Technische Universität Berlin

Faculty IV
Electrical Engineering and Computer Science
Computer Vision and Remote Sensing

Master's Thesis

Machine Learning Methods for Ozone Total Column Retrieval from Sentinel-5 Precursor Data: Application to Synthetic and Real Measurements

---

Submitted by:
Himani Jain

November 28, 2019

Internal Supervisor:
Prof. Dr.-Ing. Olaf Hellwich
Technical University of Berlin (TUB)

DLR

External Supervisor:
Dr. Habil. Dmitry S. Efremenko
German Aerospace Center (DLR)

Statutory Declaration

I hereby certify that this thesis has been composed by me and is based on my own work, unless stated otherwise. No other person's work has been used without due acknowledgment in this thesis. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged.

Berlin, November 28, 2019 Himani Jain

Acknowledgment

First and foremost, I would like to thank my parents and my sister for being extremely supportive and understanding throughout my master's course and my partner Rishi for his immense support, encouragement, valuable feedback on my work and keeping me on my toes to give my 100%. I could have never achieved this without them. I dedicate this thesis to them.

A special thank you to Prof. Dr.-Ing. Olaf Hellwich for always being available through emails and his close guidance, even from a distance.

I am highly grateful to Prof. Dr. Thomas Trautmann for the opportunity to work at the German Aerospace Center (DLR). I would particularly like to thank my supervisor Dr. Habil. Dmitry S. Efremenko at DLR for proposing such an exciting topic and extending his guidance, expertise, time and support during the thesis work. I will forever appreciate how much I learned and grew as an individual while working on this thesis. An extended thank you to Dr. Jian Xu for entertaining my queries and for his insightful comments and feedback during the thesis work.

I would also like to thank my office colleagues Ana and Sruthy for keeping such a cheerful and warm environment at work and for keeping a check if I need some help with my thesis. Bernd and Nathalie for providing all the resources needed for smooth work in the office and from home.

A special thank you to Rishi's family for their motivational talks on the video call.

Lastly, but by no means least, I would like to thank Luca for proofreading the German version of the abstract and my lunch buddies Joana, Marvin, Ilija, Mafalda, Mario, and Sneha for refreshing breaks to keep me going with my work. A special thanks to my Berlin friends Uroosa and Vijay for keeping me motivated even from a distance and all other friends and family members who have supported directly or indirectly.

Abstract

The new generation of atmospheric composition sensors onboard satellites deliver a huge amount of data due to their unprecedented high spatial resolution. This data is used to extract information on atmospheric constituents including trace gases, e.g. ozone. The key component of retrieval algorithms is radiative transfer models which are quite time-consuming. In this regard, new retrieval approaches are required to cope with near real-time requirements. To accelerate data processing, machine learning approaches are planned to be used in the new generation of atmospheric processors. There has been already a success with the so-called physically-based machine learning methods, in which the complex radiative transfer models are parameterized by artificial neural networks. Despite, significant performance enhancement in radiance simulations, the retrieval procedures meet several difficulties since the problem is severely ill-posed. In this work, an alternative method is to use machine learning techniques to analyze the data which has been already processed by using conventional retrieval algorithms. In this way, a fast operator can be derived which converts the measurements into desired atmospheric parameters (ozone total column, for this work), and the radiative transfer model is included in the training process indirectly. Here, artificial neural networks are trained using real and synthetic data. The goal is to derive a fast yet stable operator for retrieving ozone total column from spectral radiances measured by TROPOMI. The efficiencies of dimensionality reduction techniques, linear and non-linear regression schemes are analyzed, as well. In particular, an artificial neural network trained on real data was able to retrieve ozone total column with an accuracy of 99.93%.

Zusammenfassung

Die neue Generation von Sensoren für die atmosphärische Zusammensetzung an Bord von Satelliten liefert aufgrund ihrer beispiellosen hohen räumlichen Auflösung eine enorme Datenmenge. Diese Daten werden verwendet, um Informationen über atmosphärische Bestandteile, einschließlich Spurengasen wie zum Beispiel Ozon, zu gewinnen. Die Schlüsselkomponente von Abrufalgorithmen sind Strahlungstransfermodelle, die sehr zeitaufwendig sind. In dieser Hinsicht sind neue Abrufansätze erforderlich, um Anforderungen nahezu in Echtzeit erfüllen zu können. Um die Datenverarbeitung zu beschleunigen, sollen in der neuen Generation atmosphärischer Prozessoren Ansätze für maschinelles Lernen verwendet werden. Es hat bereits Erfolge mit den sogenannten physikalisch basierten maschinellen Lernmethoden gegeben, bei denen die komplexen Strahlungstransfermodelle durch künstliche neuronale Netze parametrisiert werden. Trotz erheblicher Leistungssteigerung in Radiance-Simulationen stoßen die Abruf Verfahren auf mehrere Schwierigkeiten, da das Problem sehr schlecht gestellt ist. In dieser Arbeit besteht ein alternativer Ansatz darin, maschinelle Lerntechniken zu verwenden, um die bereits verarbeiteten Daten unter Verwendung herkömmlicher Abrufalgorithmen zu analysieren. Auf diese Weise kann ein schneller Operator abgeleitet werden, der die Messungen in gewünschte atmosphärische Parameter (Ozongesamtkolonne für diese Arbeit) umwandelt. Dadurch wird das Strahlungstransfermodell indirekt in den Trainingsprozess einbezogen. Hier werden künstliche neuronale Netze mit realen und synthetischen Daten trainiert. Ziel ist es, aus den von TROPOMI gemessenen spektralen Strahlungsdichten einen schnellen und dennoch stabilen Operator für die Ermittlung der Ozonsäule abzuleiten. Die Effizienzen von Dimensionalitätsreduktionstechniken, linearen und nichtlinearen Regressionsschemata werden ebenfalls analysiert. Insbesondere ein künstliches neuronales Netzwerk, das auf realen Daten trainiert war, konnte eine Ozon-Gesamtsäule mit einer Genauigkeit von 99,93% abrufen.

Contents

Declaration iii

Acknowledgement v

Abstract vii
English . . . vii
German . . . ix

Contents xii

List of Tables xiii

List of Figures xv

List of Acronyms xvii

1 Introduction 1
1.1 Remote Sensing & Atmospheric Science . . . 1
1.1.1 Ozone . . . 2
1.2 Radiative Transfer Model . . . 2
1.3 Machine Learning . . . 4
1.4 Objective . . . 5

2 Related Work 6

3 Dataset 7
3.1 Sentinel-5 Precursor . . . 7
3.1.1 TROPOspheric Monitoring Instrument . . . 8
3.1.1.1 Data Product . . . 9
3.2 Parameters . . . 10
3.2.1 Earth Radiance . . . 10
3.2.2 Solar Irradiance . . . 11
3.2.3 Zenith Angles . . . 12
3.2.4 Azimuth Angles . . . 12
3.2.5 Surface Albedo . . . 13
3.2.6 Ozone Total Column . . . 13

3.3 Preparation of Real Dataset . . . 14
3.3.1 Retrieval Algorithm for Level-2 Data Product . . . 14
3.3.1.1 NRTI Differential Optical Absorption Spectroscopy . . . 15
3.3.1.2 GOME Direct-fitting . . . 16
3.4 Preparation of Synthetic Dataset . . . 17

4 Methodology **19**
4.1 Inverse Model Parameterization . . . 19
4.2 Dimensionality Reduction Technique . . . 19
4.3 Linear Regression . . . 21
4.4 Neural Network . . . 23
4.4.1 NN Architecture . . . 25
4.4.2 Activation Function . . . 25
4.4.3 Levenberg-Marquardt Backpropagation . . . 26

5 Implementation **28**
5.1 Linear Regression . . . 29
5.1.1 Case 1: Synthetic Dataset for All Clusters . . . 29
5.1.2 Case 2: Synthetic Dataset for One Cluster . . . 29
5.1.3 Case 3: Real Dataset . . . 29
5.1.4 Case 4: Real Dataset with Albedo = (0.1 to 0.2) . . . 30
5.1.5 Case 5: Real Dataset with Solar Zenith Angle = (0° to 20°) . . . 30
5.2 Neural Network . . . 30
5.2.1 Case 1: Synthetic Dataset for One Cluster . . . 31
5.2.2 Case 2: Synthetic Dataset for All Clusters . . . 32
5.2.3 Case 3: Real Dataset . . . 32

6 Results and Discussion **35**
6.1 Linear Regression for Synthetic Dataset: All Clusters . . . 36
6.2 Linear Regression for Synthetic Dataset: One Cluster . . . 36
6.3 Linear Regression for Real Dataset . . . 38
6.4 Linear Regression for Real Dataset: Albedo = (0.1 to 0.2) . . . 40
6.5 Linear Regression for Real Dataset: Solar Zenith Angle = (0° to 20°) . . . 42
6.6 Neural Network for Synthetic Dataset: One Cluster . . . 44
6.7 Neural Network for Synthetic Dataset: All Clusters . . . 50
6.8 Neural Network for Real Dataset . . . 55

