates can be selected and filtered through the website which then generates a list of downloadable files in He5 format that can be used for further processing. The date range input to select the data for the year 2006 to be compared with the POLDER 2006 data was 2006-01-01 to 2006-12-31. This results in approximately 5256 links for download relating to L2 OMAERO data.

The OMAERO corresponds to Aerosol multi-wavelength algorithm. OMAERO is based on a multi-wavelength algorithm that utilizes up to twenty wavelength bands spanning from 331 and to 500 nm. The first release data of the data was 23 November 2007. The files are available in level0, level 1, level 2 and level3 formats. Each Level 2 he5 file contains data for a single orbit. For each day there was 14 passes resulting in 14 L2 files. An L3 format was generated taking the best measurements of the day generating a single file for each day. For the purpose of brevity, the concise L3 files will be used. 363 files, 2.3G were present in the 2006 year data. The entire dataset could be downloaded in 33 mins. OMI data has the following parameters: Here d refers to the 1 degree by 1 degree resolution while 3e refers to the 0.25 by 0.25 degree resolution. OMAERUVd - OMI/Aura Near-UV Aerosol Absorption and Extinction Optical Depth And Single Scattering Albedo This product is used for accurate readings over the land. OMAEROe - OMI/Aura Aerosol Extinction Optical Depth and Aerosol Types product is used for accurate reading over the ocean surface.

5.3. Machine Learning Algorithms

In supervised machine learning, an algorithm learns a model from the training data. The aim of the algorithm is to best estimate the mapping function (f) for the output variable (Y) given the input data (X). The table 5.2 reviews the advantages and disadvantages of the supervised Machine Learning algorithms found in the literature for Earth Observation. Based on the research articles reviewed, it was evident that there was no one size fits all solution and each algorithm had its advantages and disadvantages. The following table summarizes the broad level merits and demerits of the algorithms as seen from the literature. From the algorithms mentioned, KNN, RF and SVM are implemented in this study. Chapter 4 details the explanation of these algorithms used.

Table 5.2: Machine Learning algorithm merits and demerits

Techniques	Merits	Demerits
Support Vector Machine [29] [40]	It is highly accurate and handle many features	It requires more time to process, High computation cost
Single Decision trees [24] [58]	Easy to process high dimension data	has the risk of overfitting. It is restricted to one output attribute and generating categorical output , If the type of data is numeric then it generates complex decision tree.
Random Forest [11] [53]	Efficient method to handle large datasets	Building and testing the model is slower process , More complex to interpret than decision trees.
K Nearest Neighbour [55]	Classes need not be linearly separable	It is sensitive to noisy or irrelevant attributes

Interpretability is a measure of the machine learning algorithm to explain its predictions. Linear Regression has the highest degree of interpretability followed by decision trees. This is followed by KNNs which has a high degree of interpretability through feature importance. The interpretability reduces with Random Forests and Support Vector Machines. *Artificial Neural Network* (ANN)s on the other hand are black-box algorithms since the underlying reason for the classification is not evident.

Training time refers to the time taken to build the model from a training data. This is the time the model takes to learn the desired parameters for classification that is used for evaluating testing set. Since, the algorithms should be able to run on a limited CPU hardware, GPU intensive ANNs are not preferred for the thesis. Hardware specifications available for the implementation are as follows:

- 1.4 GHz Quad-Core Intel Core i5
- 8 GB memory

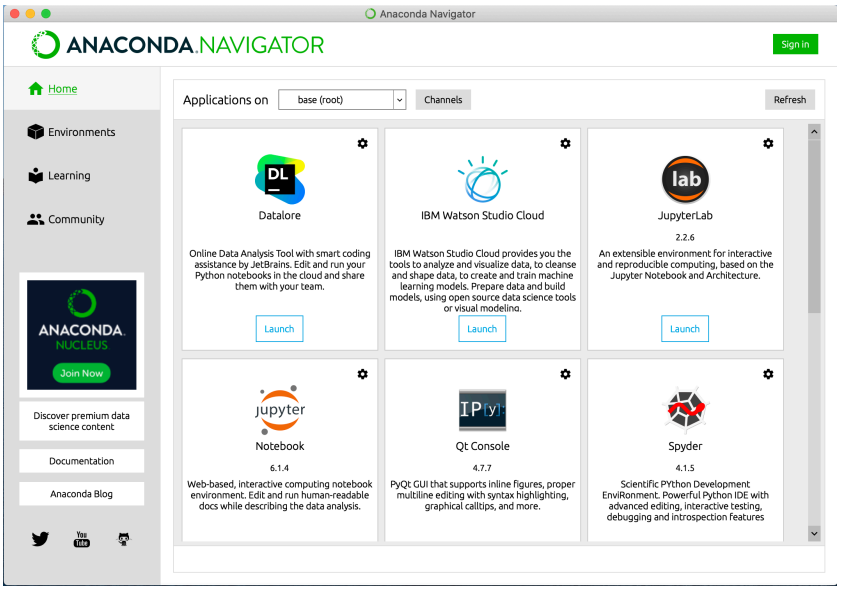
As a result of the requirements set on the selection of the algorithms, between the trade-off of higher Interpretability and lower training time, SVM, KNN and RF have been chosen to be compared.

5.4. Environment

The choice has been made to implement the machine learning code in Python. Python has established itself as a well-established language in the recent years. Further, it is the language that the author knows best, thus favouring python over other languages.

Anaconda simplifies the package management for Python for machine learning. Package management in Anaconda is performed by Conda. It comes with over 250 packages automatically installed , and over 7500 packages that can be installed using Conda. Anaconda Navigator as shown in Figure 5.5, a graphical user interface included in the Anaconda distribution provides access to the Jupyter notebook while taking care of all of the Python dependencies.

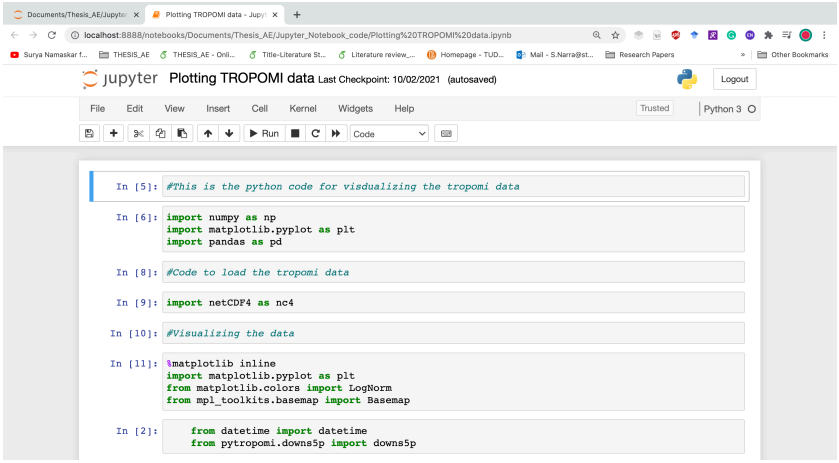
Figure 5.5: Anaconda Graphical User Interface



Jupyter Notebook supports languages like Python, Julia and R. It is an open-source integrated development environment. It is a web-based interactive computational environment [3]. It aids in exploratory data analysis as we can see inline visualization of a particular piece pf code with any dependencies on other parts of the code. Figure 5.6 shows the user interface of a jupter notebook.

[3] https://www.geeksforgeeks.org/difference-between-jupyter-and-pycharm/

Figure 5.6: Jupyter Notebook



- Git - Version 2.24.3 was used for Git. This was the Apple Git-128. The link to the Git repository is added in Appendix B

- java 15.0.2 2021-01-19 Java(TM) SE Runtime Environment (build 15.0.2+7-27) Java HotSpot(TM) 64-Bit Server VM (build 15.0.2+7-27, mixed mode, sharing).

At the time of writing the thesis, there were several free and open source software available to implement the machine learning algorithms. Some examples are: Caffe, Keras, Octave, Pandas, Scikit-learn, TensorFlow, XG-Boost [4]. Amongst these, Scikit-learn has been chosen to perform further analysis. One of the main advantages of Scikit-learn is its accessibility and simplicity. It provides simple and efficient tools for data mining and data analysis. Scikit-learn[51] is an open-source machine learning library for Python built on NumPy and SciPy python libraries released in 2007. It provides various algorithms that can be implemented for machine learning. Since, it houses the machine learning algorithms SVM,KNN and RF and has a demonstrated history of faster implementation time as compared to other libraries like mlpy, pybrain, pymvba, mdp and shogun, Scikit-learn has been shortlisted as the suitable library to implement the machine learning algorithms for the thesis. Furthermore, it has modules for pre-processing data, extracting features, optimizing hyperparameters and evaluating models. Additionally, the previous experience of using Scikit-learn by the author has also factored in the selection of the platform for the implementation.

Model Implementation

In this section, we briefly describe the implementation considerations for SVM, RF and K-NN in Scikit-learn.

Random Forest

In the case of the random forest, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node. Scikit-learn package implements a default set of hyperparameters for all models, but these are not always the most optimal set of hyperparameters. The best hyperparameters are not known ahead of time and tuning the model results in trial-and-error approach to detect the best fit. Now, to find the best fit we try different combinations to evaluate the performance of each model. Presently, the default parameters for Random Forest in Scikit-learn are as follows:

Listing 5.1: Default parameters selected for Random Forest

```
Parameters currently in use:

{'bootstrap': True,
```

[4] https://www.netguru.com/blog/top-machine-learning-frameworks-compared

```
'ccp_alpha': 0.0,
'class_weight': None,
'criterion': 'gini',
'max_depth': None,
'max_features': 'auto',
'max_leaf_nodes': None,
'max_samples': None,
'min_impurity_decrease': 0.0,
'min_samples_leaf': 1,
'min_samples_split': 2,
'min_weight_fraction_leaf': 0.0,
'n_estimators': 10,
'n_jobs': None,
'oob_score': False,
'random_state': None,
'verbose': 0,
'warm_start': False}
```

Out of all of these parameters, the important ones are explained below:

- Bootstrap : If the bootstrap is true, then the sub-sample size is set with the max samples parameter. On the other hand, if the bootstrap parameter is false, then the entire dataset is used to build the tree.
- n estimators: N estimators takes an integer as input. It refers to the number of trees in the forest.
- criterion : This parameter measures the quality of a split. In this study we look at two parameters: 'Gini' which refers to Gini impurity and 'Entropy' for information gain. Gini impurity is the probability of incorrectly classifying a randomly chosen element in the dataset if it were randomly labeled according to the class distribution in the dataset. It is given by

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

  where C is the number of classes and p(i) is the probability of of the event i being true. The optimal split for the root node is chosen while training a decision tree by maximizing the Gini Gain, which is determined by subtracting the weighted impurity of the branches from the original.

  The formula for entropy h(S) is given as follows:

$$E(S) = \sum_{i=1}^{C} -p(i) \log_2 p_i$$

- max depth: We can control the maximum depth of the trees in the random forest using this setting. To regularize the classifier model and to reduce the chance of overfitting the model we reduce the max depth of the tree.
- minimum samples leaf: It specifies the minimum number of samples required at a leaf node. If the available samples are less than the given minimum samples leaf, the leaf will not split. By increasing the minimum samples leaf, the model will be more regularized, reducing the possibility of overfitting.
- maximum features: It specifies the number of features to examine when looking for the best split. It can be of form int or float. The value can be 'auto','sqrt','log2'. If it is an integer then max features are considered at each split. For float values , then max features are a fraction and round(max features * n features) are considered at each split. For 'auto' and 'sqrt', sqrt(n features) is considered. For log2 max features is taken as log2(n features). Even if it means effectively inspecting more than max features features, the search for a split does not end until at least one valid partition of the node samples is identified.

Support Vector Machine

The implementation of SVM in Scikit-learn is based in libsvm package. The following are the parameters that need to be configured for SVM.

- The term ”kernel” is employed because the Support Vector Machine uses a collection of mathematical functions to give a window to change data. Here, we use kernel function as RBF. RBF refers to radial bias function. It is quite similar to the gaussian distribution. RBF Kernel overcomes the space complexity problem as RBF kernel SVMs only store the support vectors during training and not the entire dataset.

  The RBF kernel function for two points X_1 and X_2 computes the closeness between the two points:

  $$K(X_1, X_2) = \exp\left(-\gamma||X_i - X_j||^2\right)$$

  where $\| X_1 - X_2 \|$ is the Euclidean distance between the two points X_1 and X_2
- C refers to the regularization parameter. C is a positive float value and is inversely proportional to the strength of regularization. Recall that regularizations are techniques for reducing error and limiting overfitting by fitting a function adequately on the specified training set. The default value is ’1’.
- Gamma refers to the kernel coefficient for ’rbf’, ’poly’ and ’sigmoid’. Gamma takes float as input or it can be set to ’scale’ or ’auto’. Default value used $1/(n_features * X.var())$

Listing 5.2: Default parameters selected for SVM

```
{'C': 1.0,
 'break_ties': False,
 'cache_size': 200,
 'class_weight': None,
 'coef0': 0.0,
 'decision_function_shape': 'ovr',
 'degree': 3,
 'gamma': 'scale',
 'kernel': 'rbf',
 'max_iter': -1,
 'probability': False,
 'random_state': None,
 'shrinking': True,
 'tol': 0.001,
 'verbose': False}
```

K-Nearest Neighbour

In Scikit-learn library, the KNeighborsClassifier function is used to implement the K-Nearest Neighbour. KNeighborsClassifier learns based on each query point’s nearest neighbors, where is an integer value given by the user. The hyperparameter k specifies how many neighbours are used in the output label prediction and is dependent on the data. A larger value of k reducing the effects of noise in the data while making the classification boundaries less defined. k is often set to a odd number to prevent a situation when there is a tie between the neighbours. For further discussion on how choosing a small or a large k alffects the algorithm refer to Section 4.1 in Chapter 4. The following are the parameters that need to be configured for SVM.

- n neighbours: It takes integer as the datatype and its default value for n is 5.
- algorithm: The algorithm parameter specifies which algorithm will be used to implement the k nearest neighbour search. The options include brute-force search, KDtree, BallTree and auto setting. Using the auto setting fits the most approriate algorithmdepeding on the values passed to the algorithm.
- leaf size: The default value for the leaf size is 30 for Balltree and KDTree algorithms. The best value depends on the data passed and affects the construction speed and memory for storing the tree.
- metric: This parameter refers to the distance metric for building the tree. The default value is Minkowski distance.
- p: p is the power parameter for minkowski metric. When p=1 manhattan distance is selected and when p=2 euclidean distance is selected.

Listing 5.3: Default parameters selected for KNN

```
Parameters currently in use:
{'algorithm': 'auto',
 'leaf_size': 30,
 'metric': 'minkowski',
 'metric_params': None,
 'n_jobs': None,
 'n_neighbors': 4,
 'p': 2,
 'weights': 'uniform'}
```

5.5. Model Evaluation

In Scikit-learn the negative root mean square is the chosen scoring metric. By negating the mean square error metric we can obtain the negative mean square error metric. It is a measure of how close a classifier line is to the actual data points. MSE has the units of the variable plotted on the vertical axis squared. The equation below shows the Mean Squared error. The squaring is performed so that the negative value doesnt cancel out the positive value.

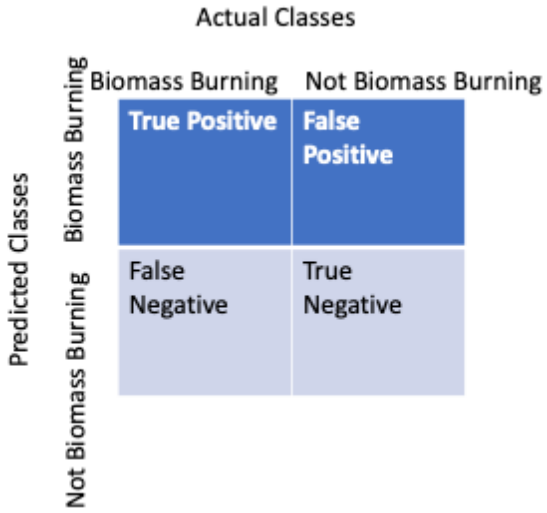
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y}_i)^2$$

MSE and RMSE are related as RMSE is the root of mean squared error. By taking the absolute value of Negative Mean squared error and then taking the root we get the root mean squared error. The root mean square error (RMSE) has been used as a common statistical metric to quantify model performance in air quality, and climate research investigations [9]. One of the reasons for its wide use is that it is the most easily interpretable metric since it has the same units as the variable plotted on the Y axis. One of the underlying assumptions in RMSE is that the errors are unbiased and follow a normal distribution. RMSE is the distance, on average of a data point from the classier fitted line measured almong the Y axis.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}$$

⋆Confusion Matrix another name for error matrix facilitates visualization of the algorithm performance and is common in supervised learning [23]. Each row of the matrix represents the instances of the actual class, whilst every column represents the predicted class. It is denoted by a matrix of size a x a associated with a classifier showing the predicted and actual classification, where a is the number of different classes. Figure 5.7 below shows the sample confusion matrix for OMI with the sample example of Biomass burning. We use this example to understand the concepts of True Positive, True Negative, False Positive and False Negative.

Figure 5.7: Sample Confusion Matrix format for OMI biomass burning



From the confusion matrix, we can deduce a number of results. Firstly, we can use it to calculate the accuracy with respect to the ground truth of the classification. Secondly, it is used to recognize the specific errors affecting each class labels. Thirdly, it help evaluate and quantify the extent to which the model underfits or overfits each class label/category. Below, the formulas for calculating five different metrics measuring the performance of the model are given.

- Accuracy =

$$\frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

- Misclassification =

$$\frac{FalsePositive + FalseNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

- Precision =

$$\frac{TruePositive}{TruePositive + FalsePositive}$$

- Sensitivity/Recall =

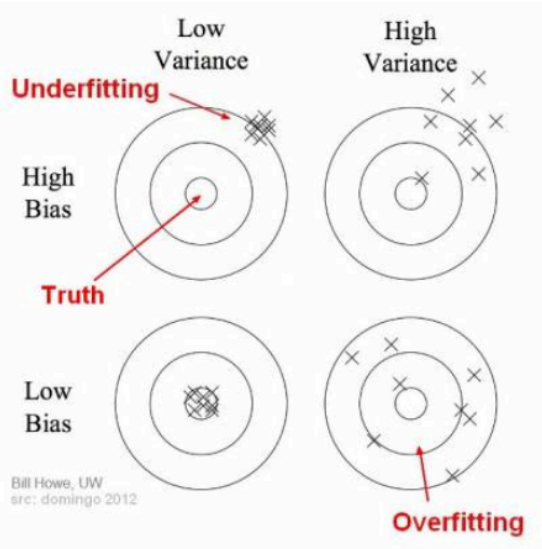
$$\frac{TruePositive}{TruePositive + FalseNegative}$$

- Specificity =

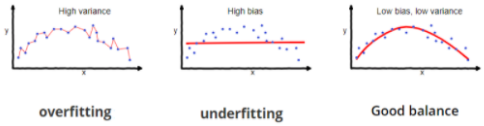
$$\frac{TrueNegative}{TrueNegative + FalsePositive}$$

⋆ Bias and Variance can determine where a model is well performing in the true sense or if it is overfitting or underfitting. Bias is the difference between the Predicted Value and the Expected Value. When there is a high bias error, it results in a very simplistic model that does not consider the nuances in the data. Since, it does not learn the data properly, it is called underfitting. Variance on the other hand is when the model is too much in sync with the variations in the data along with the noise. What this results in is that the new model is predict accurately on the new data since it has learnt too much from the training data.
In the Figure 5.8 the bulls-eye is the correct prediction of the target. The goal is to achieve low bias and variance for the most accurate predictions as can be seen in the Figure the predictions all lie inside the bulls-eye area. When there is low bias and high variance in the data, it leads to overfitting of the data. When there is low variance and high bias, it leads to underfitting of the data.

Figure 5.8: Bias and Variance Tradeoff [18]



The below Figure shows the different fits of the models: Underfitting, Overfitting and Optimal fit, along with the trade=off between bias and error. The total error of a machine learning model is the sum of the bias error and variance error. The aim is to balance bias and variance such that the model does not underfit or overfit the data. Overfitting refers to error which occurs when a function is too closely fit to a limited set of data points. Underfitting refers to the modelling error when the model can neither model the training data nor generalize to new data.



(a) Underfitting, Overfitting and optimal fit [60]

(b) Reaching optimal fit for bias and variance [60]

Figure 5.9: Bias and Variance

The performance of the three machine learning models will be evaluated in the Chapter 6.

5.6. High Level Requirement Traceability Matrix

The following table encompasses the Requirement Traceability Matrix which would effectively guide as a measurable progress bar for the developments achieved in the project.

ReQ-ID	Requirement	SubCategory
R-001	The application shall implement supervised learning algorithm for the purpose of aerosol classification on a flying mission	Application
R-002	The application shall be be validated against previous satellite mission	Application
R-003	The application shall to a reasonable degree to accuracy be able to distinguish between at least five major types of aerosols.	Application
R-004	The application shall be capable to be deployed and tested on a computer with the specifications of 1.4 Ghz processor and 8 Gb RAM	Performance
R-005	The application shall reach a performance metric of accuracy of at least 90% on the test dataset	Performance

6

Implementation and Results

The implementation section is divided into two main categories. First, we look at the implementation with POLDER-3 data. The implementation begins with first obtaining the data, understanding its structure, cleaning the data and exporting it into a format that can be used for further processing. Then, the steps mentioned in the Methodology section are repeated with OMI data. In the results section, we look at the results derived for the satellite instruments POLDER-3 and OMI by using the supervised learning algorithms of SVM, RF and KNN. After following all the steps mentioned in Chapter 5 methodology, we compare the performance of the three supervised learning algorithms against each other and study how varying the hyperparameters affects the accuracy of the algorithm.

6.1. POLDER Data

POLDER-3 was an instrument on the PARASOL microsatellite. It was launched on 18 December 2004. The POLDER-3 instrument and the corresponding PARASOL mission was decommissioned on 18 December 2013. The POLDER instrument collected accurate observations of the polarized and directional solar radiation reflected by the Earth-atmosphere system.

The data presented for POLDER-3 instrument is in the following format. The year 2006 was chosen since that was the reference year shared by SRON, who provided the dataset. In the Table 6.1 the various data parameters that can be extracted from the POLDER-3 data are presented.