7 Conclusion **62**
7.1 Summary . . . 62
7.2 Future Work . . . 63

Bibliography **65**

List of Tables

3.1 Spectral characteristics of TROPOMI spectrometers. 7
3.2 Albedo values for various types of surfaces. 13

5.1 Different NN configurations for non-reduced synthetic data for one cluster (cluster 10). 33
5.2 Different NN configurations for non-reduced synthetic data for all clusters. . 33
5.3 Different NN configurations for non-reduced real data for one day (04/04/18). 34
5.4 Different NN configurations for reduced synthetic or real data. 34

List of Figures

1.1 Distribution of various gases in the earth's atmosphere. 2
1.2 Schematic representation of forward model of RTM. 3
1.3 Sentinel 5P ozone total column. 5

3.1 Sentinel-5 Precursor. 8
3.2 TROPOMI measurement principal. 9
3.3 TROPOMI data product file naming convention. 10
3.4 Schematic representation of radiance, irradiance and reflectance phenomenon. 11
3.5 Schematic diagram of zenith and azimuth angles. 12
3.6 Processing chain of TROPOMI data products. 14
3.7 Pictorial representation of the two steps involved in NRTI DOAS. 16

4.1 Training scheme of ANN for retrieval problem. 19
4.2 Visual representation of PCA with 2 PCs. 21
4.3 Schematic representation of empirical orthogonal functions of TROPOMI data. 21
4.4 Linear regression using ordinary least square. 22
4.5 A schematic diagram of a neural network. 23
4.6 A 3-layer MLP neural network. 25
4.7 Activation functions. 26
4.8 A schematic diagram of Backpropagation. 27

5.1 Workflow. 28

6.1 Linear regression results for synthetic data for all clusters. 36
6.2 Linear regression results for synthetic data for one cluster (cluster 1). 37
6.3 Linear regression results for synthetic data for one cluster (cluster 10). . . . 37
6.4 Linear regression results for real data for 01/01/18. 38
6.5 Linear regression results for real data for 04/04/18. 39
6.6 Linear regression results for real data for 08/08/18. 39
6.7 Linear regression results for real data for 12/12/18. 40
6.8 Linear regression results for real data with SA (0.1-0.2) for 04/04/18. 41
6.9 Linear regression results for real data with SA (0.1-0.2) for 08/08/18. 41
6.10 Linear regression results for real data with SA (0.1-0.2) for 12/12/18. 42
6.11 Linear regression results for real data with SZA (0° to 20°) for 04/04/18. . . . 43
6.12 Linear regression results for real data with SZA (0° to 20°) for 08/08/18. . . . 43
6.13 Linear regression results for real data with SZA (0° to 20°) for 12/12/18. . . . 44

6.14 Histogram of percentage of relative errors of the OTC for NN with reduced synthetic spectra for one cluster (cluster 10). 45
6.15 Results for synthetic data cluster 10 using NN configuration 1. 46
6.16 Results for synthetic data cluster 10 using NN configuration 2. 47
6.17 Results for synthetic data cluster 10 using NN configuration 3. 48
6.18 Results for synthetic data cluster 10 using NN configuration 4. 49
6.19 Results for synthetic data cluster 10 using NN configuration 5. 50
6.20 Results for NN trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 1. 52
6.21 Results for NN trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 2. 53
6.22 Results for NN trained on reduced synthetic data (all clusters) and tested on reduced real data (04/04/18) using NN configuration 1. 54
6.23 Results for NN trained on reduced synthetic data (all clusters) and tested on reduced real data (04/04/18) using NN configuration 2. 55
6.24 Histogram of percentage of relative errors of the OTC for NN with reduced real spectra (04/04/18). 56
6.25 Results for real data (04/04/18) using NN configuration 1. 57
6.26 Results for real data (04/04/18) using NN configuration 2. 58
6.27 Results for real data (04/04/18) using NN configuration 3. 59
6.28 Results for real data (04/04/18) using NN configuration 4. 60
6.29 Results for NN trained on real data (04/04/18) and tested on real data (08/08/18) using NN configuration 2. 61

List of Acronyms

CH_2O	Formaldehyde
CH_4	Methane
CO_2	Carbon Dioxide
NO_2	Nitrogen Dioxide
O_3	Ozone
SO_2	Sulfur Dioxide
2-D	Two-dimensional
AMF	Air Mass Factor
ANN	Artificial Neural Network
AU	Astronomical Unit
CFCs	Chlorofluorocarbons
CNN	Convolutional Neural Network
CO	Carbon Monoxide
DLR	German Aerospace Center
DOAS	Differential Optical Absorption Spectroscopy
DU	Dobson Unit
EOF	Empirical Orthogonal Functions
ESA	European Space Agency
GODFIT	Gome Direct Fitting
GOME-2	Global Ozone Monitoring Experiment-2
IMF-ATP	Remote Sensing Technology Institute, Atmospheric Processors
IR	Irradiance
L1b	Level 1-b
L2	Level 2
LR	Linear Regression
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multi Layer Perceptrons
MSE	Mean Squared Error
NetCDF	Network Common Data Form
NIR	Near-infrared
NN	Neural Network
NRTI	Near-real-time
OFFL	Off-line
OTC	Ozone Total Column

PC	Principal Component
PCA	Principal Component Analysis
PCNN	Principal Component Neural Network
PCR	Principal Component Regression
RA	Radiance
RAA	Relative Azimuth Angle
ReLu	Rectified Linear Unit
RNN	Recurrent Neural Network
RPRO	Reprocessing
RTM	Radiative Transfer Model
S5P	Sentinel-5 Precursor
SA	Surface Albedo
SAA	Solar Azimuth Angle
SCD	Slant Column Density
SCIAMACHY	Scanning Imaging Absorption Spectrometer for Atmospheric Chartography
SP	Surface Pressure
SWIR	Short-wave Infrared
SZA	Solar Zenith Angle
TanH	Hyperbolic Tangent
TROPOMI	TROPOspheric Monitoring Instrument
UV	Ultraviolet
UVIS	Visible
UVN	Ultraviolet, Visible, & Near-infrared
VAA	Viewing Azimuth Angle
VCD	Vertical Column Density
VZA	Viewing Zenith Angle

1 Introduction

This chapter addresses the use of remote sensing in atmospheric science, the concept and problems faced while retrieving the atmospheric constituents using the conventional method (radiative transfer model (RTM)) thus setting the motivation for designing new retrieval approaches and the objective of this thesis.

1.1 Remote Sensing & Atmospheric Science

Remote sensing is the science of acquiring information about a target by an instrument without any physical contact [1]. This technique mainly uses electromagnetic radiation for obtaining information [2]. The term remote sensing was coined by Evelyn Pruitt of the Office of Naval Research in the 1960s [3]. However, the idea was first realized and practice in 1858 when the balloonist G. Tournachon captured photographs of Paris remotely from his balloon. Until the First World War, different platforms from pigeons to airplanes have been used for aerial photography for military purposes. By the 1960s, with the formation of space programs, the satellite remote sensing came into existence for imaging surfaces using several types of sensors from the spacecraft [4]. Over the last 50 years, remote sensing has developed at a great pace through technical advances in scientific instrumentation, optics, and rocketry. The period has seen a pioneering voyage of discovery, and Earth observation has come of age during this period [5].

Earth observation is the study of the earth and its atmosphere using space-based instrumentation. It is a new field and of much importance to the modern world [6]. Since the industrial revolution with the availability of inexpensive energy and materials from fossil fuels, both the population of the earth and its standard of living, have been increasing tremendously. Such activity has resulted in air pollution, climate change, and global warming intensifying on local scales and expanding regionally and globally [7]. This has urged the need to study the distributions and amounts of trace gas constituents in the atmosphere, using satellite instruments orbiting in space, which is having a large influence on both monitoring the global and regional atmospheric environment and within the research field of Earth Observation [7].

Remote sensing methods are increasingly being used to quantify and draw connections between rapidly changing climatic conditions and environmental impact [8]. Atmospheric parameter retrieval (in particular, trace gas retrieval) is an important application of remote sensing [9]. Apart from nitrogen, oxygen, and argon that contributes up to 99.93% of gases in the atmosphere, there are other gases such as carbon dioxide, sulfur dioxide, ozone, methane, nitrous oxide, etc. available in the atmosphere called the trace gases (see Figure

1.1), as they are present in very low concentration, yet important to maintain the climate and life on Earth [10]. Atmospheric composition sensors onboard satellites provide a huge amount of data of high spatial resolution about atmospheric constituencies [11]. These sensors measure the spectral radiances reflected by the terrestrial atmosphere [2]. For instance, the newest sensor TROPOMI [5] onboard sentinel-5 precursor (S5P) provides a wealth of atmospheric data that can be used to retrieve trace gases such as ozone (O_3) in particular for this work.

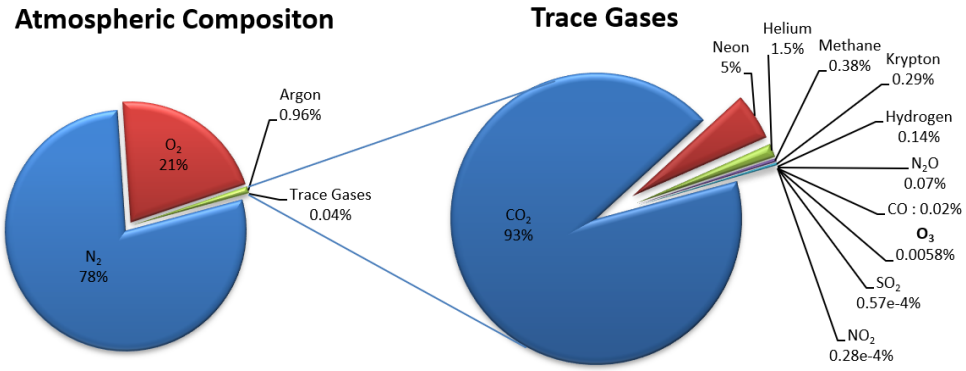


Figure 1.1: Distribution of various gases in the earth's atmosphere.