Table 6.1: POLDER-3 data parameters [16]

Abbreviation	Full name	Wavelengths / Modes / Variants
AOT	Aerosol Optical Thickness	440, 490, 563, 670, 865, 1020 nm
SSA	Single Scattering Albedo	440, 490, 563, 670, 865, 1020 nm
reff	Effective Radius	Fine, Coarse Mode Fraction
veff	Effective Variance	Fine, Coarse Mode Fraction
m_r	Real Refractive Index	Fine, Coarse Mode Fraction
m_i	Imaginary Refractive Index	Fine, Coarse Mode Fraction
sphere_frac	Sphericity	Fine, Coarse Mode Fraction
lat	Latitude	Coordinates of Centers and Corners
long	Longitide	Coordinates of Centers and Corners
psurf	Surface Pressure	
N	Aerosol Column Number Density	Fine, Coarse Mode Action
number_of_points	Number of Data Points	Over Ocean, Land
error	Parameter Retrieval Uncertainty / Error	AOT, SSA, reff, veff, m_r, m_i, N, sphere_frac

POLDER-3 was introduced in Section 3.2 where we saw the description of the PARASOL mission and an introduction to the POLDER-3 data. In Figure 5.4, we saw the flow diagram for the implementation of the methodology on POLDER-3 data. The first step was data extraction from POLDER. The year 2006 was chosen since that was

the reference year shared by Otto Hasekamp from SRON, who provided the dataset. The data was provided in .nc format which refers to netcdf or network common data files. The nc file was extracted in R due to the ease of extraction with the packages in R and a dataframe was created which would be ported in Python for further processing and implementation of machine learning algorithm.
A dataframe is a two dimensional data structure containing columns of required variables. It is much like a spreadsheet. The top row is called the header and each individual row that follows is called a cell. The benefit of using a dataframe is the easy portability between the R environment and python environment. In Figure 6.12d, we see the dataframe with 1131324 rows and 10 columns. The rows represent the datapoints collected across the 12 months for POLDER, while the columns represent the variables that will be used for machine learning chosen from table 6.1. But first, we look at the correleration matrix created from this dataframe

Figure 6.1: Dataframe created from the extracted variables from the POLDER-3 data

	Latitude	Longitude	Day	SSA_490nm	AE	sphere_frac_coarse	m_r_fine	Pixel_ID	Month	Season
0	27.0	-160.0	1	0.868218	0.408807	0.908730	1.558705	3718	1	Winter
1	40.0	-160.0	1	0.939893	0.175073	0.643669	1.684650	3731	1	Winter
2	25.0	-159.0	1	0.843641	0.436078	1.000000	1.470230	3896	1	Winter
3	26.0	-159.0	1	0.819170	0.305589	0.981574	1.455080	3897	1	Winter
4	27.0	-159.0	1	0.877588	0.376715	1.000000	1.508170	3898	1	Winter
...	...	...	...	...	...	...	...	...	...	...
1131319	-14.0	179.0	29	0.999674	1.170798	0.472841	1.361380	64697	12	Winter
1131320	-13.0	179.0	29	0.991572	0.508327	0.419278	1.486690	64698	12	Winter
1131321	-12.0	179.0	29	0.979904	0.997900	0.941176	1.383620	64699	12	Winter
1131322	-11.0	179.0	29	0.998360	1.035866	0.896197	1.426510	64700	12	Winter
1131323	-2.0	179.0	29	0.965086	0.384436	1.000000	1.417645	64709	12	Winter

1131324 rows × 10 columns

A correlation matrix is a table that presents the coefficients of correlation between variables. Each cell in the table contains the information about the correlation between two variables. The value of correlation ranges from 0 to 1, with 0 being no correlation between the features to 1 being highly correlated. Correlation gives us a clear understanding of the importance of the features before building the machine learning model. With the help of the correlation analysis we can also check if we have multicollinearity i.e. high correlation between the variables, which is undesirable in the role of building a machine learning model. From 6.2 the feature correlation matrix, we can deduce the following points:

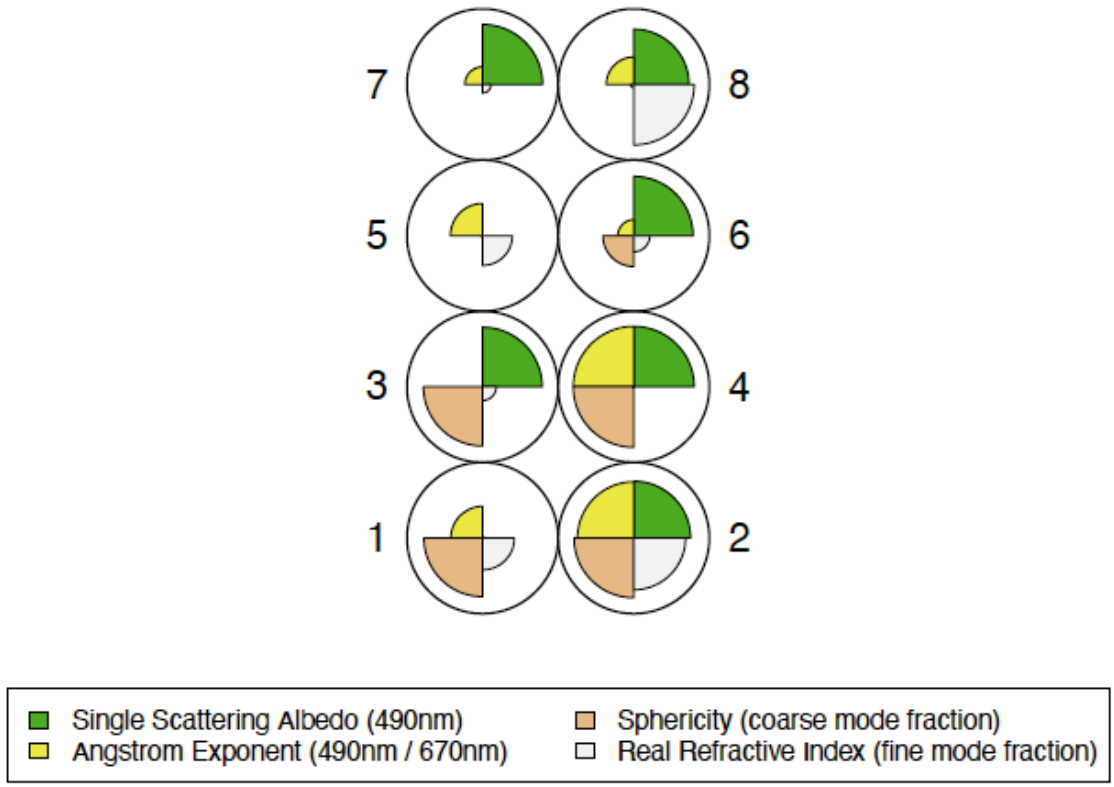
- There is no strong correlation between the features and the cluster label. This means that there is no dominant features dictating the output, thus giving us unbiased labels independent of the feature.
- The cluster number label and variables: Angstrom Exponent and sphericity are mildly negatively correlated. This means that as the value of angstrom exponent and sphericity increases the clustering label number decreases.
- The features RRI(mrfine) and SSA are mildly positively correlated. This means that as the value of RRI(mrfine) and SSA increases the clustering label number increases.
- Sphericity seems to the have the highest negative correlation to other measured parameters.

Figure 6.2: Feature correlation matrix for the four features.

	Latitude	Longitude	Day	SSA_490nm	AE	sphere_frac_coarse	m_r_fine	Month	Cluster
Latitude	1.000000	0.077774	-0.005893	-0.052624	0.186939	-0.069630	0.118656	-0.069772	0.056969
Longitude	0.077774	1.000000	-0.001220	-0.024316	0.159946	-0.057605	0.001564	0.022173	0.035861
Day	-0.005893	-0.001220	1.000000	0.008380	-0.001961	0.003749	0.004470	0.056760	-0.001718
SSA_490nm	-0.052624	-0.024316	0.008380	1.000000	-0.091750	0.006134	-0.157240	0.030243	0.138471
AE	0.186939	0.159946	-0.001961	-0.091750	1.000000	0.133670	0.038676	0.065547	-0.169765
sphere_frac_coarse	-0.069630	-0.057605	0.003749	0.006134	0.133670	1.000000	-0.057375	0.044663	-0.899141
m_r_fine	0.118656	0.001564	0.004470	-0.157240	0.038676	-0.057375	1.000000	0.038991	0.062133
Month	-0.069772	0.022173	0.056760	0.030243	0.065547	0.044663	0.038991	1.000000	-0.032020
Cluster	0.056969	0.035861	-0.001718	0.138471	-0.169765	-0.899141	0.062133	-0.032020	1.000000

In Figure 6.3, we can see the correlation matrix being visualized in terms of weights in the clusters. We saw from the correlation matrix that sphericity is negatively highly correlated and thus here we see that for lower cluster numbers (1,2,3,4), the weights of the sphericity is higher as compared to higher cluster numbers (5,6,7,8).

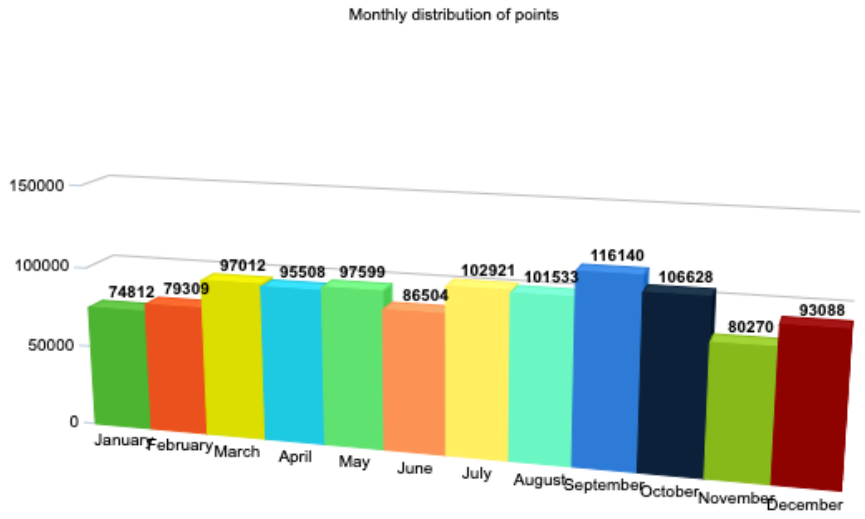
Figure 6.3: Weights for each hyperphysical parameter in POLDER-3 data for the year 2006 [16]



After reading all the data points, we got a total of 23,651,900 points. However, many of these observations are NANs where possibly data was not collected efficiently. When only the complete cases were considered, the output resulted in final values of 1,131,224 data points. It is worth noting that there is not an equal distribution of datapoints across the twelve months of the year 2006. From 6.4, we can see that there is a difference of nearly 41,328 datapoints between the month with the highest datapoints September and the month with the lowest datapoints January. What this means is that the dataset is not a perfect distribution of data across the months but infact is a results of error in collecting data or NaNs that were created in the capturing of the data. It is worth noting that while the data is not a perfect distribution that difference is not large enough to create an imbalance

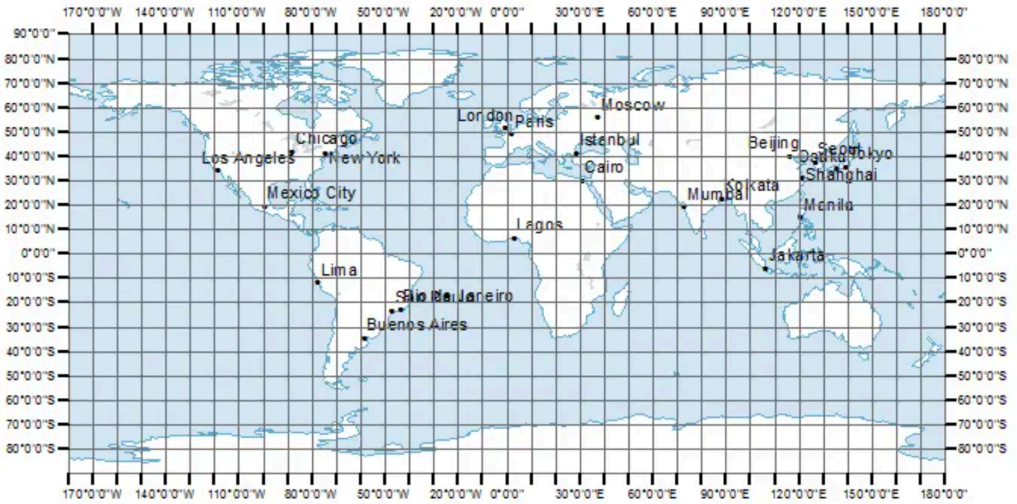
in the data that would affect the machine learning model.

Figure 6.4: Monthly distribution of the data points



From the extracted data points in Figure 6.4, we create the seasonal labels. December, January, February make 'Winter'; March, April, May make 'Spring'; June, July, August make 'Summer' and September, October and November make 'Autumn'. Next, we plot the seasonal (northern hemisphere seasons) i.e. Winter, spring, summer and fall variations in the satellite acquired data regarding the aerosols. In Figure 6.5 the world maps with major cities are plotted along their latitude and longitude across the x and y axis. This reference frame will be used throughout the rest of the project to plot the data of the three satellite data.

Figure 6.5: Coordinates for the aerosol reading from the satellite data [21]



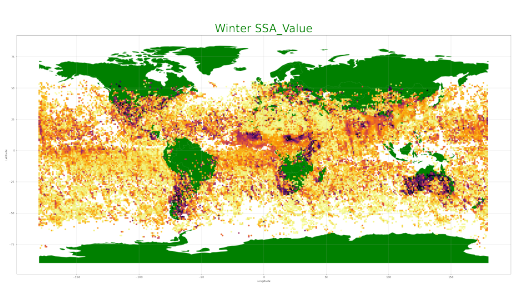
Here, we plot the following four variables

- 1) SSA - Single Scattering Albedo is defined as the ratio of scattering efficiency to the total extinction efficiency or the sum of both absorption and scattering. SSA is a unitless quantity, which if zero indicates that the absorption is the cause of particle extinction. Whereas, if the SSA is 1 then scattering is the cause

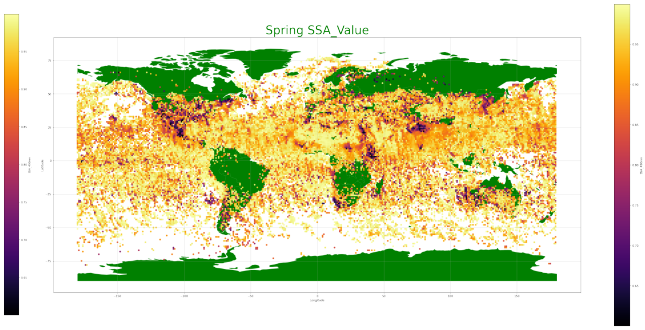
of particle extinction. Particle extinction refers to the reduction in the radiation between an emitting source (e.g. Sun) and the observer (e.g. satellite) due to dust and gas (e.g. aerosols) in between the path.

Figure 6.6: Seasonal variation in Single Scattering Albedo

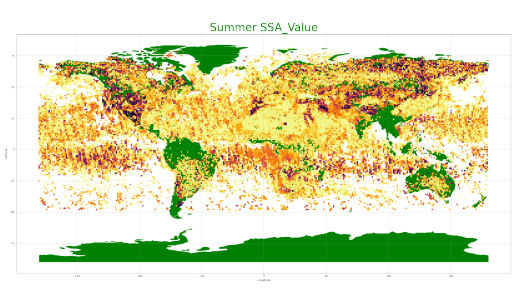
(a) Winter SSA 490 nm distribution



(b) Spring SSA 490 nm distribution



(c) Summer SSA 490 nm distribution



(d) Autumn SSA 490 nm distribution

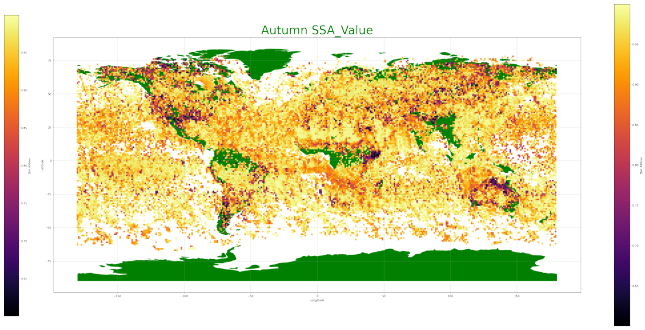
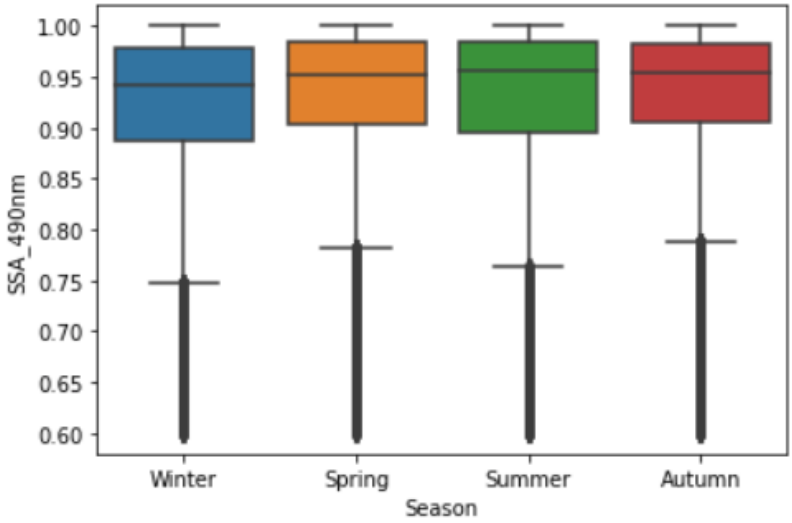


Figure 6.6 plots SSA values on the world map in the color gradient scheme. The colour indicates the value of SSA, with darker blue color meaning a lower value of SSA and lighter yellow value meaning a greater value of SSA. The colour gradient is shown in the plot legend. From Figure 6.6, we can see that the SSA value varies seasonally over geographical locations on earth. Multiple observations can be deduced. Firstly, the data is incomplete along the north and south poles. This might be due to weather disturbances or cloud cover impeding data collection. Secondly, we see that high SSA values occur over South America during the summer months of June, July and August as marked by higher yellow and orange values. This means that there was higher reflecting aerosols in summer. This is in sync with literature which states that agricultural fires and land clearing common in the Amazon basin spanning South America.

From Figure 6.7, we can see that the upper whisker or the upper bound of all seasons is the maximum value of SSA i.e. 1. The median has only minor variation across the four seasons. The same can be seen with the first and third quartiles. However, there is a significant variation between the minimum SSA values across the four seasons. All the seasons display outliers beyond the minimum values. Another observation is that SSA is limited by the range between 0.6 and 1 while theoretically the range acn be from 0 to 1 for SSA.

Figure 6.7: Box plot of SSA distribution vs seasons

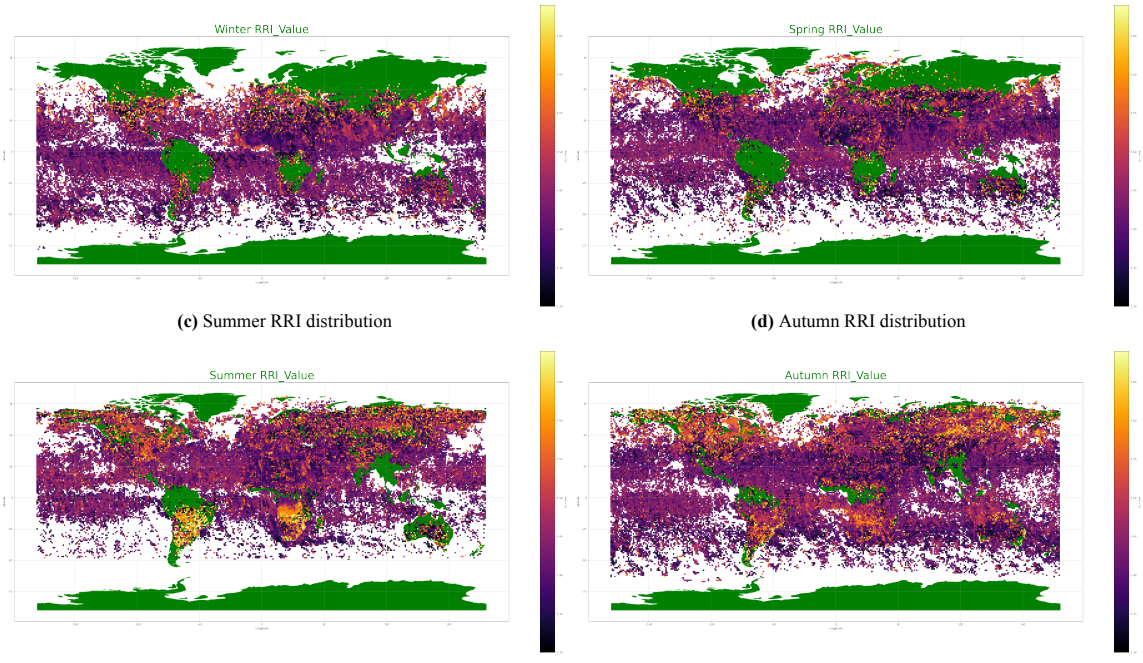


- 2) RRI - Real Refractive Index is an important parameter to determine the absorption and scattering of the particle along with the information regarding the particles size. It consists from both real and imaginary parts and for the purpose of this study we will be concentrating on the real part.

Figure 6.8: Seasonal variation in Real Refractive Index

(a) Winter RRI distribution

(b) Spring RRI distribution



(c) Summer RRI distribution

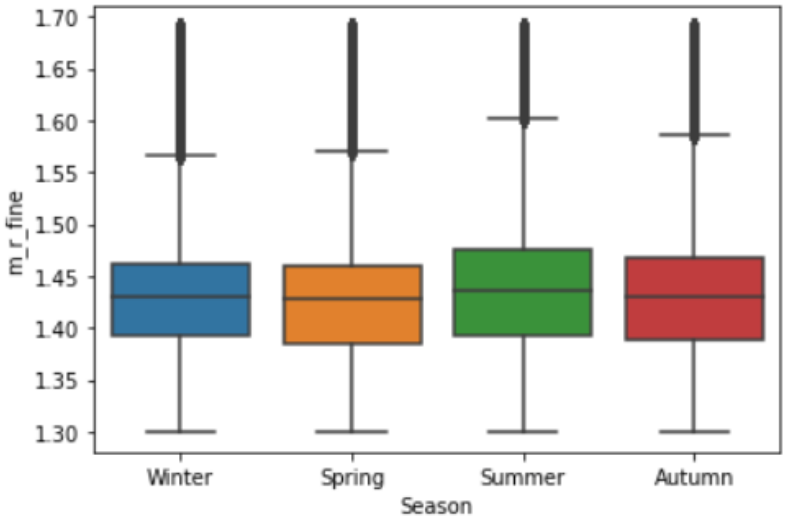
(d) Autumn RRI distribution

Figure 6.8 plot RRI values on the world map in the color gradient scheme. The colour indicates the value of RRI, with darker blue colour meaning a lower value of RRI and lighter yellow value meaning a greater value of RRI. The colour gradient is shown in the plot legend. From Figure 6.8, we can that RRI value varies seasonally over the geographical locations on earth. It is worth noting that the data is incomplete along the north and south poles as was the case with SSA.

It is observed that only in the summer months, the RRI is highest in the South American and African regions compared to the rest of the world and rest of the seasons. Moreover, only in summer and autumn there is

a lot of scattering of RRI in contrast with winter and spring RRI distribution.