1.1.1 Ozone

Ozone is one of the most important trace gases in the atmosphere, even though present in very low concentration approximately 10 parts per million volume (ppmv) [12]. It is found in the troposphere and stratosphere layer of the atmosphere. In the lower stratosphere, it forms the ozone layer that blocks most of the harmful ultraviolet shortwave radiation from the sun (in particular UV-B) from penetrating the atmosphere [12]. In the troposphere, it acts as a cleaning agent but due to its high tendency to react with other molecules at the earth's surface, it is considered toxic to living organisms. It forms photo-chemical smog near the earth's surface which is a health risk. It is also harmful to plant life and warms up the atmosphere. With the increase in human-produced chemicals such as chlorofluorocarbons (CFCs) emitted from the use of refrigerators, air conditioners, etc the ozone layer is depleting [13]. It is thus, of utmost importance to retrieve ozone total column to monitor its distribution in the atmosphere.

1.2 Radiative Transfer Model

The radiative transfer concept was first presented in the 1950s by Chandrasekhar Subrahmanyan. The process of energy transfer in the form of electromagnetic radiation affected by the absorption, scattering and emission processes is termed as radiative transfer [8]. It is these interactions that are described in the radiative transfer equation [14]. The radiative

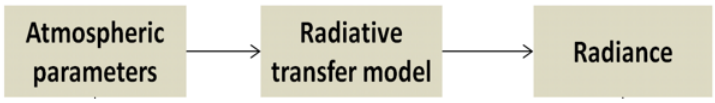


Figure 1.2: Schematic representation of forward model of RTM.

transfer equation is defined as follows [14]:

$$\frac{1}{c}\frac{\partial}{\partial t}I_\nu + \hat{\Omega}\cdot\nabla I_\nu + \left(k_{\nu,s}+k_{\nu,a}\right)I_\nu = j_\nu + \frac{1}{4\pi}k_{\nu,s}\int_\Omega I_\nu d\Omega \tag{1.1}$$

where c is the speed of light, j_ν is the emission coefficient, $k_{\nu,s}$ is the scattering opacity, $k_{\nu,a}$ is the absorption opacity and the integral part represents radiation scattered from other directions onto a surface [14]. It is a monochromatic equation which calculates the radiance's of a single layer of the earth in the forward model of RTM. By forward model it means that the atmospheric parameters such as ozone total column (OTC), optical angles and surface albedo are fed as input to the RTM to produce radiances as output as illustrated in Figure 1.2. In order to calculate the radiance for a spectral region with a fixed width, one has to integrate over a band of frequencies (or wavelengths).

The RTMs are the key components of the algorithms designed for the retrieval of atmospheric constituents from remote sensing data. The RTMs encompasses the understanding of the physics behind the measurement process and relates the optical parameters of the medium with the signal measured by the sensor. In the framework of the conventional approach, the retrieval problem is reduced to an exercise in optimization. Following Tikhonov [15], the retrieval algorithm finds the state vector x of medium parameters that minimize the discrepancy between the simulated and measured spectra in the following sense:

$$x = \underset{x}{\operatorname{argmin}}\left\{\|y - RTM\left(x\right)\|^2 + \alpha\Omega\left(x\right)\right\} \tag{1.2}$$

where y is the vector of measurements (in the case of atmospheric retrievals, spectral radiances at the top of the atmosphere), Ω is the stabilizing function and α is the regularization parameter. The expression in {} is referred to as the Tikhonov function. To perform minimization according to Eq.1.2 by using the Gauss-Newton method, one often requires the Jacobian matrix consisting of partial derivatives of RTM with respect to entries of x. The RTMs with capabilities to provide the Jacobian matrix are called linearized models. The RTM simulations are quite time-consuming and therefore introduce a performance bottleneck in operational retrieval algorithms. In this regard, new approaches are becoming increasingly important for designing new generation algorithms for interpretation of the optical signal [11]. With the tremendous success of machine learning approaches in solving light engineering and geoscience problems [16] including object detection and image recognition [17], using

Eq. 1.2 as the starting point and accelerating hyperspectral RTM using unsupervised machine learning (i.e dimensionality reduction) in [18], [19], machine learning seems to be a possible solution to the above problem.

1.3 Machine Learning

With the ever-increasing availability of a large amount of data and high computational power, fast and smart analysis of the data has become of paramount importance. In this context, machine learning (ML) techniques have proven to be an excellent tool. Tom M. Mitchell defines ML as:

> *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [20].*

Machine learning is broadly divided into two different categories based on their learning styles, namely unsupervised and supervised learning. In unsupervised learning, there is no information about the set of observations, it learns the structure of the data without using predefined labels. This method uses clustering to divide the datasets into one or more homogeneous sub-regions [21]. In supervised learning, there are predefined labels associated with the training samples. This method uses a function that approximates the relationship between the training samples and their predefined labels. ML is further divided into classification and regression methods depending on whether the response is quantitative or qualitative. If the response is a finite number of discrete category it is a classification problem. If the response is a continuous variable it is a regression problem [22].

The machine learning process is carried out based on the following steps[23]:

1. *Gathering of data:* Based on the problem to be solved, a relevant dataset is gathered. A large amount of data is required for a good predictive model.
2. *Preparation of data:* After the above step, the data is cleaned to remove duplicates, missing values, errors, etc. and split into two sets, train and test set for analysis
3. *Selection of ML algorithm:* Based on the problem to be solved best suited ML algorithm is chosen
4. *Building & Training the model:* This is the major step where the model is trained incrementally on the training set to improve its ability to predict
5. *Evaulating the model:* Here, the test set that is created during step 2 is used to check how well the model is performing on a completely new unseen data. This is a representation of how well the model will perform in the real world
6. *Fine-tuning the model:* Based on the above results the model is fine-tuned by tweaking its hyper-parameters for improved performance
7. *Generating performance graph*: Lastly, the model's performance score is calculated

This work closely follows the steps laid above. In Chapter 3 the gathering and preparation

of dataset is explained, Chapter 4 explains the various methods selected for solving the problem, Chapter 5 emphasizes on building, training, evaluating and fine-tuning the model and Lastly, Chapter 6 compares the performance and efficiencies of various methods tested.

1.4 Objective

The German aerospace center (DLR)-IMF-ATP department derives geophysical atmospheric parameters from remote sensing data with the focus on trace gas concentrations and characterizing clouds and aerosols. It is responsible for deriving ozone total column information from the TROPOMI sensor, which is on board Sentinel-5 precursor (S5P) shown in Figure 1.3. These sensors have more sophisticated features than passive spectrometers in the past. This includes spatial resolutions higher by a factor of 100 which requires managing a higher data rate while meeting the same or even more stringent specifications on processing time. The radiative transfer and retrieval expertise needs further development to meet the requirements of the Big Data era in earth observation. In order to cope up with the near real-time requirements of retrieval of trace gases, the objective of this thesis is to train an artificial neural network (ANN) in the backward direction (see Section 4.1) using already processed data by conventional approach and derive a stable yet fast operator to retrieve ozone total column from spectral radiances measured by TROPOMI. For this work supervised machine learning approaches such as linear regression (see Section 4.3) and neural network (see Section 4.4) are analyzed on synthetic and real measurements. Unsupervised dimensionality reduction technique (see Section 4.2), namely principal component analysis is also analyzed.

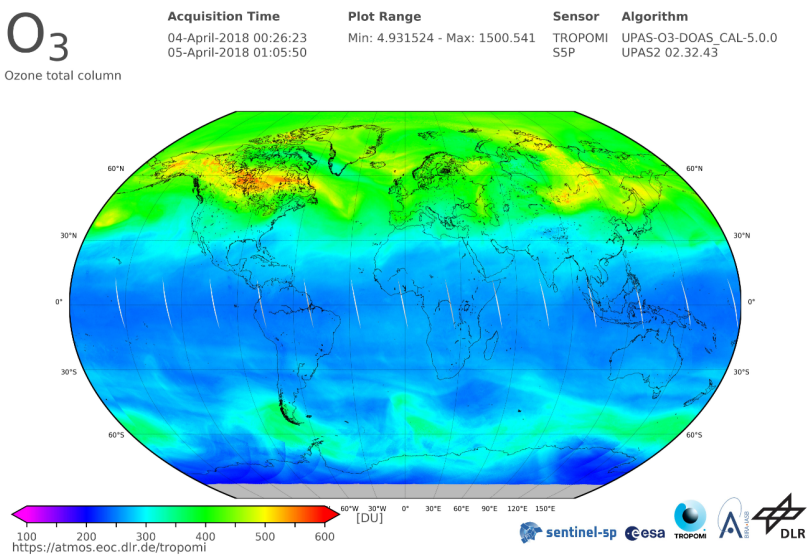


Figure 1.3: Sentinel 5P ozone total column.

2 Related Work

Since ML techniques have not been widely used for retrieving atmospheric constituents from remote sensing data, only a few studies have been done for the retrieval using ANN solely. Most of the work has focused on parameterizing the RTMs with the ANN from which we can draw certain conclusions for this work.

In the recent work by B.D Bue et al. [21] to retrieve surface reflectance over VSWIR spectral band, they parameterized the forward model of RTM using ANN. For this, they fed atmospheric parameters such as optical thickness, the single scattering albedo, the phase function for each atmospheric layer, observation geometry and the surface properties (e.g. the surface albedo) for a spectrum as input to dedicated hyperspectral RTM which provides a spectrum of radiances. These input-output parameters were then used to train the ANN in sub-networks which reproduced already convolved spectra with a mean absolute error below 1% over O_2A band. In the retrieval algorithm in Eq. 1.2, the trained ANN replaced the original RTM thereby avoiding the solving of the differential equation, while the rest part of the retrieval algorithm remained untouched.

In the work by J. Xu et al. [13] they parametrized the inverse model of RTM. For this, ANN was trained in the backward direction, using parameters to be retrieved as outputs and the rest of optical parameters and spectral radiances as inputs. Here, the backward trained ANN was applied to the ozone profile retrieval problem. The ANN training was preceded by the classification procedure, which grouped ozone profile shapes and corresponding spectral radiances in several clusters. For each cluster the inverse operator was trained, thereby restricting the space of parameters to be considered for training and making the retrieval procedure more stable overall. The main advantage of the backward trained ANNs over the classical optimization approach (Eq.1.2) was that the time-consuming training involving RTMs was performed once and offline.

The backward trained ANN were also applied to retrieve CO_2 in the work by M. Kataev et al. [9] and SO_2 plume height retrieval from ultra-violet spectra in the work by P. Hedelt et al. [24]. For the SO_2 plume height retrieval the linear regression model provided the error distribution with the standard deviation of about 1.5 km, while the ANN reduced it to 0.2 km.

3 Dataset

This work uses already processed real and synthetic measurements as datasets for machine learning methods to retrieve the ozone total column. The real measurements are from S5P satellite (see 3.1) and the synthetic measurements are generated using the forward radiative model (see 3.4). From, hereafter these measurements are referred to as real and synthetic datasets. The following sections provide detailed information about the source, nature, various parameters involved for real data and preparation of both the datasets for further experiments.

3.1 Sentinel-5 Precursor

The S5P satellite (see Fig.3.1), as one of ESA's Copernicus mission is dedicated to monitoring the earth's atmosphere. It provides atmospheric measurements that can be used for air quality, climate forcing, and ozone layer monitoring, with a high spatio-temporal resolution. The satellite's payload called TROPOspheric monitoring instrument (TROPOMI) consists of an imaging spectrometer with eight bands covering ultraviolet (UV), visible (UVIS), near-infrared (NIR), and short-wave infrared (SWIR). The UV spectrometer is used for medium-wave ultraviolet whereas the UVIS is used for long-wave ultraviolet combined with visible wavelengths. Based on this extensive spectral range, the instrument can measure key atmospheric constituents such as O_3, sulfur dioxide (SO_2), methane (CH_4), carbon monoxide (CO), nitrogen dioxide (NO_2), formaldehyde (CH_2O), aerosols, and clouds [25] as shown in Table 3.1.

TROPOMI spectral bands – based on calibration data

Spectrometer	UV		UVIS		NIR		SWIR	
Band ID	1	2	3	4	5	6	7	8
Spectral range [nm]	267-300	300-332	305-400	400-499	661-725	725-786	2300-2343	2343-2389
Spectral resolution [nm]	0.45 - 0.5		0.45 - 0.65		0.34 - 0.35		0.227	0.225
Spectral sampling [nm]	0.065		0.195		0.126		0.094	
Spatial sampling [km^2]	7.1 x 28.8	7.1 x 3.6	7.1 x 3.6		7.1 x 3.6		7.1 x 7.5	
Detector binning factor	16	2	2	2	2	2	1	1
Minimum signal-to-noise ratio	50*	50-600*	100-1200*	1200*	500*	200-600*	100-120**	

*Based on simulations for low albedo mid-latitude radiance **Based on design values

Table 3.1: Spectral characteristics of TROPOMI spectrometers.[25]

3.1.1 TROPOspheric Monitoring Instrument

TROPOMI passively measures the solar radiation reflected by and radiated from the earth. It operates in push-broom mode (non-scanning) to map the earth's atmosphere on a two-dimensional (2-D) image detector. For every 1 second, the instrument measures a strip of the earth's surface of dimensions approx. 2600 km across the track and 7 km in the along-track direction of the satellite as illustrated in Figure 3.2. Each square in the image represents a ground pixel. TROPOMI provides about 10^7 ground pixels each day with a ground pixel size of approximately 7x3.5km^2[1] at nadir which is two order higher in magnitude compared to GOME-2, SCIAMACHY & other previous sensors. Also, the signal measured is 1000 times stronger than the noise (see Tab.3.1) thus a very high precision model is required to retrieve the measurements from TROPOMI. The detector's across-track direction is used for detecting ground pixels and the along-track for wavelengths [5]. The instrument provides two types of data products, namely level 1-b (L1b) and level 2 (L2). These data are freely available to the public on ESA Copernicus Open Access Hub. All the products are stored in network common data form (NetCDF)-4 format.

Figure 3.1: Sentinel-5 Precursor.
Source: https://earth.esa.int/web/guest/missions/esa-eo-missions/sentinel-5p

[1] spatial resolution for measurements before August 6 2019, since then, the resolution has been improved to 5.5*3.5 km^2

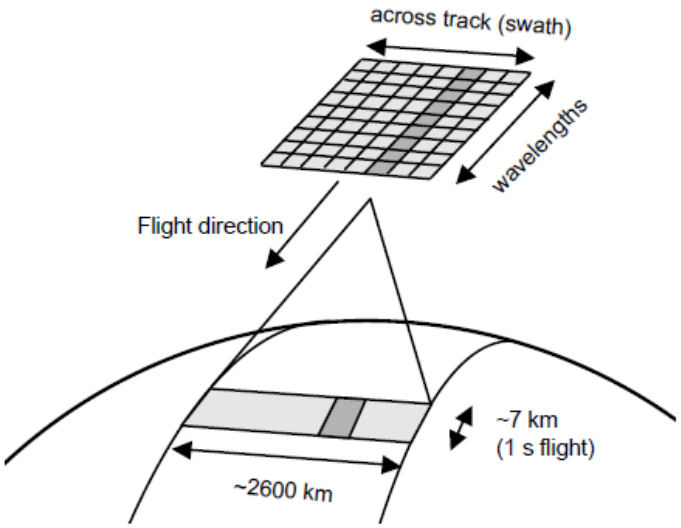


Figure 3.2: TROPOMI measurement principal. The dark-grey ground pixel is imaged on the 2-D detector as a spectrum. All ground pixels in the 2600 km wide swath are measured simultaneously.[5]

3.1.1.1 Data Product

The L1b data product consists of two types, the earth radiance spectra including the geometry data and the solar irradiance spectra whereas the L2 data product consists of information about the atmospheric constituents like the O_3, clouds, NO_2, aerosols, etc. For each spectral band, there is one L1b radiance (RA) product denoted as *L1B_RA_BD#* where BD# denotes the spectral band ranging from 1 to 8. The irradiance (IR) product has two variations, ultraviolet, visible, & near-infrared (UVN) for band 1 to 6 and SWIR for band 7 and 8 denoted as *L1B_IR_UVN* and *L1B_IR_SWIR*. To retrieve ozone total column, L1b RA for band 3 and IR UVN component are taken into consideration. There are three different data processing modes that provides the near-real-time (NRTI), off-line (OFFL), and reprocessing (RPRO) products. The NRTI product is available within 3 hours after acquisition, OFFL is available within few days and the RPRO is the latest version. These are distinguished by means of their filenames [26]. The logical file name convention of TROPOMI data product is illustrated in Figure 3.3.

Here is a full example of physical filename for L1b containing RA measurements of band 3 in NetCDF format: *S5P_RPRO_L1B_RA_BD3_20180404T000447_20180404T014617_-02449_01_010000_20180502T201749.nc*. Each calendar day has approx. 15 orbits and the RA component is recorded for each orbit whereas the IR component is recorded once per day by the solar IR port of TROPOMI. If no solar irradiance measurements are available no irradiance product would be generated for that day.