Figure 6.9: Box plot of RRI distribution vs seasons



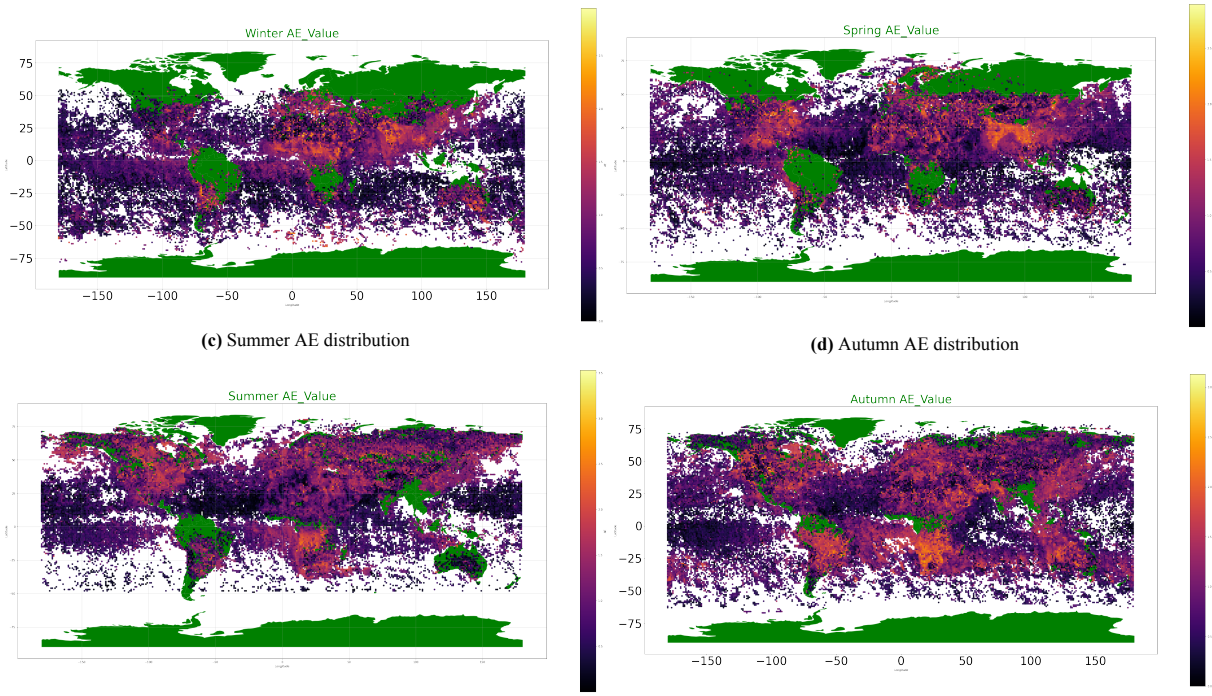
From Figure 6.9, it is observed that the medians for all four seasons are well under 1.45 RRI value. This indicates that there isnt much scattering in the value of RRI among the four seasons. Further more, there is a significant difference in the upper limits, with summer season having the highest RRI around 1.6 while winter has the lowest. All the value of RRI are contained between 1.30 and 1.70 across the four seasons.

- 3) AE - Angstrom Exponent a parameter that specifies how an aerosol's optical thickness is conventionally affected by the wavelength of light. $(\tau_\lambda)/(\tau_{\lambda 0}) = (\lambda)/(\lambda_0)^\alpha$ where $(\tau_(\lambda))$ is the optical thickness, (λ) is the wavelength and (α) is the angstrom exponent. The larger the angstrom exponent the smaller the particle and the smaller the angstrom exponent the larger the particle size. For example, the clouds have a large size, nearly zero angstrom exponent and thus have negligible relationship between optical depth and wavelength. This results in the clouds appearing white.

Figure 6.10: Seasonal variation in Angstrom Exponent

(a) Winter AE distribution

(b) Spring AE distribution

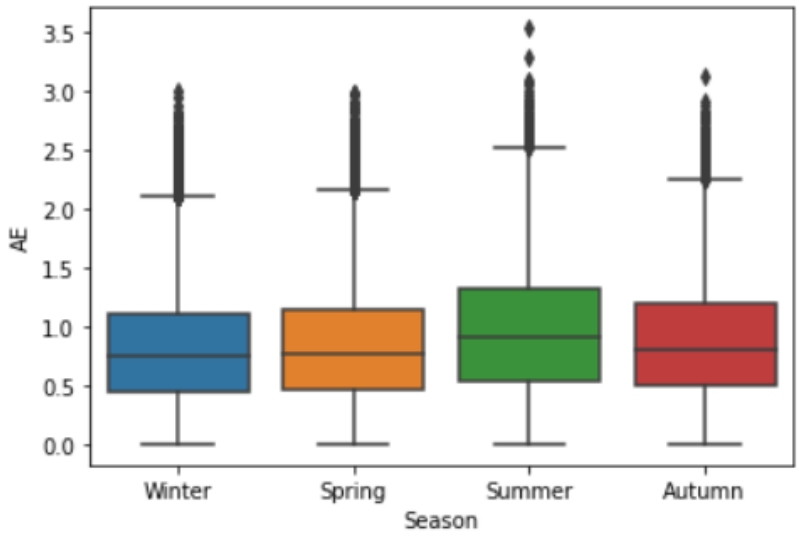


(c) Summer AE distribution

(d) Autumn AE distribution

From Figure 6.10, we can see that larger particle size i.e. lower angstrom exponent is observed over the oceans. It is seen that the denser the population of an area the lower the particle size. The same can be seen in the map in populated and industrious areas of India, China and Africa.

Figure 6.11: Box plot of Angstrom exponent distribution vs seasons

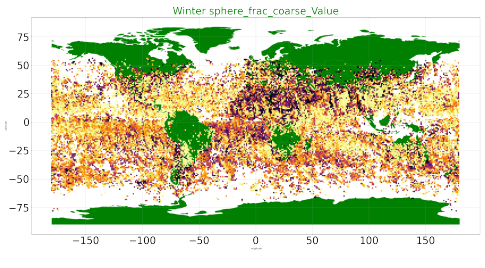


From the boxplot in Figure 6.11 we can see that the values for angstrom exponent is between 0 and 3.5 across the four seasons. Summer has highest value of AE at 2.5. All the four seasons have the lowest whisker of the box plot at 0. Summer has a slightly higher median than the other seasons which have comparable median.

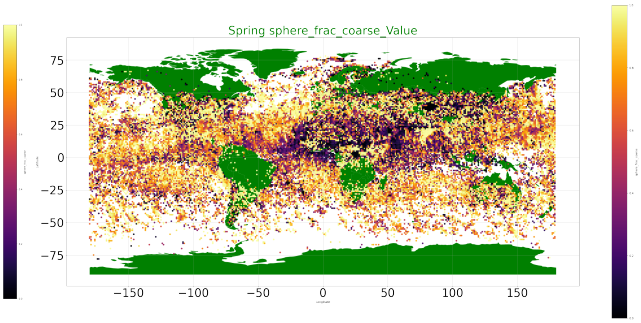
- 4) Sphericity - Spherical Fraction Coarse Distribution is defined as the ratio of the surface area of the sphere of the same volume as the particle under reference to the actual surface area of the particle. A perfect sphere has a sphericity of 1, other shapes have a sphericity less than 1.

Figure 6.12: Seasonal variation in spherical coarse fraction

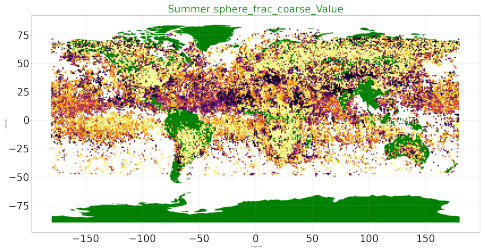
(a) Winter spherical fraction coarse distribution



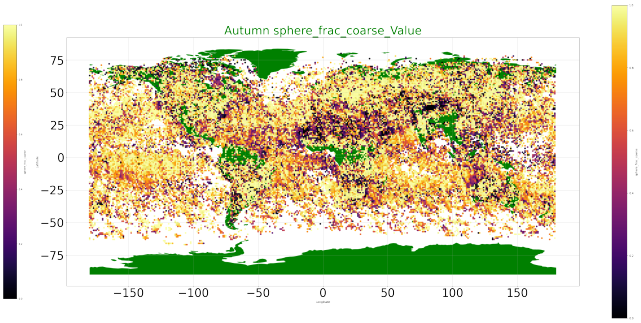
(b) Spring spherical fraction coarse distribution



(c) Summer spherical fraction coarse distribution



(d) Autumn spherical fraction coarse distribution



From Figure 6.12, we see that the sphericity data points are distributed across the world during summer and autumn seasons. It is seen that in the spring season, the East Asian region has least amount of sphericity. Overall, high sphericity is maintained across the four seasons.
From the boxplot in Figure 6.13, it is seen that sphericity has a range between 0 and 1. Spring has a lower first quartile as compared to other seasons. The spread between first and third quartile is significantly larger in the spring season compared to other seasons. In Summer and Autumn, the sphericity value are closer to 1.

Figure 6.13: Box plot of Spherical Fraction distribution vs seasons

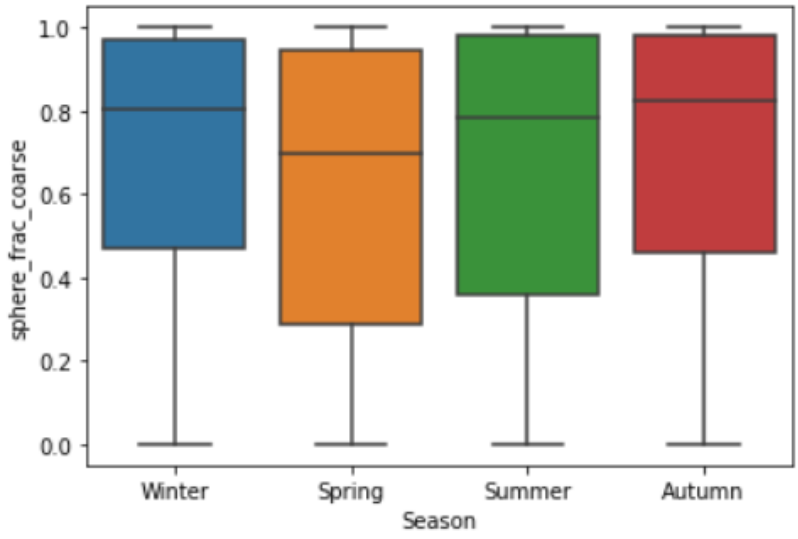


Table 6.2: Eight clusters found in POLDER-3 data that used as labels for supervised learning for N = 1, 131, 324 resulting in cluster labels for aerosols on 339,402 datapoints. observations [16]

heightCluster	Proposed Aerosol Types	No. of data points
1	Smoke	25,328
2	Mixed Smoke	31,212
3	Marine	76,391
4	Urban-Industrial	56,527
5	Dusty Smoke	15,105
6	Marine Dust	63,879
7	Dust	52,711
8	Polluted Dust	18,249

In Table 6.2, eight clusters found in POLDER-3 data that used as labels for supervised learning for N = 339,402 observations are stated. We use these class labels as the beginning step in the classification of the supervised learning algorithm. In table 6.3 we see the centroids and their distribution of the standard values of the four hyperparameter variables SSA, AE, sphere_frac_coarse and m_r_fine.

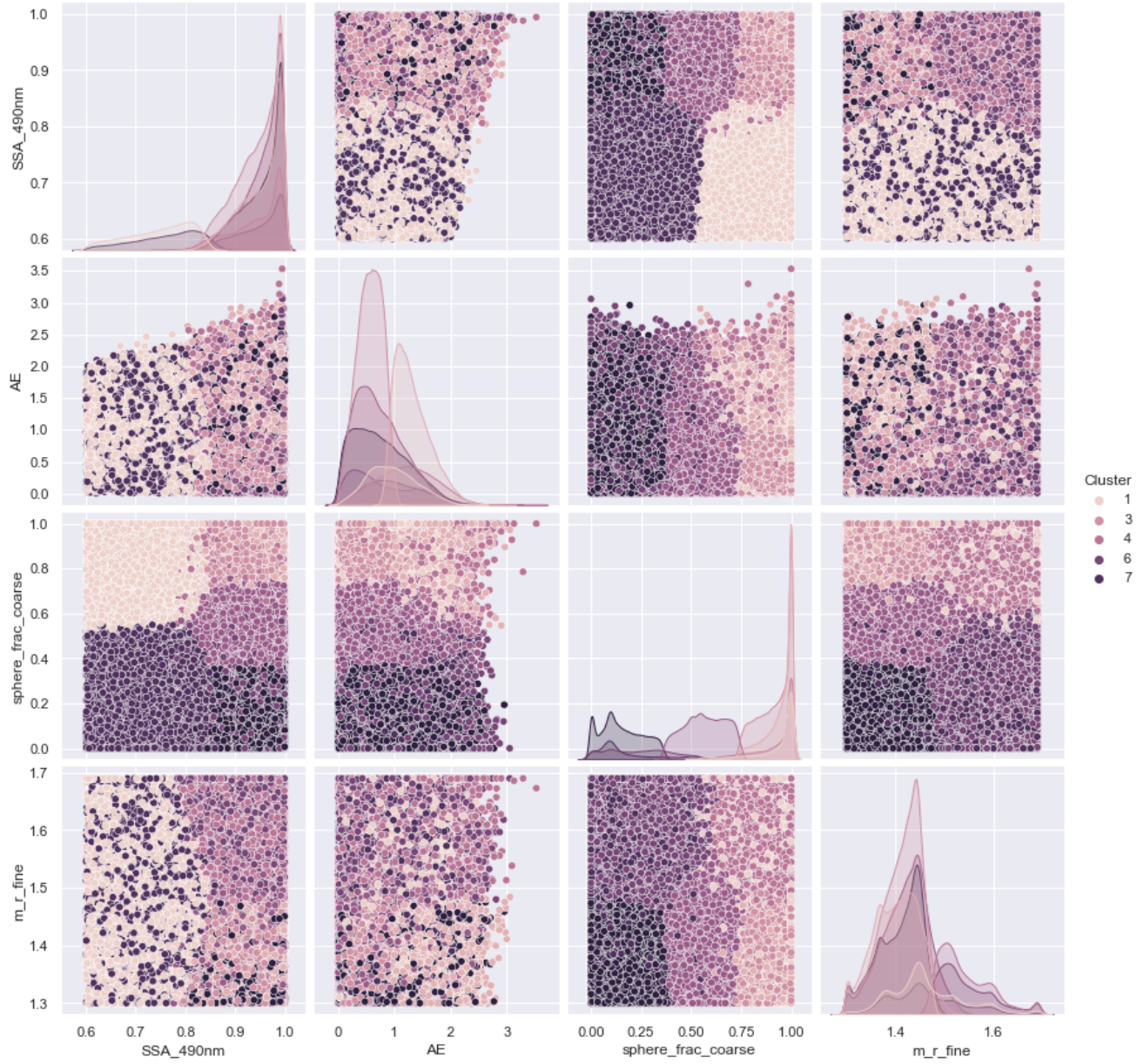
Table 6.3: Centroids of the eight found clusters and their hyperparameter values [16]

	SSA_490nm	AE	sphere_frac_coarse	m_r_fine
V1	0.9576819	0.7875106	0.1670247	1.401883
V2	0.7673035	0.9227107	0.1968260	1.459614
V3	0.9429534	0.7984468	0.2313506	1.549334
V4	0.9505147	0.7686054	0.5509413	1.425842
V5	0.9456691	0.5514180	0.9155904	1.414115
V6	0.9407932	1.2898002	0.9177906	1.534532
V7	0.7485480	0.9893963	0.9272278	1.463060
V8	0.9466955	1.2850705	0.9317799	1.390002

With these centroids and labels collected from Vincent de Bakker's thesis, we implement the supervised learning algorithms from scikitlearn library in python in jupyter notebook environment.
The pair-wise plot in Figure 6.14 is used to understand the relationship between the microphysical parameters derived from POLDER-3 data. The plots are displayed in the matrix format where the row name represents the x axis and the column name represents the y axis. Here the feature label 1,2,3,4,5,6,7 and 8 refer to Smoke, Mixed Smoke, Marine, Urban Industrial,Dusty Smoke, MarineDust,Dust and PollutedDust respectively. Each plot in the figure apart from the diagonal plots are a pairwise plot of two microphysical parameters and show how the clusters are distributed for different ranges of these parameters. The diagonal plots are different from the rest of the pairwise plots. A univariate distribution plot is displayed to show the marginal distribution of the data in each column in the form of a kernel density plot. The remaining plots are scatter plots.

Recall from the previous chapter in implementation that we left of with finding the eight centroids for POLDER-3. R Each of the 339,402 datapoints for POLDER-3 for the year 2006 was assigned a class label associated with these centroids. We use these labels as the ground truth for training the data. We split that data into 70 % for training and 30 % for testing. In the 70 % of the data that is used as the training data, we again split it further according to the number of folds for cross-validation. Cross-validation is used to evaluate a model's ability to predict new data that was not used in its estimation, in order to identify issues such as overfitting or selection bias, as well as to provide insight into how the model will generalize to a different dataset.

Figure 6.14: Feature Pair Plot



In the code below we are making four demarcations of the data for the purpose of fitting and predicting values. X_train includes all the data that will be used to train the model. For our use case 70% of the data (339,402 data points) are in X_train. The remaining 30% of the data (101,821 datapoints) are in X_test. These values are used to predict the class label and the accuracy of the model. y_train refers to the class labels. y_test contains the category class labels for the test data. This is used to test the accuracy between the actual and predicted categories.

Listing 6.1: Code to split the dataframe into train and test set

```
X_train, X_test, y_train, y_test = train_test_split(X,Y,
    train_size=0.7,test_size=0.3,random_state=100)
```

In the code below, seven folds are made to meet the cross-validation step as discussed in Chapter 5. Recall that in methodology we explained the k-fold cross validation technique. Here, we use k = 7. For each fold, 1/7 data is used as a testing set and 6/7ths of the data is used as a training data set. This step is repeated 7 times for each algorithm. Scoring is a method of evaluating the performance of the algorithms on the test set depending on the number of correct predictions and can be measured in terms of accuracy, root mean square error or other

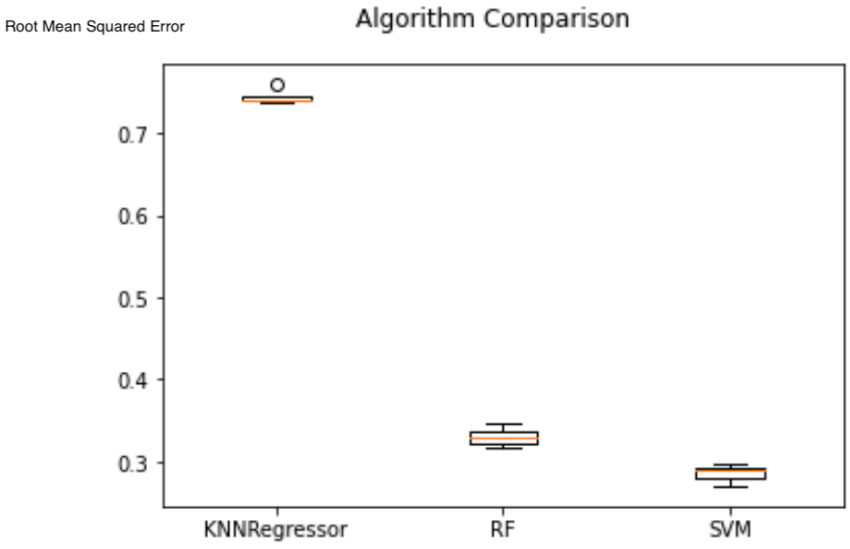
parameters. Once the model is trained on 70 % of the data, we used the negative mean squared error metric to calculate the score of each of these models on the test set. As mentioned in Chapter 5, we now compare the performance of the three algorithms: SVM, KNN and RF based on root mean square scoring metric. The theory behind each of these algorithms is mentioned in Chapter 4 and the implementation is provided in Chapter 5.

Listing 6.2: Code to calculate the least mean squared error

```
for name, model in models:
    kfold = model_selection.KFold(shuffle=True,n_splits=7,random_state=0)
    cv_results = model_selection.cross_val_score(model, X_train, y_train,cv=kfold,
        scoring='neg_mean_squared_error')
    results.append(np.sqrt(np.abs(cv_results)))
    names.append(name)
    print("%s: %f (%f)" % (name,
        np.mean(np.sqrt(np.abs(cv_results))),np.std(np.sqrt(np.abs(cv_results)),ddof=1)))
   #boxplot algorithm comparison
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results )
ax.set_xticklabels(names)
plt.show()
```

In 6.15, we are looking at the algorithm with the least mean squared error. The Root Mean Squared Error or (RMSE) is a metric that indicates how near a fitted line is to the data points and is our chosen metric for scoring the algorithms. The lower the mean squared error the better the performance of the model. In this instance, from Figure 6.15, it is evident that the algorithm SVM with a score of 0.3 has the least mean squared error and thus has the best performance.

Figure 6.15: Root Mean Squared Error results of KNN, RF and SVM



Next, it is worth noting that in the preliminary analysis, we took the default hyperparameter tuning setting for fitting the SVM. However, running a simulating training for different parameter setting yields different estimator performance results. In listing 6.3, we can see that there are three hyperparameters that can be varied to improve the performance of the SVM algorithm. GridsearchCV, a library function in Scikit-learn provides a comprehensive search over specified parameter values for an estimator. For this grid search we vary the values of C and Gamma. C is varied from 0.1, 1, 10 and 100. Gamma is varied from 1, 0.1, 0.01 and 0.001. The term ”kernel” is employed because the SVM uses a collection of mathematical functions to give a window to change data. Here, we use kernel function as *Radial Basis Function* (RBF). It is quite similar to the Gaussian distribution. RBF Ker-

nel overcomes the space complexity problem as RBF kernel SVMs only store the support vectors during training and not the entire dataset. Note that while scoring the algorithm on default parameter we had chosen k = 7. Here, we used cross-validation with k = 5 to reduce the time it takes run the optimization.

The RBF kernel function for two points X_1 and X_2 computes the closeness between the two points:

$$K(X_1, X_2) = \exp\left(-\gamma||X_i - X_j||^2\right)$$

where $\| X_1 - X_2 \|$ is the Euclidean distance between the two points X_1 and X_2

Listing 6.3: We use the code below to find the best parameter search for SVM algorithm.

```
from sklearn.model_selection import GridSearchCV

# defining parameter range
param_grid = {'C': [0.1, 1, 10, 100],
              'gamma': [1, 0.1, 0.01, 0.001],
              'kernel': ['rbf']}

grid = GridSearchCV(svm.SVC(), param_grid, refit = True, verbose = 3)

# fitting the model for grid search
grid.fit(X_train, y_train)
```

In Figure 6.16, 80 different fits were generated for the SVM algorithm. The resulting fits with different hyperparameters, i.e. different combinations of C and gamma for the kernel as 'rbf' are generated. The resulting fits are compared with each other against the score generated. The higher the score, the better performing is the algorithm. Among the two hyperparamters being tuned, low C means low error and if we have large C it means large error. The second hyperparameter Gamma dictates the curvature in the decision boundary. The higher the Gamma, the higher the curvature. Conversely, the lower the gamma the lower the curvature.'C': 100, 'gamma': 0.001, 'kernel': 'rbf' SVC(C=100, gamma=0.001) generated the best fit. It is worth noting that each hyperparameter fit takes a different amount of time to train. For example, for C = 0.1 and gamma =1 took a total of 41 mins to train.

Figure 6.16: Parameter search

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] ......... C=0.1, gamma=1, kernel=rbf, score=0.940, total=41.1min
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed: 41.1min remaining:    0.0s
[CV] ......... C=0.1, gamma=1, kernel=rbf, score=0.939, total=39.1min
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed: 80.3min remaining:    0.0s
[CV] ......... C=0.1, gamma=1, kernel=rbf, score=0.938, total=38.0min
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] ......... C=0.1, gamma=1, kernel=rbf, score=0.938, total=37.0min
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] ......... C=0.1, gamma=1, kernel=rbf, score=0.940, total=37.6min
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.980, total= 8.3min
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.979, total= 8.5min
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
```

From Figure 6.17, we can see that the mean score is lowest for parameter gamma = 1. The mean test score is the highest for gamma = 0.001. The general trend seen in the graph in Figure 6.18 is that mean test score rises initially and then falls off with increasing gamma. For lower value of gamma at 0.001 for C =100 the score was the highest and then it falls off for C=100. The highest accuracy is for C = 100 and gamma = 0.001.

Figure 6.17: Dependence of C and Gamma with accuracy. The Y axis represents the mean test score of accuracy taken over the five folds of cross-validation. The X axis represents the parameter C varied over 0.1, 1, 10 and 100. The line graphs represent the four parameters of gamma at 0.001, 0.01, 0.1 and 1.

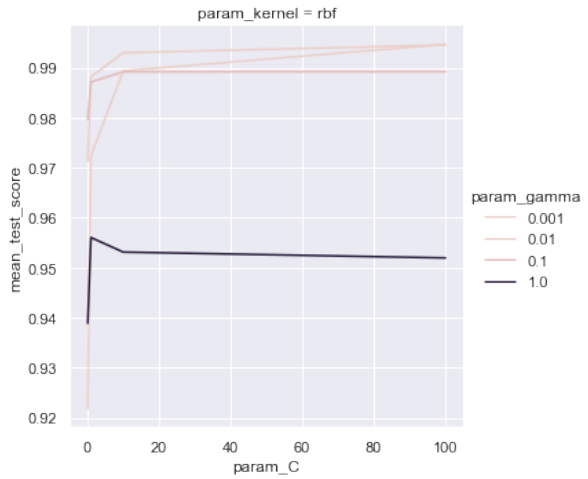
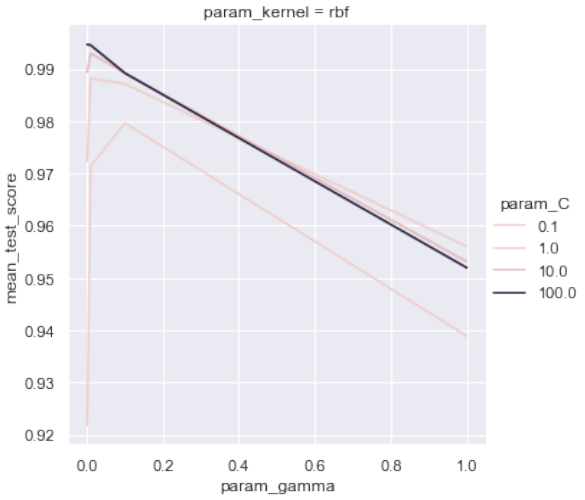


Figure 6.18: Dependence of C and Gamma with accuracy. The Y axis represents the mean test score of accuracy taken over the five folds of cross-validation. The X axis represents the parameter Gamma varied over 0.001, 0.01, 0.1 and 1. The four line graphs represent different values of parameter C at 0.1, 1, 10 and 100.



From Figures 6.19 and 6.20 we see the dependence of C and Gamma with mean rank score.Rank 1 is for C =100 and gamma =0.001. The rank 16 or lowest rank is for C =0.1 and gamma = 0.001.

Figure 6.19: Dependence of C and Gamma with rank score. The Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter C. The four line graphs are for parameter gamma at 0.001, 0.01, 0.1 and 1.0.

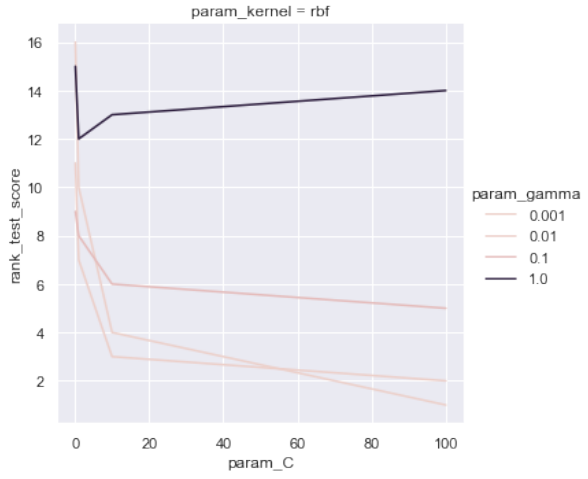
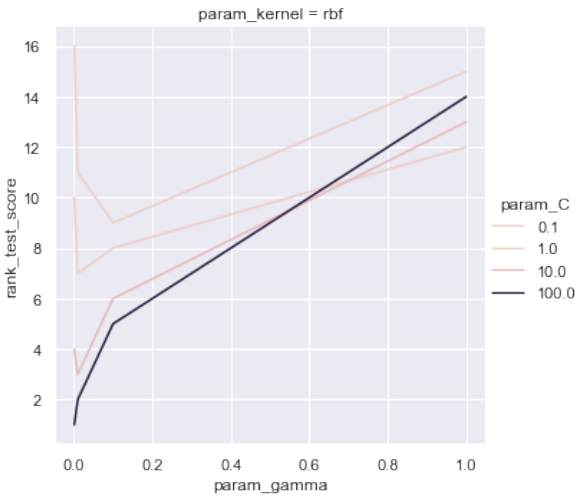


Figure 6.20: Dependence of C and Gamma with rank score. The Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter Gamma. The four line graphs are for parameter C at 0.1, 1.0, 10.0 and 100.0. Rank 1 is for C =100 and gamma =0.001. The rank 16 or lowest rank is for C =0.1 and gamma = 0.001.



In Figures 6.21 and Figures 6.22, we see the dependence of C and Gamma with mean fit time. The Y axis represents the mean fit time in seconds. The general trend observed in these graphs is that as the parameter gamma increases the mean fit time increases while not actually resulting in an increase in accuracy. Rank 1 is for C =100 and gamma =0.001 also takes the least amount of fit time. The rank 16 or lowest rank is for C =0.1 and gamma = 0.001 also takes a low amount of train time. The general trend observed in these graphs is that as the parameter gamma increases the mean fit time increases while not actually resulting in an increase in accuracy. From the graph it is evident that the variable does no have a major impact on the fit time as parameter Gamma does. From a variation of gamma of 0.1 to 1, we see a great increase in the mean fit time.

Figure 6.21: Dependence of C and Gamma with mean fit time. The Y axis represents the mean fit time in seconds. The X axis contains the parameter Gamma. The four line graphs are for parameter C at 0.1, 1.0, 10.0 and 100.0.

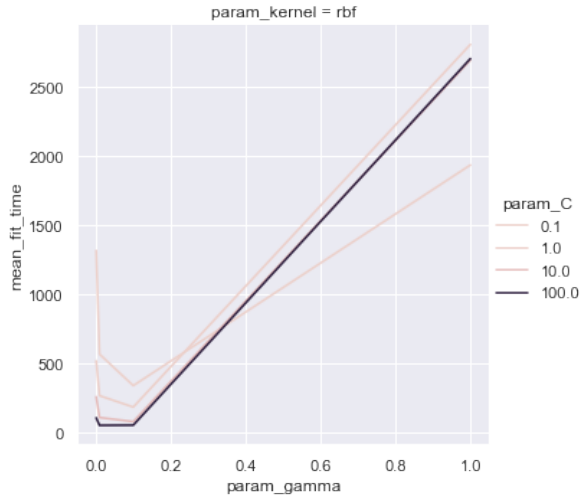
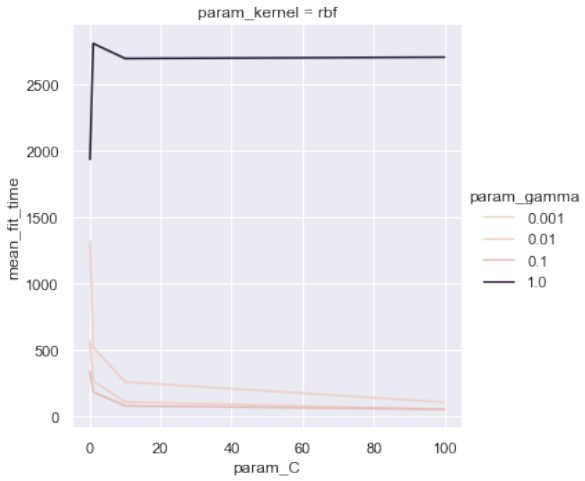
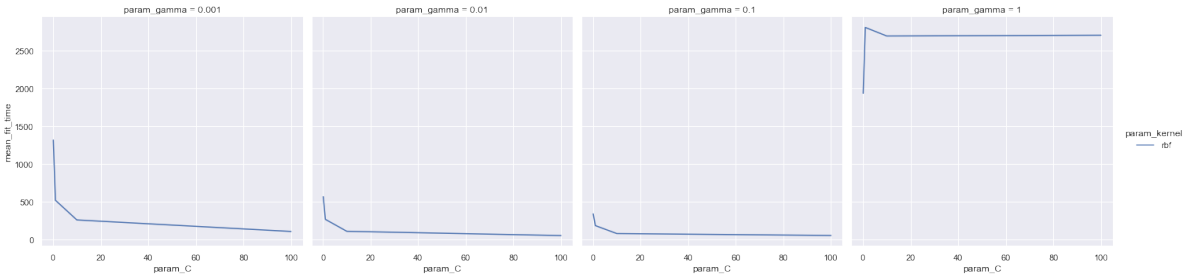


Figure 6.22: Dependence of C and Gamma with mean fit time. The Y axis represents the mean fit time in seconds. The X axis contains the parameter C. The four line graphs are for parameter Gamma at 0.001, 0.01, 0.1 and 1.0.



From Figure 6.23 we see that dependence of C and Gamma with mean fit time. The Y axis represents the mean fit time in seconds. The X axis contains the parameter C. The four line graphs are for parameter Gamma at 0.001, 0.01, 0.1 and 1.0. we see that the Rank 1 is for C =100 and gamma =0.001 also takes the least amount of fit time. The rank 16 or lowest rank is for C =0.1 and gamma = 0.001 also takes a low amount of train time. The general trend observed in these graphs is that as the parameter gamma increases the mean fit time increases while not actually resulting in an increase in accuracy. From the graph it is evident that the variable does no have a major impact on the fit time as parameter Gamma does. From a variation of gamma of 0.1 to 1, we see a great increase in the mean fit time.

Figure 6.23: Dependence of C and Gamma with mean fit time. The Y axis represents the mean fit time in seconds. The X axis contains the parameter C. The four line graphs are for parameter Gamma at 0.001, 0.01, 0.1 and 1.0.



From Figure 6.24 and 6.25 we see the dependence of C and Gamma with the test score. In Figure 6.24, the Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter C. The four line graphs are for parameter gamma at 0.001, 0.01, 0.1 and 1.0. Rank 1 is for C =100 and gamma =0.001. While in Figure 6.25 we see that the Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter Gamma. The four line graphs are for parameter C at 0.1, 1.0, 10.0 and 100.0. Rank 1 is for C =100 and gamma =0.001. The rank 16 or lowest rank is for C =0.1 and gamma = 0.001. From the graphs we learn that the rank 16 or lowest rank is for C =0.1 and gamma = 0.001.

Figure 6.24: Dependence of C and Gamma with rank score. The Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter C. The four line graphs are for parameter gamma at 0.001, 0.01, 0.1 and 1.0. Rank 1 is for C =100 and gamma =0.001.

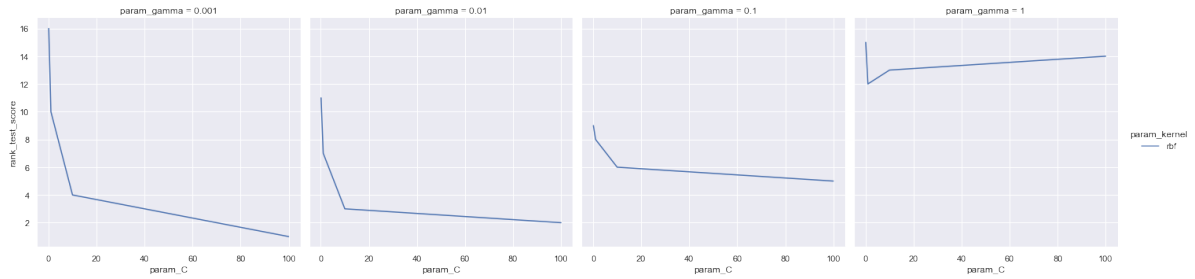
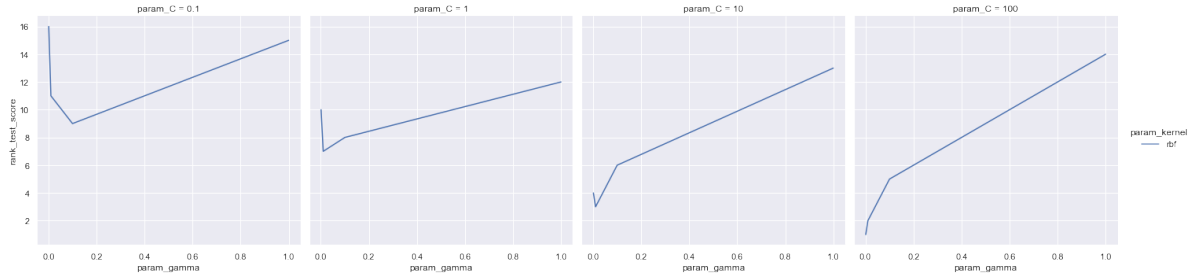


Figure 6.25: Dependence of C and Gamma with test score. The Y axis represents the rank score from 1 to 16. The lower the tank the more desirable the algorithm. The X axis contains the parameter Gamma. The four line graphs are for parameter C at 0.1, 1.0, 10.0 and 100.0. Rank 1 is for C =100 and gamma =0.001. The rank 16 or lowest rank is for C =0.1 and gamma = 0.001.



Encapsulated in Figure 6.26, we can see how the Variance and Bias is affected by changes in C and Gamma. The SVM algorithm has low bias and high variance, but this trade-off can be changed by increasing the C parameter that influences the error margin allowed in the training data which increases the bias but decreases the variance.

Figure 6.26: Dependence of Variance and Bias on C and Gamma

	Large Gamma	Small Gamma	Large C	Small C
Variance	Low	High	High	Low
Bias	High	Low	Low	High

6.1.1. Validation

Cross-validation is a scheme employed to validate the accuracy of the model. For both POLDER-3 and OMI when comparing the three algorithms of SVM, RF and KNN a seven fold cross-validation scheme was used on the default parameters. The three algorithms are compared and the best performing algorithm is chosen based on these results. The best fit for the hyperparameters are made using GridSearchCV method. To ensure that the best fit for the hyperparameters are chosen, we employ a five fold cross-validation technique. Five fold ensures that we are validating the technique while ensuring the time taken is not too high as is a case if a higher value of k is chosen for cross-validation. In the table 6.4 we can see the five fold cross-validation for selecting the best hyperparameters for POLDER-3. The scoring is done for all the five folds and then we calculate the mean and standard score. Finally, we calculate the rank based on these metrics, which decides the best performing hyperparameters which in our case are as given below:

Listing 6.4: Best fit parameters from GridSearchCV

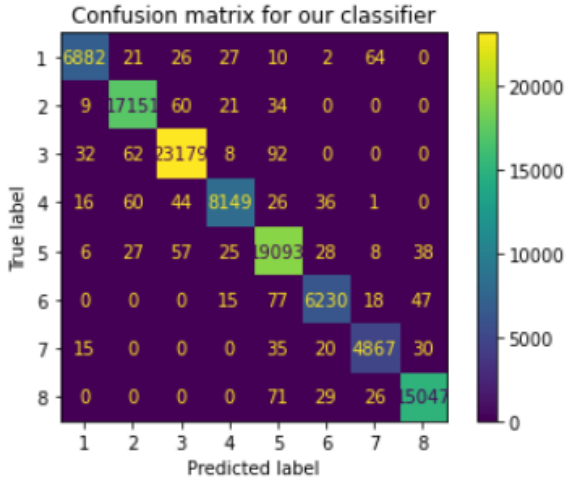
```
SVC(C=100, Gamma=0.001, kernel='rbf')
```

Table 6.4: Cross-validation Table for POLDER-3 data

C	gamma	kernel	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank
0.1	1	rbf	0.939916	0.939220	0.937937	0.937663	0.939726	0.938892	0.000925	15
0.1	0.1	rbf	0.980197	0.979417	0.979691	0.978744	0.980049	0.979620	0.000516	9
0.1	0.01	rbf	0.972220	0.971020	0.971420	0.970915	0.971189	0.971353	0.000466	11
0.1	0.001	rbf	0.922701	0.921437	0.921647	0.921837	0.921479	0.921820	0.000463	16
1	1	rbf	0.957026	0.956878	0.955426	0.956225	0.954626	0.956036	0.000904	12
1	0.1	rbf	0.987920	0.987267	0.987036	0.986510	0.986889	0.987124	0.000468	8
1	0.01	rbf	0.988762	0.987646	0.988656	0.987288	0.988383	0.988147	0.000580	7
1	0.001	rbf	0.973146	0.972052	0.972788	0.971946	0.972030	0.972393	0.000484	10
10	1	rbf	0.953827	0.954037	0.952858	0.952542	0.952374	0.953128	0.000678	13
10	0.1	rbf	0.989772	0.989540	0.988509	0.989014	0.988951	0.989157	0.000449	6
10	0.01	rbf	0.994002	0.992929	0.992697	0.992403	0.992760	0.992958	0.000549	3
10	0.001	rbf	0.990067	0.988909	0.989456	0.988720	0.989498	0.989330	0.000477	4
100	1	rbf	0.952186	0.953384	0.951722	0.951553	0.950943	0.951957	0.000817	14
100	0.1	rbf	0.989772	0.989688	0.988193	0.988951	0.989246	0.989170	0.000572	5
100	0.01	rbf	0.995033	0.994528	0.994212	0.994423	0.994360	0.994511	0.000280	2
100	0.001	rbf	0.995391	0.994212	0.994507	0.994297	0.994844	0.994650	0.000430	1

Once the hyper-parameters are tuned and the model implementation is performed, it is time to interpret and understand the results obtained from the algorithm. From figure 6.27, we can see the confusion matrix for SVM implementation. It is an array of size n x n, where n refers to the number of output label classes. For our application this number is eight since we have eight aerosol clusters that we are classifying. Here, the rows and the columns represent the class labels. In each cell of the confusion matrix $C_{i,j}$ we have the value equal to the number of observations known to be in group i and predicted to be in group j. The function to create the confusion matrix takes as input X_test and y_test. Recall that X_test refers to the 30 % of the dataset (101,821) used for testing the dataset. Recall from Chapter 5 that from [16] we get the clusters where 1: Smoke, 2:Mixed Smoke, 3:Marine, 4:Urban-Industrial, 5:dusty Smoke, 6:Marine Dust, 7:dust and 8:Polluted Dust. In the confusion matrix given below, the Y axis denotes the True label and the X axis denotes the Predicted label.

Figure 6.27: Confusion Matrix for the 8 cluster classes



From the confusion matrix, we can see that for a total of 7032 datapoints Class label 1 : Smoke was correctly classified as Smoke for 6882 datapoints. It was incorrectly labelled as label 2:Mixed Smoke for 21 data points, incorrectly labelled as label 3:Marine for 26 datapoints, incorrectly labelled as label 4:Urban Industrial for 27 datapoints, incorrectly labelled as label 5:dusty Smoke for 10 datapoints, incorrectly labelled as label 6:MarineDust for 2 datapoints, and incorrectly labelled as label 7:dust for 64 datapoints. The algorithm did not misclassify any class 1 label:Smoke as class 8 label:Polluted Dust.

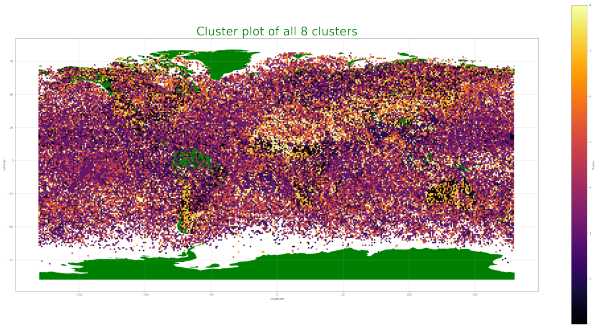
This gives us a f1-score of 0.98 as seen in figure 6.28. Classes 4:Urban Industrial, 6:Marine Dust, 7:dust have a similar precision of 0.98. The classes 2:Mixed Smoke, 3:Marine, 5:Dust Smoke and 8:Polluted Dust have a higher f-1 score of 0.99. We saw in Chapter 5 methodology that the f-1 score is the harmonic mean of precision and recall. The score corresponding to each class tells the accuracy of the classifier in classifying the data points in the particular class compared other classes. The support refers to the number of samples of the true predictions in detected in the particular class. In the figure below, we see the precision, recall and f-1 score along with the support for the best fit of SVM algorithm.

Figure 6.28: Precision, Recall and F-1 Score

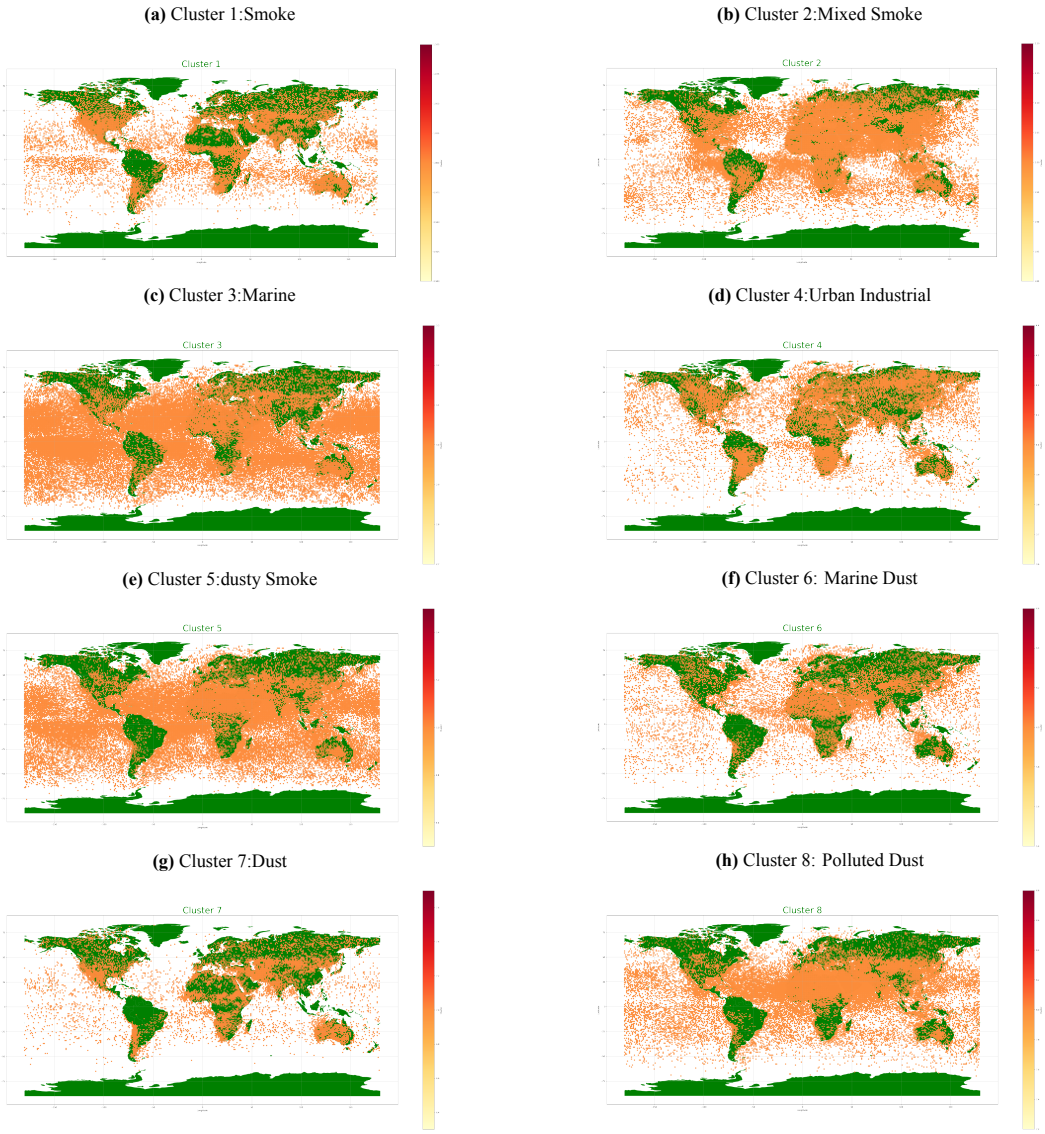
	precision	recall	f1-score	support
1	0.99	0.98	0.98	7032
2	0.99	0.99	0.99	17275
3	0.99	0.99	0.99	23373
4	0.99	0.98	0.98	8332
5	0.98	0.99	0.99	19282
6	0.98	0.98	0.98	6387
7	0.98	0.98	0.98	4967
8	0.99	0.99	0.99	15173
accuracy			0.99	101821
macro avg	0.99	0.98	0.99	101821
weighted avg	0.99	0.99	0.99	101821

In a nutshell, to sum up the results we found out that the algorithm SVM outperforms the algorithms KNN and RF for the POLDER-3 data set. Furthermore, tuning the hyper parameters further increases the accuracy of the algorithm. Using the supervised learning algorithm we can classify the aerosol classes from Figure 6.29 as shown below into Figure 6.30 with 99% accuracy. We arrived at these results using a series of assumptions and considerations, it is worth noting that any variation in these assumptions and considerations might lead to a varied results. These will be discussed in the discussion section Section 6.1.2.

Figure 6.29: Plot of all eight clusters



Since plotting all the eight cluster gets too crowded on the world map, we now visualize each cluster on the world map.



(a) Cluster 1:Smoke **(b)** Cluster 2:Mixed Smoke

(c) Cluster 3:Marine **(d)** Cluster 4:Urban Industrial

(e) Cluster 5:dusty Smoke **(f)** Cluster 6: Marine Dust

(g) Cluster 7:Dust **(h)** Cluster 8: Polluted Dust

Figure 6.30: Aerosol Clusters throughout the year. Here the eight clusters are plotted on the world map. Cluster 1 is smoke, 2:Mixed Smoke, 3:Marine, 4:Urban Industrial, 5:dusty Smoke, 6:Marine Dust, 7:dust and 8:Polluted Dust.

6.1.2. Discussion on POLDER-3

Our requirement when we started this study was to meet an accuracy of 90% on aerosol classification using supervised learning. Ideally, when working on a classification problem like ours the best score is 100% accuracy. However, this score is impossible to achieve as an upper bound. Predictive modelling tends to inherently have prediction error from a range of sources including incompleteness of the data sample, noise in the data and stochastic nature of the modeling algorithm. Thus, we have set the baseline to be 90% accuracy. Using SVM on POLDER-3 dataset with hyperparameter tuning we reached an overall accuracy of 99%, precision of 99%, recall of 98% and f1-score of 99% on POLDER-3 dataset for the eight classes of aerosols. Thus, at first glance we have met the 90% accuracy requirement. However, as previously stated there were a few assumptions and considerations we made to arrive at these results. Changing these assumptions and considerations would change the results. Firstly, in our study we only used a selected group of combination of hyperparameters in our GridSearchCV algorithm, if we were to do a more comprehensive and exhaustive search over a larger hyperparameter range, it might result in an even greater accuracy than achieved here. However, having already met the 90% requirement we set out to achieve, we did not focus on further optimizing the result. Secondly, we used the negative root mean square metric as the scoring metric to choose between the three algorithms RF, KNN and SVM as was seen in Figure 6.15. We saw from the Figure that SVM had the least error and thus we chose it for further optimizing. It is worth noting that when these three algorithms were compared they were run on the default settings in the Scikit-learn