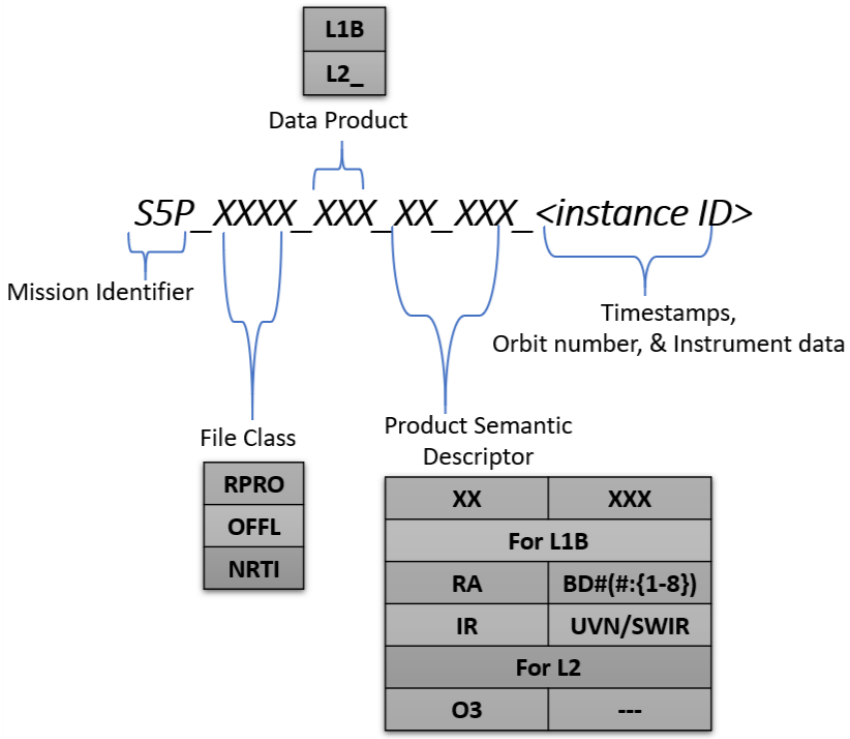


Figure 3.3: TROPOMI data product file naming convention.

3.2 Parameters

In this work, official TROPOMI data products are taken into consideration as input or output for machine learning methods. From L1b the earth radiance, solar irradiance and geometry parameters such as zenith and azimuth angles are selected. From L2 the surface albedo (SA) and OTC are selected. The following sections (see 3.2.1 to 3.2.6) describe in detail these parameters.

3.2.1 Earth Radiance

Earth radiance is the amount of energy (radiant flux) reflected, emitted or transmitted by the earth's surface and the atmosphere per unit solid angle per unit projected area as shown in Figure 3.4 [27]. The SI unit is watt per steradian per square meter ($W.sr^{-1}.m^{-2}$). Spectral radiance is the amount of energy received or radiated per unit area per unit solid angle as a function of wavelength and is expressed as ($W.m^{-2}.sr^{-1}.nm^{-1}$). TROPOMI in actual provides spectral photon radiance measurements i.e the rate of photons per second received per unit area per unit solid angle as a function of wavelength and is expressed as mole per second per square meter per steradian per nanometer ($mol.s^{-1}.m^{-2}.sr^{-1}.nm^{-1}$) [2] and is normalized

[2] Mole (mol.) is used for calculating the amount of substance (i.e photons), where 1 mol. is equal to Avogadro's number $N_A = 6.02214076 \times 10^{23}$ photons

to Earth-Sun distance of 1 astronomical unit (AU) [3]. The radiance component of L1b is a function of time (=1), scanline [4], ground pixel, and spectral channel [5]. It is the main input to the L2 processor.

3.2.2 Solar Irradiance

Solar irradiance is the amount of solar power incident on the earth's surface per unit area as shown in Figure 3.4 [27]. Here as well the L1b product provides spectral photon irradiance that is normalized to Earth-Sun distance of 1AU and is measured in $mol.s^{-1}.m^{-2}.nm^{-1}$. The irradiance component of L1b is a function of time (=1), scanline (=1), pixel [6] and spectral channel. The L2 processor uses this irradiance component to calculate the reflectance [7] from the radiance data.

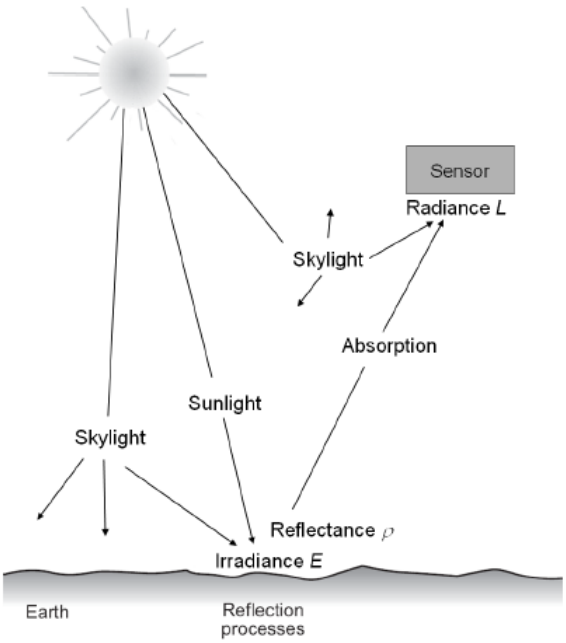


Figure 3.4: Schematic representation of radiance, irradiance and reflectance phenomenon.[27]

[3] 1AU = 149,597,870,700 meters
[4] Scanline is along-track dimension of the measurement and starts with 0 for each product
[5] Spectral channel denotes the wavelength dimension index
[6] Since during irradiance measurement the instrument is measuring solar irradiance and not imaging the earth surface hence referred to as pixel instead of ground pixel
[7] Reflectance of surface material is the measure of the proportion of incident light reflected in a particular wavelength range

3.2.3 Zenith Angles

The geometry parameters consist of two types of zenith angles, namely solar and viewing. These angles play an important role in determining from which location of the column of the atmosphere the detector is receiving the radiation measurements. The solar zenith angle (SZA) is the angle between the center of the sun and the zenith measured from the ground pixel whereas viewing zenith angle (VZA) is the angle between the line of sight to the satellite and the zenith measured from the ground pixel as illustrated in Figure 3.5 [28]. The angles are expressed in degrees. The L1b SZA and VZA component is a function of time (=1), scanline and ground pixel.

3.2.4 Azimuth Angles

There are two types of azimuth angle, namely solar and viewing. The solar azimuth angle (SAA) is the angle between the north and the sun, measured clockwise around the observer's horizon and the viewing azimuth angle (VAA) is the angle between the north and the satellite, measured clockwise around the observer's horizon. The geometry parameter used for this work is the relative azimuth angle (RAA) which is the relative difference of the SAA and VAA shown in Figure 3.5. It ranges from 0 to 180 degrees such that angles less than 90 degrees are for pixels between the satellite and the sun and the angles greater than 90 degrees are for pixels behind the sun [**?**]. The L1b RAA component is also a function of time (=1), scanline and ground pixel.

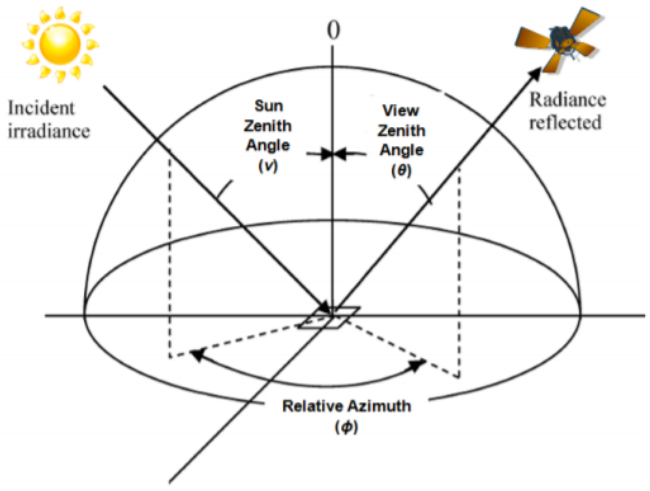


Figure 3.5: Schematic illustration of SZA, VZA and RAA.[29]

3.2.5 Surface Albedo

SA is a key factor in determining the atmospheric properties from the space. It is the amount of incident light reflected from the surface. The light that is not reflected is absorbed by the surface thereby raising the surface temperature and energizing the turbulent heat exchange between the lowest layer of the atmosphere and the surface [26]. Each type of surface has a different albedo value as depicted in Table 3.2. Surfaces covered with snow, sea ice and desert areas have relatively higher albedo value thereby reflecting a large fraction of incident light whereas forests, lakes, and oceans reflect a relatively smaller fraction of incident light and hence have a low albedo value. The albedo value ranges between 0 to 1 where values closer to 1 depict high albedo value and values closer to 0 depicts low albedo value of the surface. It is highly dependent on the wavelength and angular distribution of the incident light which is governed by the direction of light from the sun and the atmospheric composition [30]. The L1b SA component is a function of time (=1), scanline and ground pixel.

Surface	Albedo
Soil	0.05 - 0.40
Sand	0.15 - 0.45
Grass	0.16 - 0.26
Agricultural Crops	0.18 - 0.25
Tundra	0.18 - 0.25
Forest	0.05 - 0.20
Water	0.03 - 1.00
Snow	0.40 - 0.95
Ice	0.20 - 0.45
Clouds	0.30 - 0.90

Table 3.2: Albedo values for various types of surfaces.
Source: www.eoearth.org

3.2.6 Ozone Total Column

OTC is the total amount of ozone molecules in a column of air above the earth's surface from the troposphere to the top of the atmosphere. It also measures the ozone layer thickness. The unit of measurement is Dobson unit (DU) where 1 DU is equal to 0.01mm of ozone molecules in a column of air at 0℃ temperature and 1 atmospheric pressure. The distribution of ozone is not uniform through the vertical column. The value of ozone concentration varies from 100 to 600 DU and on average it is 300 DU in the atmosphere which is about 3mm thickness [31]. The so-called "ozone hole" is a region in the southern polar region where ozone is depleted severely by chemical reactions involving chlorine and bromine. The value of 220 DU is chosen as a baseline for interpreting the Antarctic ozone hole. The L1b OTC component is a function of time (=1), scanline and ground pixel. For this study, OTC in the

UV Huggins band (325-335nm) is considered as it provides information about the total ozone retrieval and the threshold mean absolute error (MAE) according to [32],[33] for OTC is 2-3% of 300 DU which is $\sim$ 6 to 9 DU.