python library. RF was a very close in error to SVM with the difference being 0.05 root mean squared error. Had a different set of hyperparameters been chosen, RF might have been the best performing model with least error. We can conclude here that tuning the other algorithms might have lead to different results than what was achieved using the default hyperparameters. This leads us to believe that optimization bias has occurred in choosing SVM as the best performing model for POLDER-3 dataset.

In this study we classified eight types of aerosols classes using microphysical parameters and cluster labels as input. In this study we used the same microphysical features as used by de Bakker. We used four features: fraction of spheres - sphere_frac_coarse, real refractive index - m_r_fine, single scattering albedo - SSA_490 nm, and angstrom exponent- ae. The reader is referred to Section 5.7.1 for more information on these microphysical features. de Bakker used these features to find eight aerosol clusters

1. Smoke
2. Mixed-Smoke
3. Marine
4. Urban-Industrial
5. Dusty Smoke
6. Marine Dust
7. Dust
8. Polluted Dust

in the data using unsupervised learning. It is also worth noting that the difference in the definition of these clusters is possible. De Bakker [16] had manually labelled these algorithms based on the region of occurrence, and thus it is possible that there was a degree of overlap between the said cluster name. The difference between our study and that of de Bakker is that we use supervised learning to categorize the aerosol classes while he ran unsupervised learning to assign clusters. The training of his algorithms took well over 24 hours. Once his best fitting model was trained, we tested it on the test dataset which generated aerosol type data labels as input for our supervised learning techniques. The motivation to use supervised learning over unsupervised learning is two fold, one it is faster than unsupervised learning for aerosol classification and second once the model is trained it can be ported to different subsets of unseen data for example different years to find the aerosol types. Firstly, unsupervised learning means that to find aerosol clusters in a dataset we needed to cluster the whole dataset without any knowledge of how many clusters may be present and use a exploratory study to find the number of clusters that might be present. This exploratory study is very time consuming as was evident when running de Bakker's code to find the clusters and generate the output labels. Once we get these output labels, we need to manually label them. With supervised learning we can use baseline studies to get the number of cluster labels and then train further models faster on these baselined dataset.

In the next section we look at the implementation and results on OMI data. The reader is referred to Chapter 7 for an overall discussion of the methodology and the obtained results on OMI and POLDER-3.

6.2. OMI Data

The OMI instrument is a nadir-viewing wide-field-imaging spectrometer onboard the earth observing system Aura satellite. It provides a daily coverage of the globe at the nadir resolution of 24 x $13km^2$ [7] [1]. OMI measures nitrogen dioxide, sulphur dioxide, bromine oxide, OClO, and aerosol characteristics, which constitute the key air quality components. It measures the sunlight incident directly and back-scattered in the ultraviolet-visible spectrum ranging from 270 nm to 500 nm.

Table 6.5: OMI instrument specifications [48]

Item	Parameter
Visible	350 - 500 nm
UV	UV-1: 270 to 314 nm, UV-2: 306 to 380 nm
Spectral resolution	1.0 - 0.45 nm FWHM
Spectral sampling	2-3 for FWHM
Telescope FOV	114 (2600 km on ground)
IFOV	3 km, binned to 13 x 24 km
Detector	CCD: 780 x 576 (spectral x spatial) pixels
Mass	65 kg
Duty cycle	60 minutes on daylight side
Power	66 watts
Data rate	0.8 Mbps (average)

In Section 5.2, we saw how OMI data was collected. We mentioned that there were two aerosol data products called OMI AERO data and OMI AEROUV data. These two datasets have different variables as shown in Table 6.6. We decided to use the OMI AERO data for this thesis since that dataset has aerosol type labels present in the data.

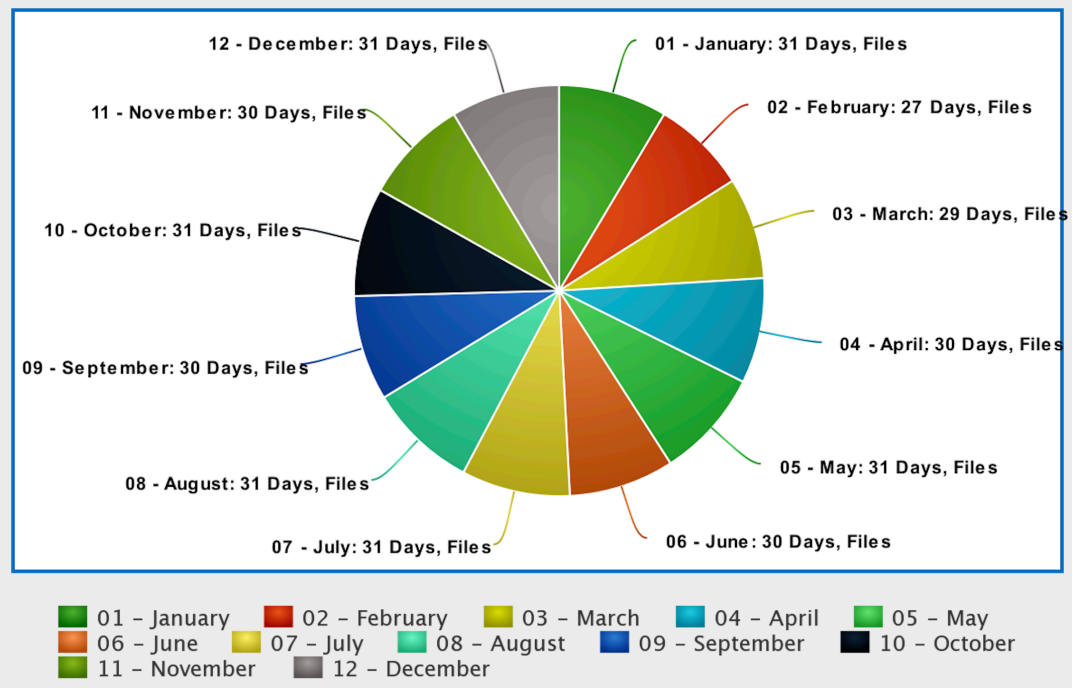
Table 6.6: Difference between aerosol products from OMI

OMI O Data Products	OMI AERO UV data Products
2) 'AbsorbingAerosolOpticalThicknessMW', 3) 'AerosolModelMW', 4) 'AerosolOpticalThicknessMW', 5) 'AerosolOpticalThicknessPassedThresholdMean', 6) 'AerosolOpticalThicknessPassedThresholdStd', 7) 'Latitude', 8) 'Longitude', 9) 'SingleScatteringAlbedoMW', 10) 'SingleScatteringAlbedoPassedThresholdMean', 11) 'SingleScatteringAlbedoPassedThresholdStd', 12) 'SolarZenithAngle', 13) 'TerrainReflectivity', 14) 'UVAerosolIndex', 15) 'VISAerosolIndex', 16) 'ViewingZenithAngle'	1) 'CloudFraction', 2) 'CloudOpticalDepth', 3) 'FinalAerosolAbsOpticalDepth354', 4) 'FinalAerosolAbsOpticalDepth388', 5) 'FinalAerosolAbsOpticalDepth500', 6) 'FinalAerosolOpticalDepth354', 7) 'FinalAerosolOpticalDepth388 8) 'FinalAerosolOpticalDepth500', 9) 'FinalAerosolSingleScattAlb354', 10) 'FinalAerosolSingleScattAlb388', 11) 'FinalAerosolSingleScattAlb500', 12) 'UVAerosolIndex'

The OMI data is split into 362 files. Some files for days are missing thus totalling 362 instead of 365 signalling the duration of the entire year. Below, in Figure 6.31 the distribution of these files can be seen. Each file is stored with the he5 extension. Consider a he5 file to be a non - readable json file that contains key-value pairs. The keys and values define the observations (date, cloud coverage, data quality, etc.), the raw data , and the processed data. The he5 files are read with Python using the h5py module.

[1] `https://aura.gsfc.nasa.gov/omi.html`

Figure 6.31: 362 daily OMI Level 3 files distribution



The OMI data for aerosols mainly consist of OMAERO: (Multi wavelength for better characterization over the oceans) and OMERUV (Two near UV wavelength for over the land). The OMAERO data consists of Aerosol Model (16 bit integer value where each bit indicates a characteristic of the model.) Sample data:

Lon	Lat	Aerosol Model Value
x	y	1211
		abcd

The Most significant bit (a) represents the aerosol type: The four aerosol models are:
1 - WA – Weakly absorbing
2 – BB – Biomass Burning
3 – DD – Desert Dust
4 - VO – Volcanic Aerosols

These are further subdivided into subtypes depending on their size distribution, refractive index and vertical distribution giving each classification a unique four digit identifier (abcd as referred before)[67].

Figure 6.32: Dataframe containing variables extracted for OMI data. Here there are six columns containing Latitude, Longitude, Aerosol Model, Aerosol Optical Thickness, Single Scattering Albedo and UV Aerosol Index for a single day in Level 3(L3) file.

	Latitude	Longitude	Data Fields/AerosolModelMW	Data Fields/AerosolOpticalThicknessMW	Data Fields/SingleScatteringAlbedoMW	Data Fields/UVAerosolIndex
0	-78.375	-164.875	1213	1	1	-1.22
1	-78.375	-164.625	1213	1	1	-1.22
2	-78.375	-164.375	1213	1	1	-1.22
3	-78.375	-164.125	1213	1	1	-1.22
4	-78.375	-163.875	1213	1	1	-1.22
...	...	...	...	...	...	...
612710	67.625	33.375	1211	5	5	0.29
612711	67.625	33.625	1211	5	5	0.29
612712	67.875	-5.625	1212	5	5	-0.41
612713	67.875	-5.375	1212	5	5	-0.41
612714	67.875	-5.125	1212	5	5	-0.41

612715 rows × 6 columns

Recall that as mentioned for description for Figure 5.7, a correlation matrix is a table that presents the coefficients of correlation between variables. Each cell in the table contains the information about the correlation between two variables. The value of correlation ranges from 0 to 1, with 0 being no correlation between the features to 1 being highly correlated. Correlation gives us a clear understanding of the importance of the features before building the machine learning model. With the help of the correlation analysis we can also check if we have multicollinearity i.e. high correlation between the variables, which is undesirable in the role of building a machine learning model. From Figure 6.33, we can deduce the following points:

Figure 6.33: Feature correlation matrix for the three features and Aerosol Model.

	Latitude	Longitude	Data Fields/AerosolModelMW	AerosolOpticalThickness	SingleScatteringAlbedo	Data Fields/UVAerosolIndex
Latitude	1.000000	0.122196	0.339963	0.013464	-0.369293	0.103684
Longitude	0.122196	1.000000	0.159779	0.054072	-0.240351	0.048899
Data Fields/AerosolModelMW	0.339963	0.159779	1.000000	0.161663	-0.684073	0.531690
AerosolOpticalThickness	0.013464	0.054072	0.161663	1.000000	-0.224647	0.138232
SingleScatteringAlbedo	-0.369293	-0.240351	-0.684073	-0.224647	1.000000	-0.323823
Data Fields/UVAerosolIndex	0.103684	0.048899	0.531690	0.138232	-0.323823	1.000000

First, we begin exploring the OMI 2006 dataset by plotting the four major aerosol classes on the world map. They actually have 50 overall classes, of which the four major types are: Weakly absorbing, Biomass Burning, Desert Dust, and Volcanic Aerosols. These are further classified into smaller subsets of each type depending on the difference on microphysical parameters. In the 2006 OMI data that these graphs are maps for the distribution of the aerosol types is as follows: Total Data points: 4061606

From 6.34, 6.35 and 6.36, we can see the seasonal changes across various months in the distribution of the four aerosol classes of weakly absorbing aerosol, biomass burning aerosol, desert dust aerosol, and volcanic aerosol. Across the summer months of June, July and August, and the autumn months of September, October and November there is a greater distribution of biomass burning aerosol. Across the winter months of December, January and February and the spring months of March, April and May, there is a smaller distribution of biomass burning aerosol. Overall, it can be observed that biomass burning aerosol has shown a downward trend in the Australian subcontinent from September to December.

Figure 6.34: Plotting four major aerosol types for OMI from January to April. 1 - WA – Weakly absorbing 2 – BB – Biomass Burning 3 – DD – Desert Dust 4 - VO – Volcanic Aerosols

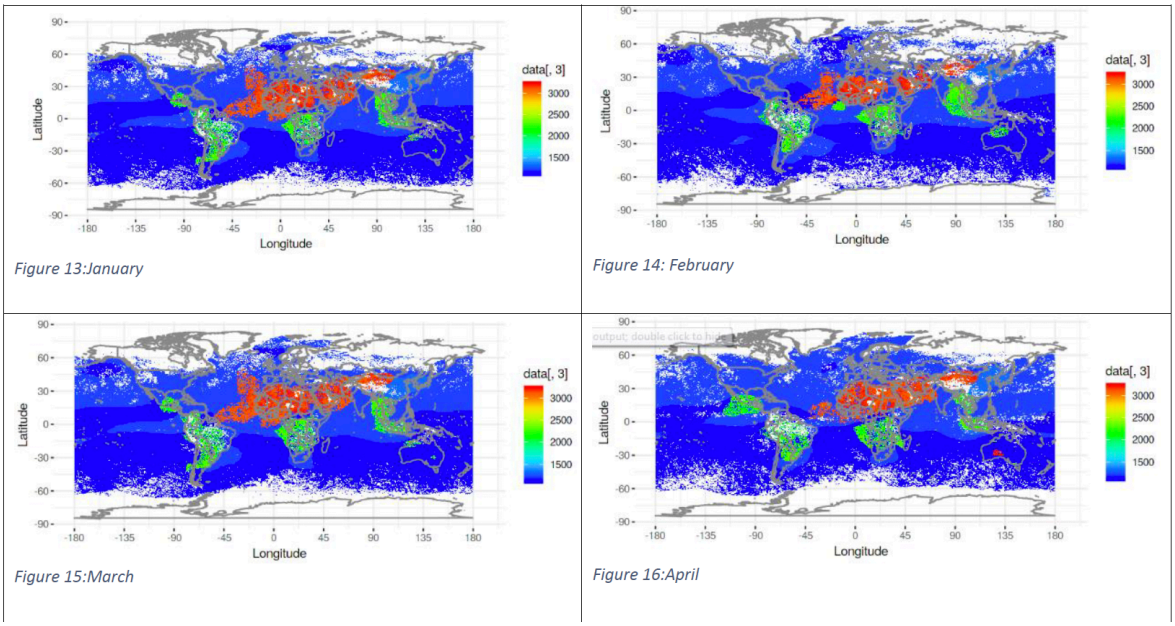


Figure 13:January

Figure 14: February

Figure 15:March

Figure 16:April

Figure 6.35: Plotting four major aerosol types for OMI from May to August. 1 - WA – Weakly absorbing 2 – BB – Biomass Burning 3 – DD – Desert Dust 4 – VO – Volcanic Aerosols

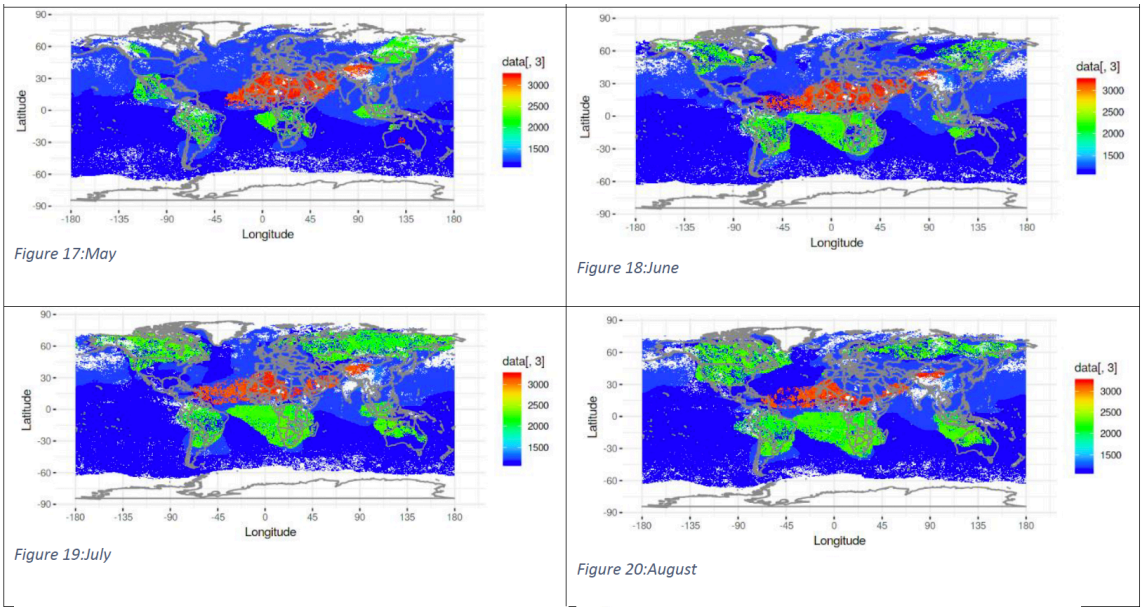


Figure 17:May

Figure 18:June

Figure 19:July

Figure 20:August

Figure 6.36: Plotting four major aerosol types for OMI from September to December. 1 - WA – Weakly absorbing 2 – BB – Biomass Burning 3 – DD – Desert Dust 4 – VO – Volcanic Aerosols

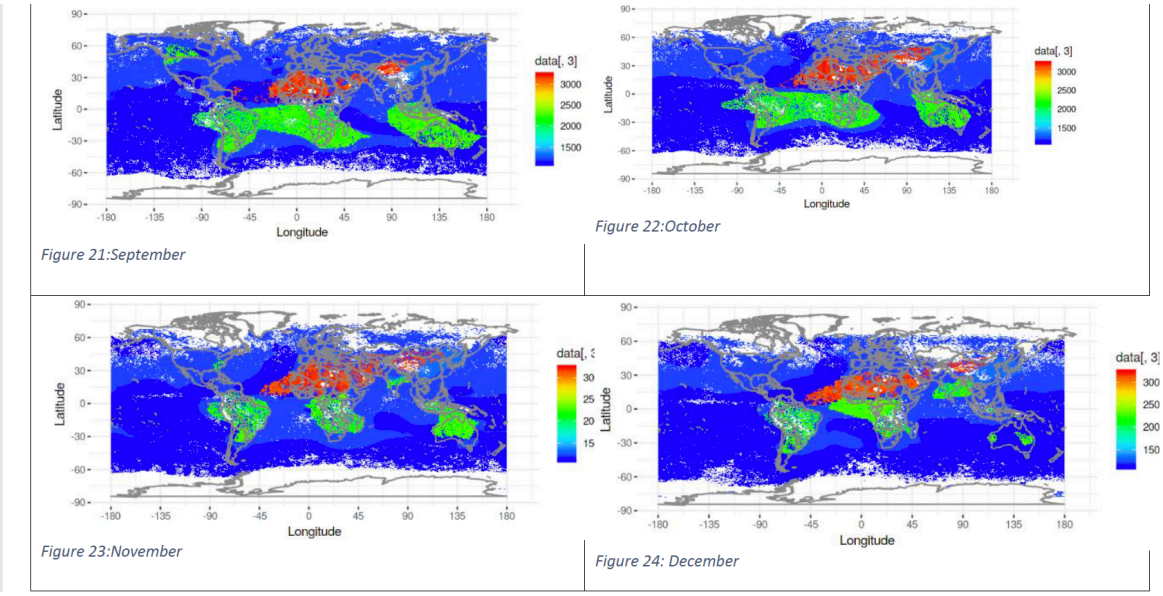


Figure 21:September

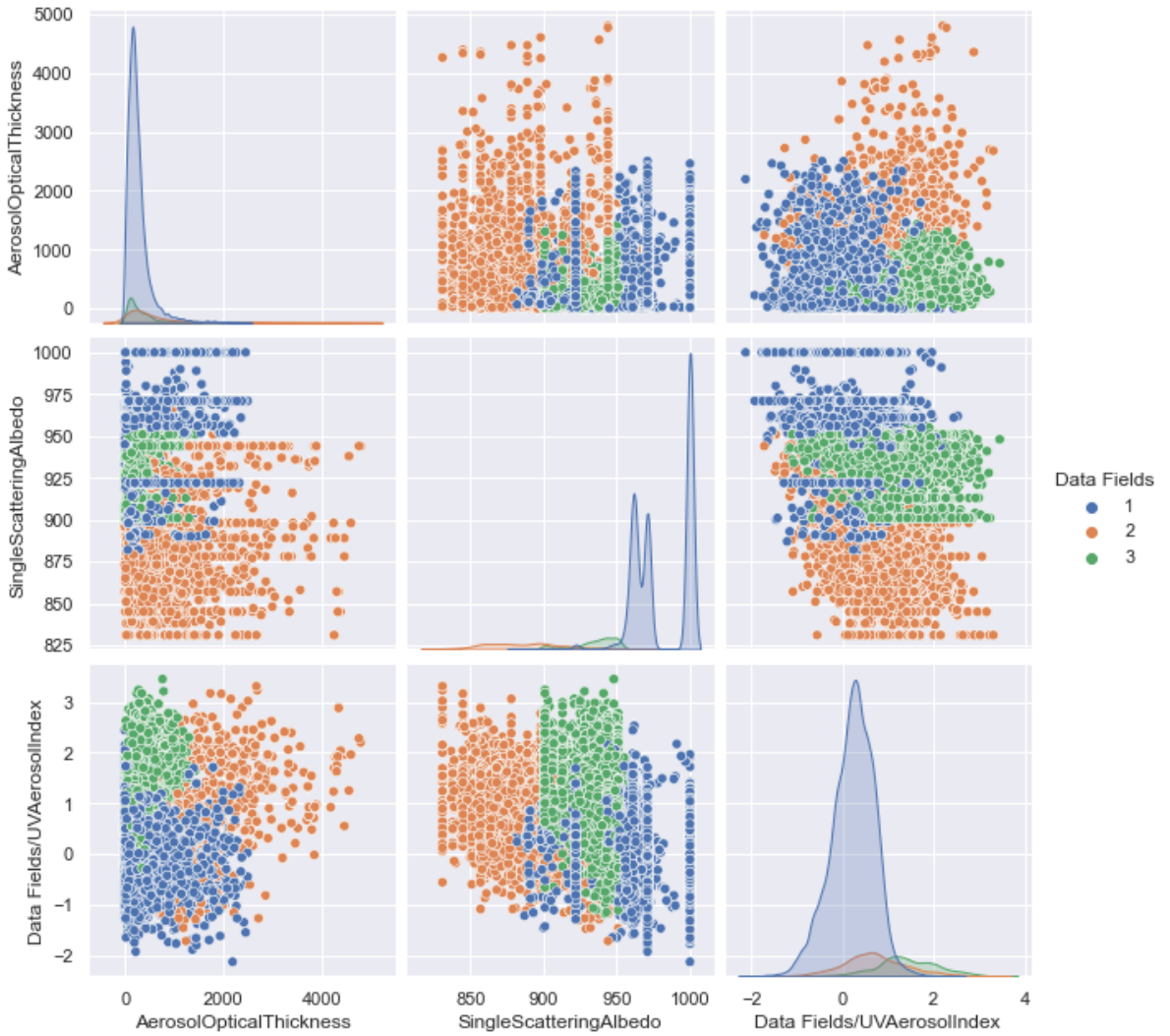
Figure 22:October

Figure 23:November

Figure 24: December

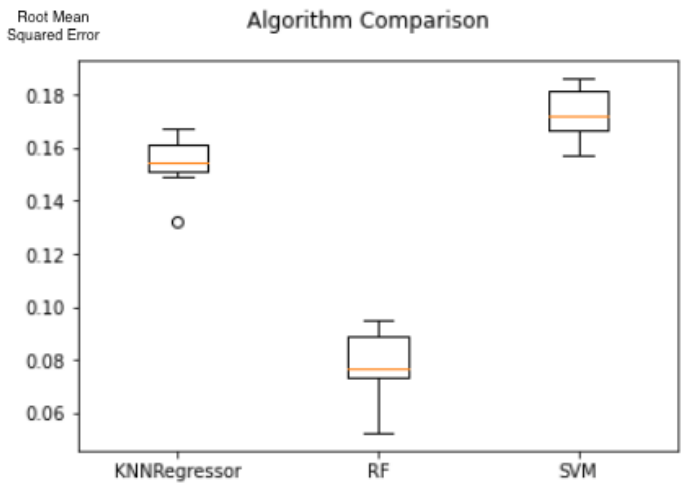
Recall that we found four classes for OMI 2006 data, namely, Weakly absorbing, Biomass burning, Desert Dust and Volcanic Aerosols. OMI generated a total of 4061606 datapoints over the entire year in terms of daily files. To limit the training time we used averaging to generate a sampled dataset. We used these labels as ground truth for training the OMI data. There were only two data points for Volcanic Aerosols in the training dataset and thus this algorithm effectively only classifies between the three classes: Weakly Absorbing, Biomass burning and Desert Dust. We have selected the microphysical parameters Aerosol Optical Thickness, Single Scattering Albedo and UV Aerosol Index to be trained in the supervised learning models. The relationship between these microphysical variables and the clusters are present in Figure 6.37.The pair plot in Figure 6.37 is used to understand the relationship between the hyperparameters derived from OMI data. Here the feature label 1,2 and 3 refer to Weakly absorbing, Biomass burning and DesertDust respectively.

Figure 6.37: Feature Pair Plot



The test set has 16314 datapoints with three class labels. The next step of implementing the supervised learning models RF, KNN and SVM on the OMI data is presented below. From the figure 6.38, we can see that the algorithm RF has the least value for the Root Mean Squared Error, and thus can conclude that for non-optimized parameters, RF is the best performing algorithm. Note that the lower the negative mean squared error the better performing the algorithm. The least performing algorithm is SVM for OMI dataset. Interestingly, SVM was the best performing algorithm for the POLDER-3 dataset. Thus, it can be concluded that there is no one size fits all solution in machine learning instead it depends on the individual case and depends on the dataset and the classification task at hand.

Figure 6.38: Root Mean Squared Error results of KNN, RF and SVM on OMI data



Now, we try to improve the accuracy of the RF algorithm using hyperparameter tuning. In Scikit-learn, there are four hyperparameters for RF that can be tuned. In the case of the RF, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node. Scikit-Learn package implements a default set of hyperparameters for all models, but these are not always the most optimal set of hyperparameters. The best hyperparameters are not known ahead of time and tuning the model results in trial-and-error approach to detect the best fit. Now, to find the best fit we try different combinations to evaluate the performance of each model.
In Chapter 5, we saw the implementation details for RF in section 5.4. We saw the various parameters and their definitions. From the documentation on RF in Scikit-Learn [61], we learn that the number of trees in the forest (n estimators) and the number of features considered for splitting at each leaf node (max features) are important. Along with these, we also train max depth and criterion. We use GridSearchCV to run fits with changing the parameters given below:

Listing 6.5: GridSearchCV selected for Random Forest

```
param_grid = {
    'n_estimators': [200, 500],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,5,6,7,8],
    'criterion' :['gini', 'entropy']
}
```

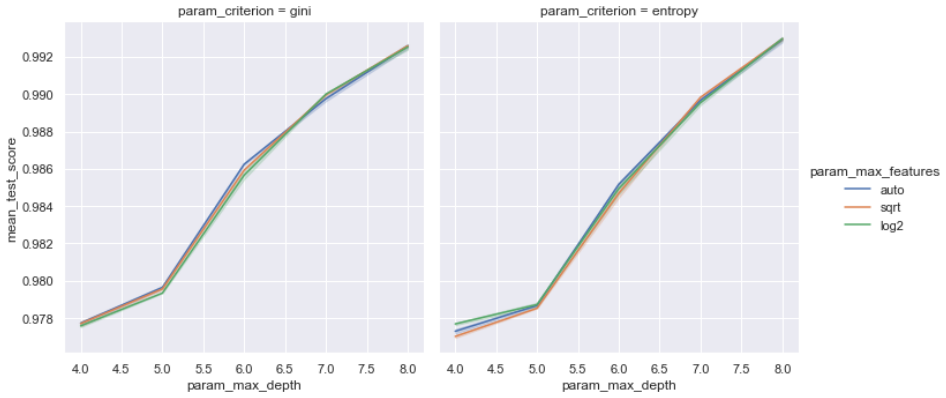
After fitting 5 folds for each of 60 candidates, totalling 300 fits we get the best parameter fit given below:

Listing 6.6: Best fit parameters from GridSearchCV

```
RandomForestClassifier(criterion='entropy', max_depth=8, max_features='sqrt',
    n_estimators=200)
```

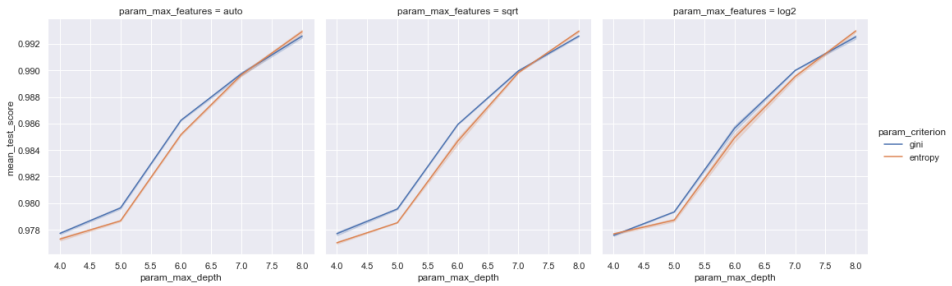
In Figures 6.39, 6.40, 6.41, we see how the mean test score varies with different combinations of the hyperparameters. We can clearly see that the the hyperparameter setting shown in the listing above are the best performing hyperparameters maximizing accuracy for the RF algorithm. In Figure 6.39 we see how the mean test score varies with changing max depth parameter for different parameter criterion 'gini' and 'entropy'. From the graphs it is evident that as the max depth parameter is increased the accuracy increases. The maximum accuracy is achieved for max_depth =8. Secondly, the parameter entropy shows the slightly higher maximum accuracy for fixed max_features. Among the three param_max_features, 'sqrt' had the maximum accuracy.

Figure 6.39: Mean test score variation with changing max depth parameter for different parameter criterion 'gini' and 'entropy'. The Y axis represents the mean test score accuracy of the five folds of cross-validation. The X axis depicts the parameter maximum depth which is varied from 4,5,6,7,8. The two column represent the parameter criterion and entropy as explained in Chapter 5. Line graphs are drawn for three parameters 'auto', 'sqrt' and 'log2'.



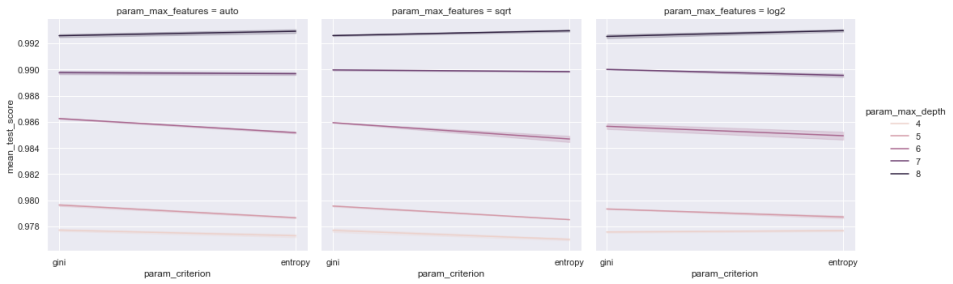
From Figure 6.40 we see how the mean test score varies with different parameter max depth for changing parameter max features as 'auto', 'sqrt' and 'log2'. The Y axis represents the mean test score accuracy of the five folds of cross-validation. The X axis depicts the parameter maximum depth which is varied from 4,5,6,7,8. The three columns represent the three parameters of max_features 'auto', 'sqrt', and 'log2'. In each graph, the two ling plots represent the criterion 'gini' and 'entropy'. From the graphs we can see that 'gini' initially has a higher accuracy than 'entropy' for lower values of max_depth. Then for max_depth = 8, the accuracy of 'entropy' is highest, making it the best performing hyperparameter.

Figure 6.40: Analyzing the optimization results. The Y axis represents the mean test score accuracy of the five folds of cross-validation. The X axis depicts the parameter maximum depth which is varied from 4,5,6,7,8. The X axis depicts the parameter maximum depth which is varied from 4,5,6,7,8. The three columns represent the three parameters of max_features 'auto', 'sqrt', and 'log2'. In each graph, the two ling plots represent the criterion 'gini' and 'entropy'.



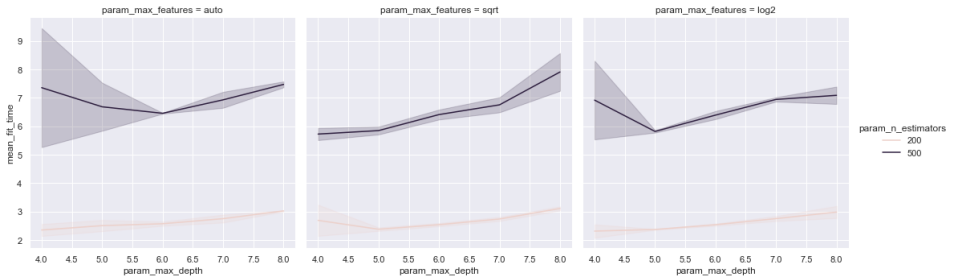
From Figure 6.41 we can see the mean test score variation for two parameter criterion gini and entropy for the parameter max features as 'auto', 'sqrt' and 'log2'. The Y axis represents the mean test score accuracy of the five folds of cross-validation. The X axis shows the categorical variation between the parameters 'gini' and 'entropy'. The three columns represent the parameter max features 'auto', 'sqrt', and 'log2'. In each plot the line graphs represent the max depth varied from 4,5,6,7,8. From these graphs we can infer that the accuracy is maximum for max depth =8. Futhermore, for all max depth except 8 , the criterion 'gini' yields a higher accuracy than 'entropy'. Finally, the second column with parameter max feature 'sqrt' with parameter criterion 'entropy' and max depth = 8 has the highest accuracy compared to all other settings.

Figure 6.41: Analyzing the optimization results. The Y axis represents the mean test score accuracy of the five folds of cross-validation. The X axis shows the categorical variation between the parameters 'gini' and 'entropy'. The three columns represent the parameter max features 'auto', 'sqrt', and 'log2'. In each plot the line graphs represent the max depth varied from 4,5,6,7,8.



In Figure 6.42, we see how the mean fit time varies for different setting of the parameters. It is evident that the settings giving the highest accuracy also take the highest time to fit. The general trend that is visible from these graphs is that n estimator =500 takes longer time to fit when compared to n estimator n = 200. In the second column for n estimator = 500, the mean fit time gradually increases with increasing max depth with the maximum time being taken for max depth = 8.

Figure 6.42: Analyzing the optimization results - 4. The Y axis contains the mean fit time taken over the five folds of cross-validation. The X axis the parameter max depth varied over 4,5,6,7,8. The three columns represent the parameter max features 'auto', 'sqrt' and 'log2'. In each plot the two line graphs show the two parameters of n estimators as 200 and 500.



6.2.1. Validation

Cross-validation is a scheme employed to validate the accuracy of the model. For both POLDER-3 and OMI when comparing the three algorithms of SVM, RF and KNN a seven fold cross-validation scheme was used on the default parameters. The three algorithms are compared and the best performing algorithm is chosen based on these results. The best fit for the hyperparameters are made using GridSearchCV method. To ensure that the best fit for the hyperparameters are chosen, we employ a five fold cross-validation technique. Five fold ensures that we are validating the technique while ensuring the time taken is not too high as is a case if a higher value of k is chosen for cross-validation. In the table 6.7 we can see the five fold cross-validation for selecting the best hyperparameters for OMI. The scoring is done for all the five folds and then we calculate the mean and standard score. Finally, we calculate the rank based on these metrics, which decides the best performing hyperparameters which in our case are as given below:

Listing 6.7: Best fit parameters from GridSearchCV