3.3 Preparation of Real Dataset

This section would walk through the various steps carried out to process and prepare the data before it is available to the public for further analysis. The S5P raw data (i.e L0 data) collected by the instrument sensor along with the auxiliary data is passed to the L01b processor. Here, it undergoes various calibration by the processor to generate L1b data product. This data is then passed on to the L2 processor which applies two different retrieval algorithms, namely NRTI differential optical absorption spectroscopy (DOAS) (see 3.3.1.1) and gome direct fitting (GODFIT) (see 3.3.1.2) to produce L2 data (i.e ozone total column). The complete processing chain of TROPOMI data products is depicted in Figure 3.6. To prepare the real dataset for this work, four days over four seasons are chosen for the year 2018 (i.e 01/01/18, 04/04/18, 08/08/18 & 12/12/18) including all the orbits. The parameters (RA,IR, SZA, VZA, SAA, VAA, SA and OTC) are then extracted from the archive in .txt format with a reduced wavelength range = [325nm to 335nm], 0.195nm resolution and 54 spectra points. The RAA is then calculated using the SAA and VAA. The fill value = 9.96921e+36 is added for the missing data and SZA are masked to 90 degrees. In this work, the RAs are divided by their IRs component to obtain the reflectances exception is 01/01/18 data as no IR component was recorded on that day. It is these reflectances that are used for the implementation, for 01/01/18 data the radiance values are only used.

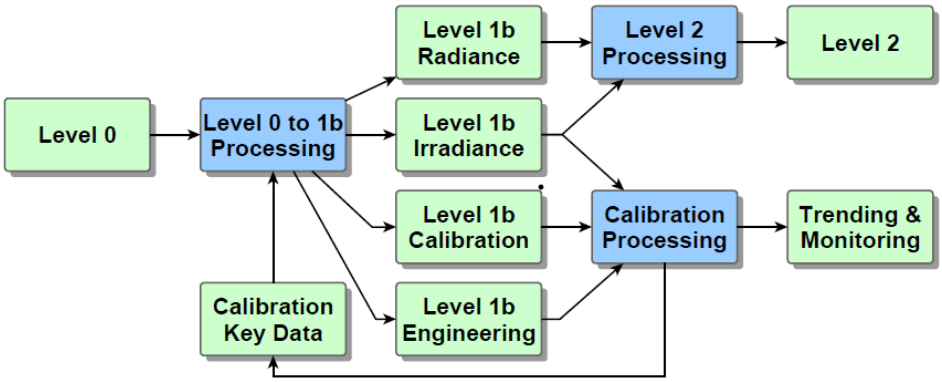


Figure 3.6: Processing chain of TROPOMI data products. The blue blocks denote processors; the green blocks denote data products.[26]

3.3.1 Retrieval Algorithm for Level-2 Data Product

This section describes in brief the two retrieval algorithms used to produce L2 data product from L1b. However, a complete explanation for the algorithm is out of scope of this work

and can be referred in [32]. The NRTI DOAS algorithm is described in the next section and GODFIT algorithm is described in 3.3.1.2.

3.3.1.1 NRTI Differential Optical Absorption Spectroscopy

NRTI DOAS is a widely used spectral fitting technique to determine the concentration of atmospheric trace gases in UVN range. It is based on the principle of absorption spectroscopy. It is a faster method compared to GODFIT and provides O_3 columns at the 1% level of accuracy and hence is the default retrieval algorithm for NRTI products. This method is divided into two steps at first it performs fitting of the slant column of effective total ozone based on the Beer-Lambert extinction law for trace gas absorption as illustrated in Equation 3.1 [32].

$$\tau_{sim}(\lambda) = -\sum_{g} N_{s,g}(\Theta)\,\sigma_g(\lambda) - \sum_{m=0}^{3} \alpha_m \left(1 - \frac{\lambda}{\lambda^*}\right)^m \tag{3.1}$$

where:

$\tau_{sim}(\lambda)$ = simulated optical density
λ = wavelength
Θ = geometrical path
$N_{s,g}(\Theta)$ = effective slant column density
$\sigma_g(\lambda)$ = associated trace gas absorption cross section
α_m = polynomial coefficient
λ^* = reference wavelength

The next step is to convert slant column density (SCD) N_s to the vertical column density (VCD) N_v of total ozone by using air mass factor (AMF) M using Equation 3.2 [32].

$$N_v = \frac{N_s}{(1-\Phi)\,M_{clear} + \Phi M_{cloud}} \tag{3.2}$$

where:

N_v = verical column density
N_s = slant column density
M_{clear} = clear-sky AMF
M_{cloud} = AMF for cloudy atmosphere
Φ = intensity-weighted cloud fraction

The AMF can include cloud effect corrections to determine the amount of trace gas

obscured by clouds. It also depends on the SA, vertical distribution of absorbing trace gases and the zenith angles. The AMF are defined as illustrated in Equation 3.3 [28]:

$$M = \frac{\ln(\frac{I_{nog}}{I_g})}{\tau_v} \tag{3.3}$$

where:

I_g = atmosphere with ozone

I_{nog} = atmosphere excluding ozone absorption

τ_v = vertical optical depth of ozone for the entire atmosphere

To compute the AMF, often calls are made to the RTM. The process of AMF/VCD is iterative, where each iteration computes the AMF and updates the VCD. After the convergence of this iteration, pixel processing is completed using destriping algorithm and L2 product is generated [32]. A pictorial representation of the two steps of DOAS is provide in Figure 3.7.

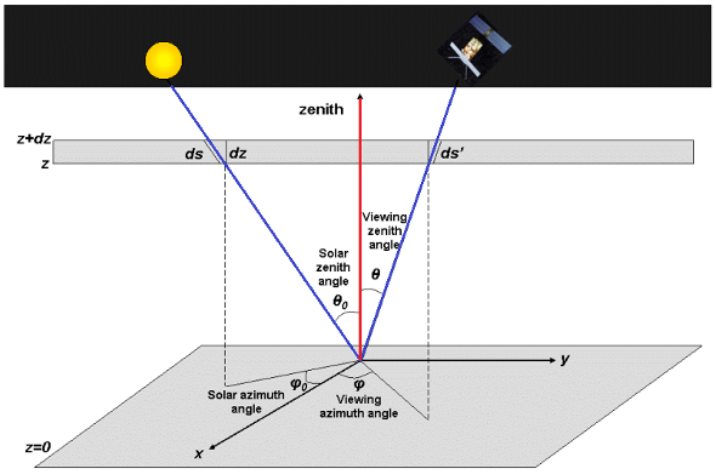


Figure 3.7: Pictorial representation of the two steps involved in NRTI DOAS. The blue lines are the optical path relevant to the SCD and the red line is relevant to the VCD
Source: https://earth.esa.int/web/sentinel/technical-guides/sentinel-5p/level-2/doas-method

3.3.1.2 GOME Direct-fitting

GODFIT provides a more accurate O_3 total column compared to NRTI DOAS but has slower computation performance. This method is mainly used for OFFL products. It is a single-step process and does not involve slant column separation and AMF computation. It uses iterative least square cost function minimization approach based on the differences between the measurements from the satellite and model-simulated radiances illustrated in Equation **??**.

The main requirement for this approach is the jacobian matrix consisting of partial derivatives of RTM concerning the state vector x in Equation 1.2. Since the computation of the jacobian matrix is quite time-consuming it slowdowns the retrieval process. An alternative to fasten the process is to use jacobians and radiances from pre-computed tables [32].

$$\mathfrak{F}(x) = \|G(x) - y\|^2 + \alpha_k \|L_k(x - x_k) - y\|^2 \tag{3.4}$$

where:

$G(x)$ = forward model
y = measurement from satellite
x = forward model simulated measurement
x_k = a priori
L = invertible square matrix
α_k = regularization parameter

3.4 Preparation of Synthetic Dataset

The OTC and ozone profiles are highly correlated and hence the outcome of clustering the ozone profile shape is used to retrieve the OTC. For this purpose, the synthetic data was generated by using the forward model VLIDORT [32], [13] with diverse ozone profiles based on the Equation 3.5.

$$y = F(x, b) + \delta \tag{3.5}$$

where:

y = noisy data vector
F = stands for forward model
x = state vector
b = model parameter vector
δ = noise vector

The state vector x consists of ozone concentration profiles at different pressure levels, b consist of SZA, VZA, RAA, SA and surface pressure (SP) and δ represents the artificial noise added to the simulated spectra for the better modeling of real dataset. The ozone profiles were clustered into groups based on their shapes and distribution of ozone using the k-means clustering method. It was grouped into 11 clusters, each consisting of 20 O_3 profiles, with spectral resolution of 0.125nm, 361 spectra points and 1000 sample points. Smart sampling technique was applied in order to cover the multi-dimensional input space in an optimized manner. The simulations were computed in the wavelength range from 290nm

to 335nm [13]. To prepare the synthetic dataset for this work, the simulated spectra with 81 wavelengths covering from 325 to 335 nm were chosen.