```
RandomForestClassifier(criterion='entropy', max_depth=8, max_features='sqrt',
    n_estimators=200)
```

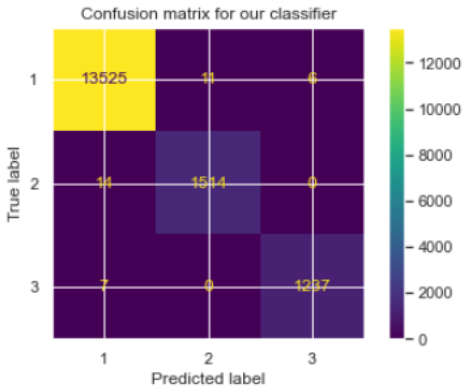
Table 6.7: Cross-validation Table for OMI data

criterion	max_depth	max_features	n	split0_score	split1_score	split2_test_score	split3_score	split4_score	mean_score	std$_score$	rank_score
gini	4	auto	200	0.975831	0.979509	0.978589	0.973857	0.981214	0.977800	0.002631	50
gini	4	auto	500	0.975437	0.978852	0.978721	0.974908	0.980294	0.977642	0.002098	53
gini	4	sqrt	200	0.975568	0.979115	0.977144	0.975039	0.980951	0.977564	0.002210	55
gini	4	sqrt	500	0.975568	0.979640	0.978195	0.975171	0.980557	0.977826	0.002146	49
gini	4	log2	200	0.975174	0.978064	0.978195	0.974645	0.982133	0.977642	0.002673	52
gini	4	log2	500	0.974780	0.978852	0.978064	0.974645	0.981214	0.977511	0.002509	56
gini	5	auto	200	0.976750	0.980034	0.981216	0.976747	0.983841	0.979718	0.002720	37
gini	5	auto	500	0.976882	0.980560	0.980297	0.976484	0.983579	0.979560	0.002620	39
gini	5	sqrt	200	0.977013	0.979771	0.980428	0.976879	0.983973	0.979613	0.002605	38
gini	5	sqrt	500	0.976882	0.980822	0.980166	0.975959	0.983710	0.979508	0.002804	40
gini	5	log2	200	0.977407	0.979903	0.980428	0.975565	0.983316	0.979324	0.002657	42
gini	5	log2	500	0.976356	0.980166	0.980560	0.975828	0.983841	0.979350	0.002955	41
gini	6	auto	200	0.984500	0.986076	0.988047	0.983973	0.988833	0.986286	0.001907	25
gini	6	auto	500	0.984106	0.987127	0.988704	0.982396	0.988702	0.986207	0.002539	26
gini	6	sqrt	200	0.983187	0.987390	0.987390	0.982922	0.988702	0.985918	0.002388	28
gini	6	sqrt	500	0.983318	0.986076	0.988178	0.983710	0.988439	0.985944	0.002150	27
gini	6	log2	200	0.983975	0.986733	0.987915	0.980557	0.988045	0.985445	0.002849	30
gini	6	log2	500	0.983187	0.986996	0.987653	0.983053	0.988439	0.985866	0.002288	29
gini	7	auto	200	0.988047	0.989492	0.991199	0.989359	0.990016	0.989622	0.001021	22
gini	7	auto	500	0.988572	0.990411	0.990937	0.988965	0.990541	0.989885	0.000936	17
gini	7	sqrt	200	0.989360	0.990674	0.990805	0.989096	0.990147	0.990017	0.000685	13
gini	7	sqrt	500	0.989098	0.990017	0.990805	0.989228	0.990410	0.989911	0.000662	16
gini	7	log2	200	0.988572	0.989886	0.991331	0.989490	0.990673	0.989990	0.000952	15
gini	7	log2	500	0.988966	0.990805	0.991068	0.988965	0.990279	0.990017	0.000895	13
gini	8	auto	200	0.991199	0.992776	0.993958	0.992643	0.992906	0.992696	0.000882	7
gini	8	auto	500	0.991068	0.992776	0.993432	0.992249	0.992775	0.992460	0.000791	11
gini	8	sqrt	200	0.990937	0.992513	0.993695	0.992643	0.992906	0.992539	0.000900	10
gini	8	sqrt	500	0.991331	0.993038	0.993826	0.992118	0.992906	0.992644	0.000851	9
gini	8	log2	200	0.991593	0.992907	0.993301	0.991724	0.992380	0.992381	0.000660	12
gini	8	log2	500	0.990937	0.992907	0.993826	0.992643	0.993037	0.992670	0.000952	8
entropy	4	auto	200	0.975043	0.977801	0.979509	0.973463	0.981214	0.977406	0.002836	57
entropy	4	auto	500	0.974911	0.977013	0.979771	0.973594	0.980557	0.977169	0.002689	58
entropy	4	sqrt	200	0.975305	0.977013	0.979115	0.973594	0.979637	0.976933	0.002275	60
entropy	4	sqrt	500	0.975437	0.977407	0.979246	0.973463	0.979900	0.977091	0.002389	59
entropy	4	log2	200	0.974911	0.978721	0.979640	0.973200	0.981608	0.977616	0.003101	54
entropy	4	log2	500	0.976094	0.979115	0.978983	0.973332	0.981214	0.977747	0.002744	51
entropy	5	auto	200	0.976356	0.979640	0.980034	0.975171	0.982396	0.978719	0.002618	44

entropy	5	auto	500	0.976619	0.979509	0.979903	0.974645	0.982396	0.978614	0.002702	45
entropy	5	sqrt	200	0.976356	0.979509	0.979509	0.974645	0.982528	0.978509	0.002746	48
entropy	5	sqrt	500	0.976488	0.979377	0.979771	0.974645	0.982396	0.978536	0.002701	47
entropy	5	log2	200	0.975962	0.979903	0.980428	0.974777	0.982002	0.978614	0.002764	46
entropy	5	log2	500	0.976619	0.979771	0.980297	0.974908	0.982528	0.978825	0.002720	43
entropy	6	auto	200	0.984632	0.985682	0.987784	0.980951	0.987126	0.985235	0.002408	32
entropy	6	auto	500	0.983581	0.985814	0.988178	0.980294	0.987651	0.985104	0.002895	33
entropy	6	sqrt	200	0.983318	0.986996	0.985682	0.981083	0.987520	0.984920	0.002406	34
entropy	6	sqrt	500	0.983449	0.985288	0.985682	0.980032	0.987782	0.984447	0.002602	36
entropy	6	log2	200	0.984106	0.985551	0.987390	0.982002	0.987126	0.985235	0.002002	31
entropy	6	log2	500	0.982136	0.985814	0.987521	0.980557	0.987126	0.984631	0.002786	35
entropy	7	auto	200	0.988704	0.989492	0.991199	0.988045	0.990410	0.989570	0.001135	23
entropy	7	auto	500	0.989360	0.990280	0.991068	0.987914	0.990279	0.989780	0.001078	20
entropy	7	sqrt	200	0.988441	0.990017	0.991593	0.988702	0.990541	0.989859	0.001170	18
entropy	7	sqrt	500	0.988572	0.989886	0.991068	0.988702	0.990804	0.989806	0.001033	19
entropy	7	log2	200	0.988966	0.989754	0.990017	0.988177	0.990147	0.989412	0.000741	24
entropy	7	log2	500	0.988966	0.989886	0.990542	0.988308	0.990673	0.989675	0.000913	21
entropy	8	auto	200	0.992644	0.993170	0.993170	0.991461	0.993431	0.992775	0.000705	6
entropy	8	auto	500	0.993038	0.992776	0.994352	0.991986	0.993169	0.993064	0.000763	2
entropy	8	sqrt	200	0.992644	0.993301	0.993170	0.992643	0.992643	0.992880	0.000293	4
entropy	8	sqrt	500	0.992776	0.992907	0.994220	0.991986	0.993300	0.993038	0.000729	3
entropy	8	log2	200	0.992907	0.992513	0.994352	0.991724	0.992906	0.992880	0.000853	5
entropy	8	log2	500	0.992907	0.992776	0.994352	0.992249	0.993037	0.993064	0.000697	1

Note that due to errors in the averaging technique no volcanic aerosols were detected in the database for OMI. Augustine Volcano, in the Cook Inlet of the Gulf of Alaska, erupted on January 13 and 14, 2006. According to the Alaska Volcano Observatory (AVO), these explosive eruptions produced clouds of volcanic ash and flows of mud and rock fragments. However, our study reported no such aerosol type in the training data. Thus we effectively classify only three aerosol types: Weakly absorbing, Biomass burning and DesertDust.
Figure 6.43 shows the confusion matrix for the three classes. From the confusion matrix we see that the three classes Weakly absorbing, Biomass burning and Desert Dust. For class 1: Weakly absorbing 13525 datapoints were correctly classified as weakly absorbing. 11 datapoints were incorrectly classified as Biomass burning and 6 datapoints were incorrectly classified as DesertDust. For class 2: Biomass burning, 1514 datapoints were correctly classified as biomass burning, 14 were misclassified as Weakly absorbing and none were misclassified as Desert Dust. For class 3: Desert Dust, 1237 were correctly classified as Desert Dust, 7 datapoints were misclassified as Weakly absorbing, and no datapoints were misclassified as Biomass burning.

Figure 6.43: Confusion Matrix for Random Forest



From Figure 6.44 we can see the precision, recall and support for the three aerosol classes. Class 1 : Weakly absorbing aerosol had a precision of 100%, recall of 99%, and f1-score of 100%. Class 2: Biomass burning had a precision of 96%, recall of 99% and f1-score of 98%. Class 3: DesertDust aerosol had a precision of 98%, recall of 99% and f1-score of 98%.

Figure 6.44: Precision, Recall and Support

	precision	recall	f1-score	support
1	1.00	0.99	1.00	13542
2	0.96	0.99	0.98	1528
3	0.98	0.99	0.98	1244
accuracy			0.99	16314
macro avg	0.98	0.99	0.99	16314
weighted avg	0.99	0.99	0.99	16314

6.2.2. Discussion on OMI

Our requirement when we started this study was to meet an accuracy requirement of 90% on aerosol classification using supervised learning. As already stated in Section 6.1.2 we set the benchmark to be 90% accounting the inherent prediction errors present in classification algorithms. The reader is referred to Section 6.1.2 for a detailed discussion. Using RF on OMI dataset with hyperparameter tuning we reached an overall macro accuracy of 98%, precision of 99%, recall of 99% and f1-score of 99% on OMI dataset for the three classes of aerosols. In a nutshell, we found out that the algorithm RF outperforms the algorithms SVM and KNN for the OMI data set. However, we interpret these results with caution since we have to account for optimization bias that might have occured when we chose default hyperparameter setting when evaluating the performance of these three algorithms.

A research by Wonie Choi et al [10] performed aerosol classification using space-borne measurement using AERONET data using RF technique. Initially, seven classes were classified that included: pureDust,Dust-dominant mixed, pollution-dominant mixed aerosols, and pollution aerosols containing four categories: strongly, moderately, weakly and non-absorbing. The model was evaluated on remaining AERONET data and resulted in an accuracy of only 59%. When only four classes pureDust,Dust-dominant mixed, strongly absorbing and non-absorbing were classified the accuracy shot to 72%. In our study we use RF as a classifier on OMI data with Aerosol Optical thickness, Single Scattering albedo and UV Aerosol Index as input and three class labels: Weakly absorbing, Biomass burning and Desert Dust. We get a classification accuracy of 99% across three classes when measured on our test set. Although this accuracy seems to be high, the limitation is the low number of classes

resulting in lower information regarding the number of aerosols. One recommendation is to increase the number of classes in the training set so as to maintain a high amount of information on aerosol distribution.
The reader is referred to Chapter 7 for an overall discussion of the methodology and the obtained results on OMI and POLDER-3.

6.3. Discussion of the Results on POLDER-3 and OMI data

We began the study by collecting POLDER-3 satellite data from SRON and OMI satellite data from the NASA website. POLDER-3 instrument on PARASOL satellite was launched just five months after OMI instrument on Aura satellite in the same year 2004. These satellite data were chosen due to the information they contained regarding aerosols and the fact that both Parasol(satellite containing POLDER-3 instrument) and Aura(containing OMI instrument) satellites were on the same A-train. A-train nickname of Afternoon constellation refers to a group of satellites following the same orbital track in a sun-synchronous polar orbit crossing the equator at about 1:30 PM local time one after the other in close succession. Being in the A-train resulted in near-concurrent observations from the instruments. These aid in further developing a comprehensive understanding of the earth science data. Inspite of being on the same A-train there are certain dissimilarities between the two instruments. POLDER-3 is a passive optical imaging radiometer polarimeter instrument whereas OMI is a visible ultraviolet spectrometer. POLDER-3 measures in the wavelength 443 and 910 nm FWHM while OMI measures from 270 to 500 nm. The reader is referred to Table 3.2 for further information regarding the two satellites. Even though POLDER-3 has many more detailed aerosol micro-physical features as compared to OMI, it is not as desirable to study anymore since the mission was discontinued on 18 December 2013, exactly nine years after launch. The reader is referred to Chapter 3 Section 3.2 for further information regarding the POLDER-3 instrument and PARASOL mission.

POLDER-3 data is not freely available on the internet databases like OMI `https://disc.gsfc.nasa.gov/datasets/OMAERUVe_003/summary?keywords=OMAERO_003`. The data for POLDER-3 had to be specifically obtained from Otto Hasekamp from SRON. The data received from SRON was in the format of 1 degree x 1 degree gridded monthly averaged files in .nc format for the year 2006. Since, POLDER-3 was obtained for the year 2006, OMI data was also downloaded from the NASA database for the year 2006 hoping to keep synergy between the observations. The OMI data was in the he5 format. Note that different tools are required to extract these different file formats. .nc is a NetCDF file format whereas .he5 is a hdf5 file. HDF5 is extremely feature-rich, and has some great performance features and easier to use with python. On the other hand NetCDF has a simpler API, and a much wider tool base. R was used to extract the data fron the .nc file due to easier library extraction in R while python was used to extract OMI data due to easier available libraries like h5py for extraction.

On further examination, it was found that dealing with OMI data was not as straightforward as working on POLDER-3 data. POLDER-3 data was rather straightforward with 12 files for each month of the year 2006. In each file 55 variables on aerosol (Refer to Appendix B.1 for all the variables) were available to use to study aerosols. On the other hand the OMI data has two variables called OMI O and OMI UV data products. On studying the user guide it was clear that these two data products had different variables. The user is referred to Table 5.7 to study these different variables. OMAERUV uses near-UV algorithm while OMAERO uses a multi-wavelength algorithm that uses upto 20 wavelength bands between 331 nm and 500 nm. On speaking with the key investigator of the product Dr. Pepijn Veefkind (as seen in Appendix D), it was evident that OMAERO product containing Aerosol Optical Depth, Single Scattering Albedo, and other geolocation was preferred over OMAERUV. Moreover it also came with the Aerosol Model Classification with the four major classes being Weakly Absorbing, Biomass Burning, Desert Dust and Volcanic Aerosols which was not present in the OMAERUV dataset. Some files were missing for the months February and March resulting in 362 daily files instead of 365 for the year 2006. Note that this was in contrast to the 12 files for POLDER-3. The OMAEROe files contain daily data from approximately 15 orbits. The maximum files size for the OMAEROe data product is about 7 Mbytes. This L3 product selects the best aerosol value from the L2 data. The spatial resolution is 0.25 degree x 0.25 degree. **Unlike the eight clusters in the POLDER-3 data, in the OMI data we could only classify three aerosol classes**. This is a consequence of the difference in cluster labels collection between POLDER-3 and OMI. POLDER-3 clusters were generated using unsupervised clustering and then manually labelling the clusters. OMI on the otherhand has four preclassified clusters as already mentioned above. With one class Volcanic Aerosol being less prevalent in the dataset due to averaging error it is not present in the training set and thus effectively only three classes are present in OMI. Volcanic eruption occurred on a few days in the month but when we averaged the aerosol on a geolocation for a month we only obtained the most prevalent aerosol and not these special cases that happen on only a few days in a month. In the future, a better averaging technique can be employed to generate the monthly averaged files such that no information is lost when averaging all the days in a month.

Now coming to the discussion on supervised learning algorithm, in our study we implemented SVM, KNN and RF in python from the scikit-learn library. These are standalone models that we used in this study. In the future, we could use a combination of models using voting classification to further increase the accuracy of the models. Another point for discussion is the hyper-parameter tuning that we used in this thesis. In our study we only used a selected group of combination of hyperparameters in our GridSearchCV algorithm, in the future we recommend running a more comphrehensive and exhaustive search over a larger hyperparameter range. Finally, coming to the discussion of attachment of the source of data in our case use used offline

data either downloaded from the online database like OMI or obtained from scientists like SRON for POLDER-3. In the future we could directly make a live connection to the data source such that we get real-time updates of the different aerosol types. One useful application could be detecting harmful aerosols as was done with air monitoring service.

To sum up, as seen in Table 6.8 our requirement was to meet an accuracy of 90% on aerosol classification using supervised learning. Using SVM on POLDER-3 dataset with hyperparameter tuning we reached an overall accuracy of 99%, precision of 99%, recall of 98% and f1-score of 99% on POLDER-3 dataset for the eight classes of aerosols. Using RF on OMI dataset with hyperparameter tuning we reached an overall accuracy of 99%, precision of 99%, recall of 99% and f1-score of 99% on OMI dataset for the three classes of aerosols. We see in Figure 6.19 that root mean square error for RF is 0.08 and that of KNN is 0.16 and that for SVM is 0.17. This difference is very small compared to the difference in POLDER-3 where the root mean square for the three algorithms were 0.2 for SVM, 0.35 for RF and 0.8 for KNN. From this we can infer two points. One, all the three algorithms perform better with overall lower error on the OMI dataset than on POLDER-3 dataset. Second, the difference of performance error between the two best performing algorithms is very close: RF and KNN being only 0.08 for OMI and 0.04 between SVM and RF for POLDER-3. This means that optimization bias plays a significant effect on both the datasets. Due to time constraints the best hyperparameter settings for all the three algorithms were not explored and only the best performing algorithm was tuned with the different combination of hyperparameter for both POLDER-3 and OMI. Furthermore, it was observed that tuning the hyper parameters further increases the accuracy of the algorithm.

In Table 6.8 we can see that we have met all the requirements we set out to meet in this thesis except for the requirement of being able to distinguish between five types of aerosol for OMI. Our algorithm on OMI can classify only three types of aerosol.

Table 6.8: Requirements

ReQ-ID	Requirement	SubCategory	Verification
R-001	The application shall implement supervised learning algorithm for the purpose of aerosol classification on a satellite data	Application	The application implemented supervised learning algorithm KNN, SVM and RF on POLDER-3 data and classified eight aerosol types
R-002	The application shall be be implemented for two satellite missions	Application	The application was implemented on two satellite data POLDER-3 and OMI.
R-003	The application shall to a reasonable degree to accuracy be able to distinguish between at least five major types of aerosols.	Application	The application on POLDER-3 can classify between eight type of aerosols. However, The application can on OMI classify between only three type of aerosols.
R-004	The application shall be capable to be deployed and tested on a computer with the specifications of 1.4 Ghz processor and 8 Gb RAM	Performance	The application was run and tested on a computer with the specifications of 1.4 Ghz processor and 8 Gb RAM
R-005	The application shall reach a performance metric of accuracy of at least 90% on the test dataset	Performance	The application reached a performance metric of 99% on the test dataset

7

Conclusion and Recommendations

In the thesis, we implemented the three supervised learning algorithms SVM, KNN, and RF on satellite data to classify aerosol types. To reach this goal we followed a series of steps. First, we researched into possible satellite data missions whose data contained information regarding atmospheric aerosols. We selected two missions and satellite instruments POLDER-3 and OMI whose data was obtained from Otto Hasekamp and NASA repository respectively. The data obtained was in different formats and the next step was to clean and preprocess the data. The NaNs in the data were removed. Following this, all the hyperparameters in the data relating to aerosols were studied and features were selected to be trained in the machine learning models. Recall that supervised learning techniques require a class label. To obtain the class label for POLDER-3, the research of Vincent de Bakker [16] was studied, understood and recreated. Unsupervised learning techniques SOM and K means were run in R studio to get these labels and the data was stored in .RDA format. This .RDA format was ported to python to implement the supervised learning techniques. Scikit-learn is a library in python which provides a library to implement these algorithms. The library and the function calls were studied and the scoring metric was chosen as negative mean squared error from which root mean squared error of the two algorithms were compared. SVM proved to be the best performing algorithm for POLDER-3 data. Following this step, optimization of the algorithm was performed and optimal combination for C and Gamma were chosen while keeping an eye out for balancing bias and variance. The algorithm improved from 93% accuracy to 99% accuracy with hyperparameter tuning. To implement the supervised learning algorithm on OMI, again, we needed the class labels as was needed for POLDER-3. However, in this case the data already came with preclassified clusters. From fifty smaller classes there were four major class labels of which three were the most dominant. With these class labels, we again trained and evaluated the performance of the model on OMI data. In this case however, RF was the best performing model. The thesis started with a few research questions listed below. This chapter presents the conclusions to the research questions that were sought out in the thesis.

1. **What are the past satellite missions that give information regarding the atmospheric aerosols?**

   As seen in Chapter 2, satellite missions gathering data regarding atmospheric aerosols began in 1975 with the Apollo-Soyuz satellite with the SAM experiment. Since then there have been various missions gathering data regarding the atmospheric aerosols. The lists of these satellite missions was gathered in Table B.1. Of these various satellites and missions, two were chosen in this thesis. These being, POLDER-3 and OMI. For further discussion on POLDER-3 and OMI, the reader is referred to Table 3.2.
2. **Which algorithms have higher accuracy on the satellite aerosol retrievals?**
   We can see in Appendix A that several studies state using artificial intelligence to give a boost of classification on satellite data. Again, the answer to this question is not straightforward since the accuracy depends on the algorithm of choice and the dataset at hand. Based on the literature review in Chapter 4, KNN, RF and SVM show the highest promise on satellite aerosol retrievals.
3. **What is the classification error of the machine learning algorithms SVM, KNN and RF for aerosol classification?**
   Using the default setting of hyperparameters in the Scikit-learn library in python we saw that the three algorithms SVM, KNN and RF performed differently on the two datasets POLDER-3 and OMI. In Figure 6.15 we saw using the root mean squared error metric that error for RF is 0.08 and that of KNN is 0.16 and that for SVM is 0.17. This difference is very small compared to the difference in POLDER-3 where the root mean square for the three algorithms were 0.2 for SVM, 0.35 for RF and 0.8 for KNN. Using SVM on POLDER-3 dataset with hyperparameter tuning we reached an overall accuracy of 99%, precision of 99%, recall of 98% and f1-score of 99% on POLDER-3 dataset for the eight classes of aerosols. Using RF on OMI dataset with hyperparameter tuning we reached an overall accuracy of 99%, precision of 99%, recall of 99% and f1-score of 99% on OMI dataset for the three classes of aerosols.

   (a) **Is it possible to build a higher accuracy than that presently achieved by the existing algorithms for aerosol classification?**

Yes, it is possible to build a higher accuracy model. A greater range of Gridsearch CV hyperparameters could result in an even greater accuracy. Further, lowering the number of aerosol classes might lead to higher accuracy but it would result in lower information being contained. For example, if we were to remove the aerosol class with more misclassifications in it, it would reduce the overall error and increase the accuracy. Note here that our algorithms achieve 99% accuracy on both OMI and POLDER-3. However, the model on POLDER-3 generated eight aerosol classes while OMI generated only three aerosol classes.

(b) **How do the classification results of SVM, KNN and RF compare with each other?**

We can see in Figure 6.38 that for the OMI dataset There is no one size fits all solution to compare the performance of the three algorithms SVM, KNN and RF. Each dataset i.e. POLDER, OMI yield a different result. In Figure 6.15 we see that SVM was the best performing algorithm for POLDER-3 dataset. While, in 6.38 we can see that RF was best performing algorithm for OMI dataset.

Through the implementation of the thesis, the following points were the key learning which were discussed in detail in Section 6.3.

1. Machine learning shows a promise to be a viable tool for the purpose of aerosol classification and inturn in earth observation.
2. Hyperparameter tuning can greatly improve the accuracy of the model
3. Through this study we demonstrate that Random forest is a viable model capable of aerosol classification on OMI data with Aerosol Optical Thickness, Single Scattering Albedo and Aerosol Index as input.
4. Through this study we demonstrate that Support Vector Machine is a viable model capable of aerosol classification on POLDER-3 data with Angstrom Exponent, Single Scattering Albedo, Sphericity and Real refractive Index as input.

7.1. Recommendations and Future Work

As was already mentioned in Chapter 7 there are a few areas recommended for future work for the thesis project. In this section we reiterate those points as well as add a few new points as recommendation for future work.