4 Methodology

This chapter explains in detail the new setup of the conventional retrieval model which is used indirectly for this work and the concepts of various ML algorithms used to solve the retrieval problem.

4.1 Inverse Model Parameterization

The RTMs, in general, uses the forward model approach as described earlier. However, this work uses the inverse model approach. Here, the ANNs are trained in the backward direction by using parameters to be retrieved i.e OTC as outputs of the model and the optical parameters along with spectral radiances as input to the model. This approach is illustrated in Figure 4.1. Through this approach, the time-consuming training involving RTMs is performed once and offline. Since the ANNs in this approach uses the output parameters already retrieved by conventional RTM and regularization procedure according to Equation 1.2, the RTM is captured by the ANN implicitly and thus, the prediction of ANN is still based on physical models comprising instrument specific features (e.g: noise, offset, wavelength calibration, etc) [11].

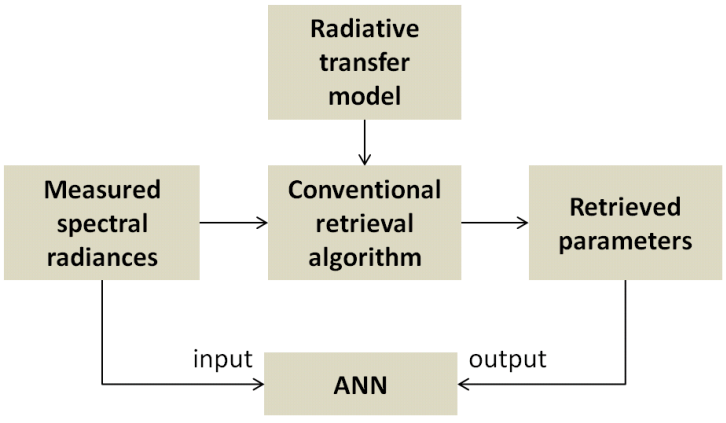


Figure 4.1: Training scheme of ANN for retrieval problem. [11]

4.2 Dimensionality Reduction Technique

Dimensionality reduction techniques are unsupervised algorithms. These are of two different forms: first, by selecting only the most relevant information from the original dataset called

the feature selection [34]. Second, by exploiting the redundancy of the input data and finding a smaller set of new variables, by the combination of original values and preserving the information in the original values called feature extraction [35]. One of the most commonly known dimensionality reduction feature extraction techniques is called principal component analysis (PCA) which is used for this work.

For PCA, given a set of x observations and M dimensions of the input dataset, it tries to find the directions of maximum variance in high-dimensional dataset and projects it into a single best linear subspace with fewer or equal dimensions using least square error for the given M dimensions [35]. To do so it follows the following steps [36]:

1. *Standarizing the data:* Here, the values of each dimension are brought into a comparable range, so that the output of the algorithm is non-biased. This is carried out by subtracting each value with the mean and dividing it by the overall standard deviation in the dataset
2. *Computing the covariance matrix:* It is a MxM matrix representing the correlation between the different variables in the dataset
3. *Compute the eigenvalues and eigenvectors:* These are computed through the above matrix to identify the principal component (PC) of the dataset. Here the eigenvectors represent the direction and the eigenvalues represent the magnitude of the input values
4. *Identify the PCs:* This step is done by sorting the eigenvector and eigenvalue in descending order, thereby the eigenvector with highest eigenvalue form the first PC. Here, the number of PCs are selected in a way that they represent the maximum information about the dataset
5. Reduce the dimension of the dataset based on the number of PCs selected from the above step

Figure 4.2 illustrates the transformation of a 3 dimensional data to a 2-D data using PCA. This work uses the inbuilt sci-kit learn PCA function that is based on the steps laid above and produces a reduced dataset based on the number of PCs passed to the function. Figure 4.3 illustrates the empirical orthogonal functions (EOF) of TROPOMI data. This technique is used in particular, to reduce the dimensions of the complex dataset, thereby reducing the input space and thus improving the computational speed for training the ML models [34]. This is also done since the dataset has a lot of attributes (i.e 54 or 81 spectra points for this work), thereby exist a high degree of redundancy or correlation between the different variables which can provide poor results. Also, this is used to get rid of the noisy data. One must be careful not to discard important information. Hence, to make sure there is no information loss due to this technique the raw data results are compared as well in this work.

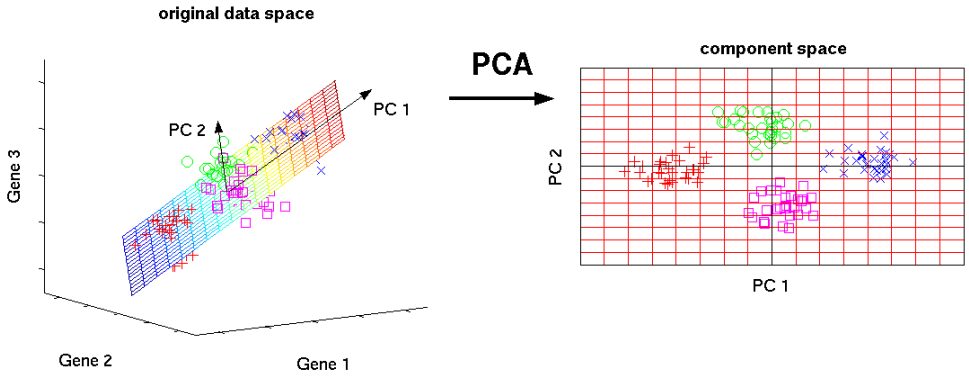


Figure 4.2: Visual representation of PCA with 2 PCs.
Source: http://www.nlpca.org/pca_principal_component_analysis.html

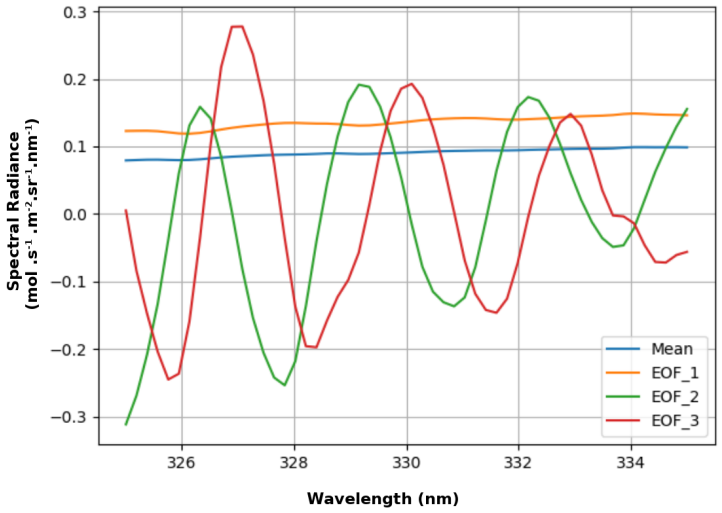


Figure 4.3: Schematic representation of EOF of TROPOMI data.

4.3 Linear Regression

The linear regression (LR) is one of the simplest and commonly used supervised ML algorithm to predict the values of a dataset [37]. It is a linear model as the name suggests. It is defined as a line fitting through a set of points and using this line to make predictions with a minimized error [37]. For, given set of input values x and output values y it tries to fit a line given by Eq.4.1 to the dataset in such a way that the error between the predicted and actual value is minimized. This is a hyper-plane instead of a line in case of higher dimensions i.e, with more than one input [38].

$$y = \beta x + \varepsilon \tag{4.1}$$

where:

$y =$ output (dependent variable)
$\beta =$ intercept
$x =$ input (independent variable)
$\varepsilon =$ error term

Learning a LR model means estimating the values of the coefficients used in the Eq.4.1 with the data that is passed as a train set [38]. There are numerous learning techniques for LR. The most common learning technique used for LR in ML and in this work is the ordinary least squares to estimate the coefficients. In this technique, given a regression line (or hyper-plane) through the data points, the distance between each data point and the regression line is calculated, squared, and all the errors are summed up together. It is this quantity that the ordinary least square tries to minimize as shown in Figure 4.4. It is very fast to calculate. This approach treats the data as a matrix and uses linear algebra operations to estimate the optimal values for the coefficients [37]. Thus, all of the data must be available and enough memory should be available to fit the data and perform matrix operations [38]. Thereafter, the estimated coefficients are plugged into the equation and the output values are predicted. This work uses the sci-kit learn linear regression function that follows the above principle.