As a first recommendation, we begin with the choice of satellite mission for future study for aerosol classification. In this study we studied POLDER-3 and OMI. In the future we recommend using data from two instruments. The first recommendation is the TROPOMI instrument on the Copernicus Sentinel-5 Precursor mission as it bridges the gap between SCIAMACHY/Envisat (which terminated in April 2012), the OMI/AURA mission, and the upcoming Copernicus Sentinel-4 and Sentinel-5 missions in terms of global atmospheric data products. When compared with its predecessors, TROPOMI has a much higher resolution of 7 km x 3 km at best, while OMI has only 24 km x 13 km. This is a huge improvement of the predecessors GOME2 at 80 km x 40 km and SCIAMACHY at 200 km x 30 km. TROPOMI has a wider wavelength range than the OMI instrument, having bands in the *Near Infrared* (NIR) and *Short-Wave Infrared* (SWIR). The major goal of using the NIR band is to improve cloud correction for trace gas retrievals. When compared to the lower oxygen A band (758–770 nm), the deeper oxygen A band (758–770 nm) includes much more information about clouds, including cloud pressure and cloud percentage. Prior to the launch of TROPOMI, daily global measurements of aerosol height were not operational. Active sensors, most notably ground-based lidar systems or the space-borne *Cloud-Aerosol Lidar with Orthogonal Polarisation* (CALIOP), provided aerosol profiles, while multi-angle sensors, most notably the *Multi-Angle Imaging SpectrRadiometer* (MISR), provided aerosol layer height . While active sensors have a high vertical resolution, CALIOP and MISR are only capable of observing narrow tracks. Passive sensors, on the other hand, such as TROPOMI, can cover the entire globe in a single day. The TROPOMI data was available to all the users beginning July 2018. The *Sentinal 5 Precursor* (S5P) mission will support the data and the data products till 2023. The data can be accessed via the Copernicus Open Access Hub.More work could be done in the future for training the TROPOMI base parameters for aerosol classification. With additional products like the Aerosol layer Height being added to the arsenal of products released to the public this could result in better training performances. Secondly we recommend 3MI, a POLDER-3 follow-on instrument to be launched in the 2020s that has extended spectral range in the shortwave infrared with enhanced capabilities for cloud retrieval [1]. With better instruments and dedicated missions which provide data with more precision and higher cloud cover retrievals, the performance of the algorithm can be expected to be better.

Our next recommendation is the choice of aerosol microphysical parameters used as an input to the supervised learning algorithm. It was clear from our study that POLDER-3 and OMI had different aerosol microphysical parameters. The choice of which microphysical parameters we chose to train the algorithm might impact the training of the algorithm. We recommend that one could use additional features from POLDER-3 dataset as used in the study by Russel et al [54] including Dust single scattering albedo, imaginative refractive index, absorption angstrom exponent and volume of fine mode divided by the total volume to form input set for supervised training. It is unclear how adding features would impact that overall accuracy since adding more features is not directly proportional to increased accuracy. To further prove this point we mention the study by Russel which could generate only seven classes

[1] https://www.sron.nl/missions-earth/aerosol-missions

1. PureDust
2. PollutedDust
3. Biomass Burning, Dark Smoke
4. Biomass Burning, White Smoke
5. Urban Industrial, developed economy
6. Urban Industrial, developing economy
7. Pure Marine

using unsupervised learning using eight microphysical parameters. de Bakker generated eight clusters using only four microphysical parameters. Further analysis on how these microphysical parameters affects the accuracy is out of scope of our study and thus is recommended for future work.

As discussed in Section 6.3 in our study we implemented standalone models SVM, KNN and RF in python from the scikit-learn library. In the future, we could use a combination of models using voting classification to further increase the accuracy of the models. Another point for discussion is the hyperparameter tuning that we used in this thesis. In our study we only used a selected group of combination of hyperparameters in our GridSearchCV algorithm, in the future we recommend running a more comphrehensive and exhaustive search over a larger hyperparameter range. Another point for recommendation is the way OMI data was averaged. Since, we received the files in daily format we wanted to create a monthly average. In doing the monthly average on a geolocation we lost some information so example on volcanic aerosols that we released when the volcani erupted on only a few days. In the future, a better averaging technique can be employed to generate the monthly averaged files such that no information is lost when averaging all the days in a month. Finally, coming to the discussion of attachment of the source of data in our case use used offline data either downloaded from the online database like OMI or obtained from scientists like SRON for POLDER-3. In the future we could directly make a live connection to the data source such that we get real-time updates of the different aerosol types. One useful application could be detecting harmful aerosols in an air monitoring service application.To end, this is only the beginning of the era of machine learning in the field of Earth Observation, and there is much work to be done to fine tune the algorithm and input precise and larger datasets aimed at a targeted application at hand.

Bibliography

[1] URL: https://lnct.ac.in/wp-content/uploads/2020/04/CS-601-Machine-learning-Unit-5.pdf.

[2] *Aerosols: Tiny Particles, Big Impact*. URL: https://earthobservatory.nasa.gov/features/Aerosols/page2.php.

[3] *ARSET - High Resolution NO2 Monitoring From Space with TROPOMI*. URL: https://appliedsciences.nasa.gov/join-mission/training/english/arset-high-resolution-no2-monitoring-space-tropomi.

[4] UN General Assembly. "Global indicator framework for the Sustainable Development Goals and targets of the 2030 Agenda for Sustainable Development". In: *United Nations Statistics Division: New York, NY, USA* (2017).

[5] YS Bennouna. "Satellite remote sensing of aerosols using geostationary observations from MSG-SEVIRI". In: (2009).

[6] YS Bennouna et al. "Aerosol remote sensing over the ocean using MSG-SEVIRI visible images". In: *Journal of Geophysical Research: Atmospheres* 114.D23 (2009).

[7] KF Boersma et al. "Near-real time retrieval of tropospheric NO 2 from OMI". In: *Atmospheric Chemistry and Physics* 7.8 (2007), pp. 2103–2118.

[8] Gustau Camps-Valls, Jose Bioucas-Dias, and Melba Crawford. "A special issue on advances in machine learning for remote sensing and geosciences [from the guest editors]". In: *IEEE Geoscience and Remote Sensing Magazine* 4.2 (2016), pp. 5–7.

[9] T. Chai and R. R. Draxler. "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature". In: *Geoscientific Model Development* 7.3 (2014), pp. 1247–1250. ISSN: 1991-9603. DOI: 10.5194/gmd-7-1247-2014. URL: https://dx.doi.org/10.5194/gmd-7-1247-2014.

[10] Wonei Choi, Hanlim Lee, and Jeonghyeon Park. "A First Approach to Aerosol Classification Using Space-Borne Measurement Data: Machine Learning-Based Algorithm and Evaluation". In: *Remote Sensing* 13.4 (2021), p. 609.

[11] Costa Christopoulos et al. "A machine learning approach to aerosol classification for single-particle mass spectrometry". In: (2018).

[12] DA Chu et al. "Validation of MODIS aerosol optical depth retrieval over land". In: *Geophysical research letters* 29.12 (2002), MOD2–1.

[13] Roger Singh Chugh et al. "A Comparative Analysis of Classifiers for Image Classification". In: *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE. 2020, pp. 248–253.

[14] *CIMON-2 is on its way to the ISS - DLR Portal*. https://www.dlr.de/content/en/articles/news/2019/04/20191205_cimon2-on-its-way-to-the-iss.html. (Accessed on 09/23/2020).

[15] David Crowe et al. "Two Supervised Machine Learning Approaches for Wind Velocity Estimation Using Multi-Rotor Copter Attitude Measurements". In: *Sensors* 20.19 (2020), p. 5638.

[16] Vincent De bakker. "A Novel Cross-Validation Framework for Identification of Atmospheric Aerosol Types". PhD thesis. 2018. URL: https://www.universiteitleiden.nl/binaries/content/assets/science/mi/scripties/statscience/2017-2018/2018_08_27_masterthesis_debakker.pdf.

[17] Gil Denis et al. "Towards disruptions in Earth observation? New Earth Observation systems and markets evolution: Possible scenarios and impacts". In: *Acta Astronautica* 137 (2017), pp. 415–433.

[18] Pedro Domingos. "A few useful things to know about machine learning". In: *Communications of the ACM* 55.10 (2012), pp. 78–87.

[19] M Esposito et al. “In-orbit demonstration of artificial intelligence applied to hyperspectral and thermal sensing from space”. In: *CubeSats and SmallSats for Remote Sensing III*. Vol. 11131. International Society for Optics and Photonics. 2019, p. 111310C.

[20] Paola Formenti et al. “Aerosol optical properties derived from POLDER-3/PARASOL (2005–2013) over the western Mediterranean Sea–Part 1: Quality assessment with AERONET and in situ airborne observations”. In: *Atmospheric Measurement Techniques* 11.12 (2018), pp. 6761–6784.

[21] GISGeography et al. *Latitude, Longitude and Coordinate System Grids*. June 2021. URL: `https://gisgeography.com/latitude-longitude-coordinates/`.

[22] Noel Gorelick et al. “Google Earth Engine: Planetary-scale geospatial analysis for everyone”. In: *Remote sensing of Environment* 202 (2017), pp. 18–27. ISSN: 0034-4257. DOI: `10.1016/j.rse.2017.06.031`. URL: `https://dx.doi.org/10.1016/j.rse.2017.06.031`.

[23] Sepand Haghighi et al. “PyCM: Multiclass confusion matrix library in Python”. In: *Journal of Open Source Software* 3.25 (2018), p. 729.

[24] Matthew C Hansen et al. “High-resolution global maps of 21st-century forest cover change”. In: *science* 342.6160 (2013), pp. 850–853.

[25] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learnin*. 2009.

[26] R.C. Hibbeler. *Enhancing aerosol remote sensing data for the air quality market*. Leiden, Netherlands: Airbus Defense and Space, 2018.

[27] Akiko Higurashi and Teruyuki Nakajima. “Detection of aerosol types over the East China Sea near Japan from four-channel satellite data”. In: *Geophys. Res. Lett.* 29 (Sept. 2002). DOI: `10.1029/2002GL015357`.

[28] Brent N Holben et al. “AERONET—A federated instrument network and data archive for aerosol characterization”. In: *Remote sensing of environment* 66.1 (1998), pp. 1–16.

[29] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. *A practical guide to support vector classification*. 2003.

[30] Myeong-Jae Jeong and Zhanqing Li. “Quality, compatibility, and synergy analyses of global aerosol products derived from the advanced very high resolution radiometer and Total Ozone Mapping Spectrometer”. In: *Journal of Geophysical Research: Atmospheres* 110.D10 (2005).

[31] *K-nearest neighbors algorithm*. Oct. 2021. URL: `https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`.

[32] Teuvo Kohonen and Timo Honkela. “Kohonen network”. In: *Scholarpedia* 2.1 (2007), p. 1568.

[33] Pavan Kumar Kolluru. “SVM based dimensionality reduction and classification of hyperspectral data”. MA thesis. University of Twente, 2013.

[34] Charles Kooperberg and Michael LeBlanc. “Multivariate Nonparametric Regression”. In: *High-Dimensional Data Analysis in Cancer Research*. Springer, 2009, pp. 1–24.

[35] Tom Landry et al. “Applying machine learning to earth observations in a standards based workflow”. In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 5567–5570.

[36] David J Lary et al. “Machine learning applications for earth observation”. In: *Earth observation open science and innovation* 165 (2018).

[37] Kwon H Lee et al. “Atmospheric aerosol monitoring from satellite observations: a history of three decades”. In: *Atmospheric and biological environmental monitoring* (2009), pp. 13–38. DOI: `10.1007/978-1-4020-9674-7_2`.

[38] Kwon Ho Lee et al. “Spatio-temporal variability of satellite-derived aerosol optical thickness over Northeast Asia in 2004”. In: *Atmospheric Environment* 41.19 (2007), pp. 3959–3973.

[39] Yingying Ma and Wei Gong. “Evaluating the Performance of SVM in Dust Aerosol Discrimination and Testing its Ability in an Extended Area”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5.6 (2012), pp. 1849–1858. DOI: `10.1109/JSTARS.2012.2206572`.

[40] Yingying Ma and Wei Gong. “Evaluating the performance of SVM in dust aerosol discrimination and testing its ability in an extended area”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5.6 (2012), pp. 1849–1858.

[41] Mohammad Reza Mahdiani et al. “Modeling viscosity of crude oil using k-nearest neighbor algorithm”. In: *ADVANCES IN GEO-ENERGY RESEARCH* 4 (Nov. 2020). DOI: 10.46690/ager.2020.04.08.

[42] Kebiao Mao et al. “A practical split-window algorithm for retrieving land-surface temperature from MODIS data”. In: *International Journal of Remote Sensing* 26.15 (2005), pp. 3181–3204.

[43] Vânia Martins et al. “Deposition of aerosol particles from a subway microenvironment in the human respiratory tract”. In: *Journal of Aerosol Science* 90 (2015), pp. 103–113.

[44] Dominic Mazzoni et al. “An operational MISR pixel classifier using support vector machines”. In: *Remote Sensing of Environment* 107.1-2 (2007), pp. 149–158.

[45] Samuel McDonald. *SAM, Launched 40 Years Ago, Opened an Era of Atmospheric Discoveries*. July 2015. URL: https://www.nasa.gov/langley/the-sam-experiment-launched-40-years-ago-opened-an-era-of-atmospheric-discoveries.

[46] Andrea Meraner et al. “Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 166 (2020), pp. 333–346. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2020.05.013.

[47] Lauri Myllyvirta. “Quantifying the economic costs of air pollution from fossil fuels”. In: *Centre for Research on Energy and Clean Air* (2020).

[48] NASA. *The Aura Mission*. 2000. URL: https://aura.gsfc.nasa.gov/omi.html (visited on 12/24/2020).

[49] Feras Al-Obeidat et al. “A fuzzy decision tree for processing satellite images and landsat data”. In: *Procedia Computer Science* 52 (2015), pp. 1192–1197.

[50] Nikolaos Papagiannopoulos, Lucas Alados Arboledas, Juan Luis Guerrero-Rascado, et al. “An automatic observation-based aerosol typing method for EARLINET”. In: (2018).

[51] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[52] MJM Penning de Vries et al. “A global aerosol classification algorithm incorporating multiple satellite data sets of aerosol and trace gas abundances”. In: *Atmospheric Chemistry and Physics* 15.18 (2015), pp. 10597–10618.

[53] Omid Rahmati et al. “Identifying sources of dust aerosol using a new framework based on remote sensing and modelling”. In: *Science of The Total Environment* 737 (2020), p. 139508.

[54] Philip B Russell et al. “A multiparameter aerosol classification method and its application to retrievals from spaceborne polarimetry”. In: *Journal of Geophysical Research: Atmospheres* 119.16 (2014), pp. 9838–9863.

[55] Luis Samaniego, András Bárdossy, and Karsten Schulz. “Supervised classification of remotely sensed imagery using a modified *k*-NN technique”. In: *IEEE Transactions on Geoscience and Remote Sensing* 46.7 (2008), pp. 2112–2125.

[56] A. L. Samuel. “Some Studies in Machine Learning Using the Game of Checkers”. In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. ISSN: 0018-8646. DOI: 10.1147/rd.33.0210.

[57] Aurélie C Shapiro et al. “The mangroves of the zambezi delta: increase in extent observed via satellite from 1994 to 2013”. In: *Remote Sensing* 7.12 (2015), pp. 16504–16518.

[58] Richa Sharma, Aniruddha Ghosh, and PK Joshi. “Decision tree approach for classification of remotely sensed satellite data using open source support”. In: *Journal of Earth System Science* 122.5 (2013), pp. 1237–1247.

[59] Svetlana Shekhtman. *NASA Applying AI Technologies to Problems in Space Science*. Nov. 2019. URL: https://www.nasa.gov/feature/goddard/2019/nasa-takes-a-cue-from-silicon-valley-to-hatch-artificial-intelligence-technologies.

[60] Seema Singh. *Understanding the bias-variance tradeoff*. Oct. 2018. URL: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229.

[61] *Sklearn.ensemble.randomforestclassifier*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[62] Yvette Smith. *Just Another Day on Aerosol Earth*. Aug. 2018. URL: https://www.nasa.gov/image-feature/just-another-day-on-aerosol-earth.

[63] Irina N. Sokolik. *Lecture notes on Regional radiative effects due to anthropogenic aerosols*. Fall 1998.

[64] Yunhui Tan et al. "Validation of POLDER-3/GRASP aerosol products using AERONET measurements over China". In: *Atmospheric Environment* 215 (2019), p. 116893.

[65] *The Aura Mission*. URL: https://aura.gsfc.nasa.gov/omi.html.

[66] Heidar Th Thrastarson et al. "AIRS/AMSU/HSB Version 7 Level 2 Product User Guide". In: *Jet Propulsion Laboratory, California Institute of Technology: Pasadena, CA, USA* (2020), pp. 83–92.

[67] Omar Torres et al. "Aerosols and surface UV products from Ozone Monitoring Instrument observations: An overview". In: *Journal of Geophysical Research* 112 (Dec. 2007), pp. 1–14. DOI: 10.1029/2007JD008809.

[68] F Vachon et al. "Remote sensing of aerosols over North American land surfaces from POLDER and MODIS measurements". In: *Atmospheric Environment* 38.21 (2004), pp. 3501–3515.

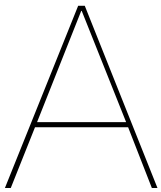
[69] *Visualizing All of Earth's Satellites*. https://www.visualcapitalist.com/visualizing-all-of-earths-satellites/. Accessed: 2020-09-30.

[70] Yixiang Wang et al. "Assessing the cytotoxicity of ambient particulate matter (PM) using Chinese hamster ovary (CHO) cells and its relationship with the PM chemical composition and oxidative potential". In: *Atmospheric Environment* 179 (2018), pp. 132–141.

[71] A Wiedensohler et al. "Mobility particle size spectrometers: harmonization of technical standards and data structure to facilitate high quality long-term observations of atmospheric particle number size distributions". In: *Atmospheric Measurement Techniques* 5.3 (2012), pp. 657–685.

[72] YC Zheng, LL Li, and YP Wang. "AN AEROSOL TYPE CLASSIFICATION METHOD BASED ON REMOTE SENSING DATA IN GUANGDONG, CHINA." In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* (2019).

[73] Zhonghua Zheng et al. *Evaluation of Machine Learning Approaches to Estimate Aerosol Mixing State Metrics in Atmospheric Models*. Tech. rep. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2019.



Overview of past studies on Machine Learning on Aerosol Classification

Table A.1: Overview of past studies on Machine Learning on Aerosol Classification

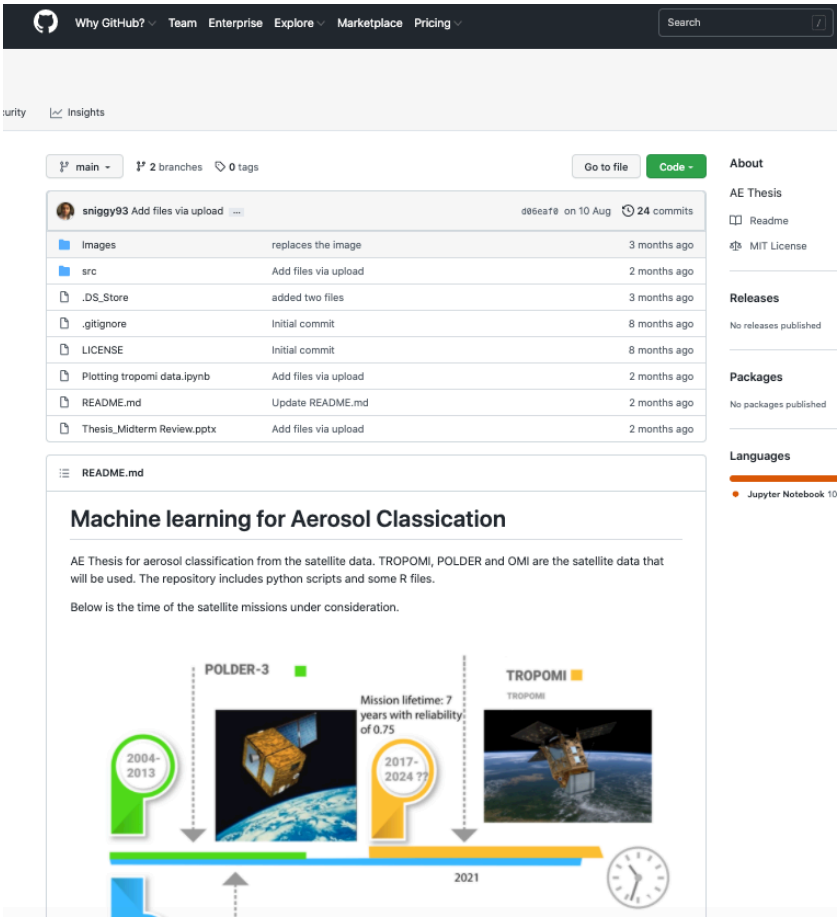
Reference	Research aim	Technique	Input data	Results	Limitation/future scope
Mazzoni et al. (2007) [44]	Detection and classification (land, cloud, aerosol, water or snow/ice) from MISR data	SVM	Multi-angle imaging spectroradiometer (MISR) image	Classification accuracy (81%) at 1.1-km pixel level	The accuracy further needs to be improved by extracting significant features from trained SVM
Ruske et al. (2017) [**ruske2017evaluation**]	Classification of biological aerosol	Decision trees and k-nearest neighbors, gradient boosting algorithm	Mixed polystyrene latex spheres (PSLS) and various laboratory-generated aerosol data	Average correctly classifying 82.8% (SVM) and 98.27 % (gradient boosting)	Misclassified as non-fluorescent since a good proportion of the particles is weakly fluorescent.
Zheng et al. (2017) [72]	Classification of aerosol types in Beijing	Fuzzy -c-mean (FCM) clustering algorithm	6732 records from AERONET sites	Capable of identifying aerosol types in regions where aerosol particles are contributed by complex components	To validate the results by adopting different features set.
Leskiewicz et al. (2018)	Investigated the real-time classification of biological aerosols.	ANN	Spectral characteristics of bioaerosols were collected using the Bardet instrument (48 instance)	Automated and speed up the analysis, which reduced the time complexity	In future, will extended to obtain potentially a vast variety of samples including atmospherically relevant bacteria and fungi.
Christopoulos et al. (2018) [11]	Aerosol classification for single-particle mass spectrometry	Random forest classifier	Single-particle mass spectrometry dataset	Classification accuracy of 87%	Further improve the classification accuracy by adopting cluster method
Zheng et al. (2019) [72]	Aerosol classification method based on remote sensing data in Guangdong, china	K-means clustering algorithm with mahala Nobis distance	Multi-decadal fundamental climate data record (FCDR) of aerosol properties from a 32-year record of satellite near-UV observations.	Cluster with aerosol type in an efficient manner	Needs to validate and compare the performance by calculating different metrics
Zhang et al. (2020)	Aerosol optical parameter was classified	SVM	Primary source (aerosol optical parameters for 370 haze-free days and 662 haze days from December 2009 to September 2014)	Accuracy (81.49%)	In future needs to consider aerosol types and their microphysical properties over the region
Crawford et al. (2020)	Investigated airborne primary biological aerosol particle	Gradient boosting ensemble decision tree algorithm's	Data was collected from laboratory experimental arrangement and ambient monitoring site	Minimizing conflation between particle types with similar fluorescent characteristics	Will improve classification capability, by providing excellent high temporal resolution datasets

B

Code repository

All the code developed as part of this thesis could be found be found at this repository: `https://github.com/sniggy93/Machinelearning_Aerosol`

Figure B.1: Github link



B.1. POLDER parameters

1. $AOT440nm$
2. $AOT490nm$
3. $AOT563nm$
4. $AOT670nm$
5. $AOT865nm$

6. $AOT1020nm$
7. $SSA440nm$
8. $SSA490nm$
9. $SSA563nm$
10. $SSA670nm$
11. $SSA865nm$
12. $SSA1020nm$
13. $refffine$
14. $veffine$
15. $mrfine$
16. $mifine$
17. $Nfine$
18. $spherefracfine$
19. $reffcoarse$
20. $veffcoarse$
21. $mrcoarse$
22. $micoarse$
23. $Ncoarse$
24. $spherefraccoarse$
25. $errorAOT440nm$
26. $errorAOT490nm$
27. $errorAOT563nm$
28. $errorAOT670nm$
29. $errorAOT865nm$
30. $errorAOT1020nm$
31. $errorSSA440nm$
32. $errorSSA490nm$
33. $errorSSA563nm$
34. $errorSSA670nm$
35. $errorSSA865nm$
36. $errorSSA1020nm$
37. $errorrefffine$
38. $errorveffine$
39. $errormrfine$
40. $errormifine$
41. $errorNfine$
42. $errorspherefracfine$
43. $errorreffcoarse$
44. $errorveffcoarse$
45. $errormrcoarse$
46. $errormicoarse$
47. $errorNcoarse$
48. $errorspherefraccoarse$
49. RRI
50. $errorRRI$
51. IRI
52. $errorIRI$
53. *Extinction Angstrom Exponent*(EAE)490670
54. EAE490865
55. EAE490670
56. EAE490865
57. $error$EAE490670
58. $error$EAE490865
59. $errorAE490670$
60. $error$EAE490865
61. $numberofpointsland$
62. $numberofpointsocean$
63. $latcorners$
64. $loncorners$
65. $latcenter$
66. $loncenter$
67. $psurf$

B.2. R coding for POLDER data

Packages are collections of R functions, data, and compiled code in a well-defined format. The directory where packages are stored is called a library. R comes with a standard set of packages.
The RData format (usually with extension .rdata or .rda) is a format designed for use with R, a system for statistical computation and related graphics, for storing a complete R workspace or selected "objects" from a workspace in a form that can be loaded back by R.

B.3. Motivation for the project

`https://www.youtube.com/watch`?v=`zIH0NimCHZ8` explains wny air pollution if worth investing in. IT further explains the history and the future of satellite missions for aerosol and air pollution monitoring.
Video from NASA explains what aerosols are and why they are important to study. Its essentially a beginners guide.
Today's space business is undergoing fast transformation, owing in major part to six transformative factors: There is an increase in the amount of high-resolution optical and radar imagery; Increased competition in the marketplace as a result of the flood of images may result in price reductions for EO data; Increased rivalry in the space sector is driving advancements in satellite hardware and software. As a result, the cost of accessing and operating in space is decreasing; Increases in temporal data resolution are altering our perspective on our globe and creating new markets; Cloud computing is cutting the cost of data storage and computation; and machine learning and artificial intelligence techniques for processing EO data are enhancing outcomes and significantly reducing the time required to examine imagery. [Source: `https://medium.com/radiant-earth-insights/how-earth-observations-cloud-computing-and-machine-learning-enables-global-development-solutions-9ad1c2e60762`]
[1] From the NASA archives we can find from literature certain regions on the globe where we expect to find a certain type of aerosols. One such example is the dust aerosol, which is expected over the desert regions like the Sahara desert. The table below shows a compilation of such data over certain regions over the globe.

Table B.1: Expected Aerosols from literature over certain regions on the globe [2]

Reference Area	Aerosol Type
North of Antarctica	Airborne salt band
Oceans	salt , sulfates from microalgae
Deserts	dust plumes
Eastern USA, Urban Europe	anthropogenic aerosols: sulphates , organic carcon
Eastern China	Heavy blankets of aerosols

[1] https://earthobservatory.nasa.gov/features/Aerosols/page2.php

C

Meeting with Atmospheric Sciences Experts

C.1. Meeting with Dr. Pepijn Veefkind

Dr. Pepijn Veefkind, having been a principal investigator for TROPOMI and a deputy investigator for OMI proved to the most reliable source to validate the assumptions made in the thesis. The following questions were asked during the meeting.

- How can the data from TROPOMI and OMI be interrelated or compared interms of atmospheric sciences?
- Can visual images for the aerosols from the OMI and TROPOMI be analysed for further processing?
- How accurate is the calibration of the OMI data aerosol classification algorithm?
- What is the best method to combine the files to create a file of L3 data?
- why we would choose imagery vs sensor data for analysis from the satellite data
- How can we easily generate L3 files from L2 data.

  Answer: The tools I was referring to is harp: https://stcorp.github.io/harp/doc/html/index.html . You want to use the regard operation (see https://stcorp.github.io/harp/doc/html/operations.html(.

C.2. Meeting with Herman Russchenberg

Herman Russchenberg is an atmospheric sciences researcher at the CiTG Geosciences department at TU Delft. His research interests are understanding aerosols, clouds and their interaction in the global cycle.