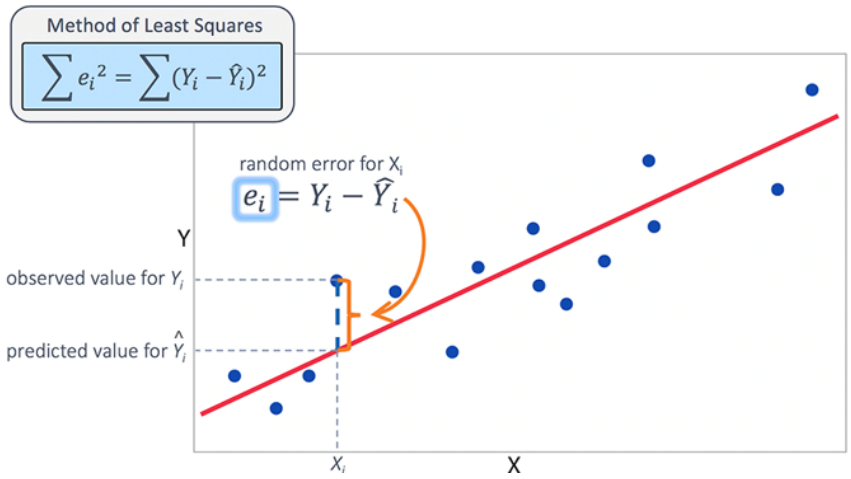


Figure 4.4: Linear regression using ordinary least square.
Source: https://www.jmp.com/en_hk/statistics-knowledge-portal/what-is-multiple-regression/fitting-multiple-regression-model.html

Along with LR, a special variant of regression analysis based on PCA is used in this work called the principal component regression (PCR). This method considers regressing the dependent variable (y) on a set of an independent variable (x) based on the standard LR model, however, it uses the PCs of the variable x (i.e the regressors) for estimating the unknown regression coefficients in the model [37]. In general, a subset of PCs with high variance is taken into consideration as regressors [39]. The main idea to use this method is

to avoid multicollinearity between predictors.

4.4 Neural Network

The neural networks (NNs) are the artificial systems that mimic the biological neural networks [40]. These are hence, often known as well the ANN. The NN is a supervised learning system built of a large number of neurons or perceptrons [17]. Each neuron accepts an input, makes simple decisions and feeds those results to the other neurons organized in interconnected layers [40]. The NN's first layer is always the input layer and the last layer is always the output layer. There is always one single input layer with the number of neurons equal to the input size (for classification)/ dimension (for regression) and a single output layer with the number of neurons equal to one (for regression) /based on the number of classes (for classification). The layer in between is called the hidden layer and can be one or more with no fixed rule for the number of neurons in the hidden layer [41]. A schematic diagram of a simple neural network is illustrated in Figure 4.5.

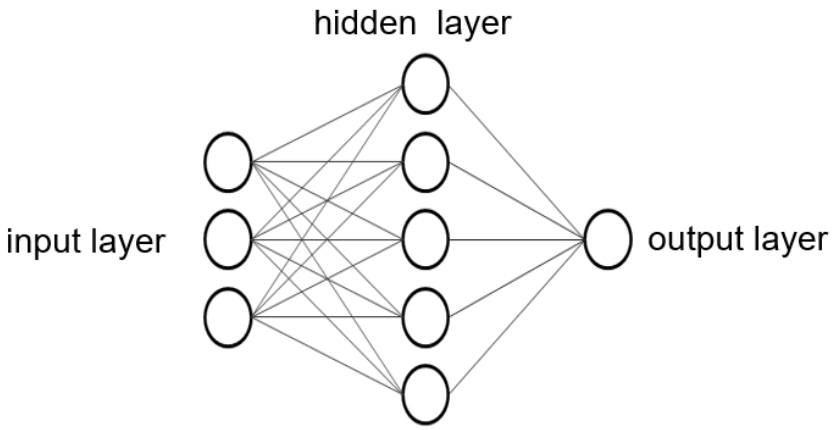


Figure 4.5: A schematic diagram of a neural network with one input, one hidden and one output layer.[11]

Each input is taken with an associated weight that can be changed to mimic synaptic learning [42]. The neuron computes its output O_i as follows [11]:

$$O_i = f\left(\sum_j w_{ij} I_j\right) \tag{4.2}$$

where:

f = node's activation fucntion

w_{ij} = weight from node j to node i

I_j = input to node j

O_i = output of the node that servers input for node i

The weights represent the strength of the connection between neurons [40]. The bias is a threshold value always equal to 1 that is added to each neuron's computation before passing to the next neuron. It is added to make sure that even when the inputs are 0 there is going to be activation in the neuron [42]. Activation functions or transfer function decides if the given neuron should be activated or not based on the weighted sum. It squashes the value into a smaller range. These are one of the deciding parameters that affect the convergence ability of the network [40]. After the above computation for each neuron, the network provides an output that is then verified with the expected output and the error is minimized using the cost function. A cost function measures how good a network performs for the expected output [42]. This is the generic working principle of any kind of neural network.

To successfully build & train a neural network several steps are to be carried out [40], [17], [42] :

1. *Selection of NN architecture:* There are several different types of NN architectures that varies depending on the structure of the network and in-depth working principle. The most common ones are the feed-forward neural network, convolutional neural network (CNN), recurrent neural network (RNN), etc
2. Selection of the number of hidden layers and neurons
3. *Selection of activation functions:* There are many different activation functions such as sigmoid, linear, hyperbolic tangent (TanH), rectified linear unit (ReLu), etc. that are used on each neuron.
4. *Selection of training algorithm/ optimization function:* Training algorithm is used for the learning process of the network, by setting the weights and biases during backpropagation to get an optimal result. These are also of different types such as gradient descent, Levenberg Marquardt, etc.
5. *Selection of cost functions:* There are several cost functions such as mean squared error (MSE), cross-entropy, etc.
6. Selection of input and output neurons based on the problem to be solved
7. *Creation of train, validate and test set:* The dataset that is fed to the network is called train set. These are labeled data i.e a set of inputs for which the correct outputs are known. This is used to train the network by learning the features or patterns of the data and thereby used to update the weights. The validation set is used to check how well the network converges and to make sure that the network is not overfitting. The test set is the unseen data that is used to finally examine the accuracy of the trained network.

8. *Setting number of epochs:* Epoch means the number of times the network is exposed to the entire training set. One epoch is equal to one forward + one backward pass of all the training samples

The implementation of NN in this work follows closely the above-mentioned steps. The next section describes in specific the NN model and its parameters that are used for this work. As the problem to address in this work is continuous in nature, only regression concepts will be focused. The implementation of NN is done using MATLAB toolbox hence some of the parameters explained in the following sections are specific to the toolbox.

4.4.1 NN Architecture

According to the universal approximation theorem, a feed-forward network with one hidden layer containing a finite number of neurons can approximate any continuous function on a subset of $\Re^n$, n is the number of inputs [42]. A feed-forward NN also called perceptrons is the simplest network with input and output layer where the input moves in the forward direction only [43]. Here, the input of i^{th} layer neuron is passed always to the $i+1^{th}$ layer neuron and so on. Based on the universal approximation theorem, a special case of this network called the multi layer perceptrons (MLP) which has two or more hidden layers is used for this work. Throughout this work, the MLP with 3 or 4 hidden layers are being implemented. A systematic diagram of 3 hidden layer MLP is illustrated in Figure 4.6.

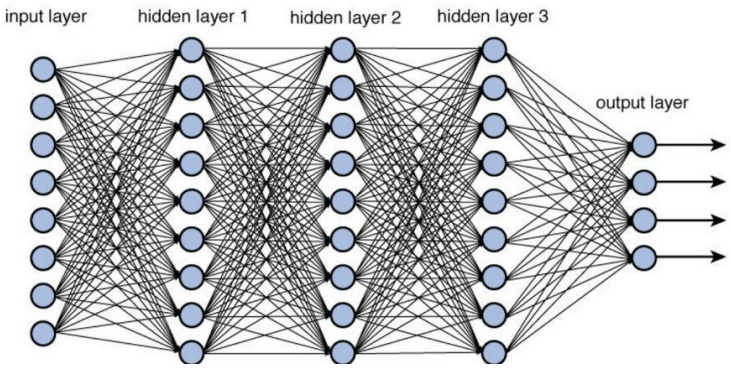
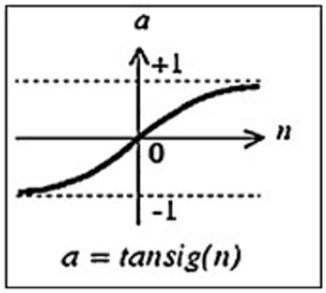


Figure 4.6: A 3-layer MLP neural network.
*Source: https://miro.medium.com/max/1200/1*N8UXaiUKWurFLdmEhEHiWg.jpeg*

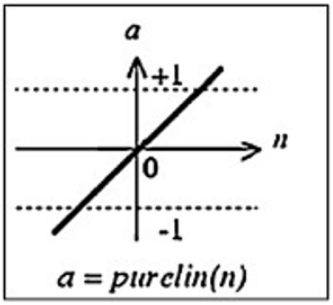
4.4.2 Activation Function

Activation functions help the model account for non-linearities and interaction effects between inputs, facilitating learning algorithm [40]. The hyperbolic tangent sigmoid (tansig) activation function is used for all the hidden layers as the optimization is easier in this method and works best for regression compared to other available activation functions in MATLAB toolbox and linear (purelin) activation function is used for the output layer. (see Figure 4.7).



tansig(n) = 2/(1+exp(-2*n))-1

(a) Tansig



Purelin(n) = n

(b) Purelin

Figure 4.7: Activation functions.
Source: Mathworks.com

4.4.3 Levenberg-Marquardt Backpropagation

The backpropagation algorithm was developed by Paul Werbos in 1974 and rediscovered independently by Rumelhart and Parker [44]. Since then it has been widely used as a learning algorithm in feed-forward multilayer NN. It is an algorithm used for supervised learning in NNs based on gradient descent to find optimal weights during the training process to improve efficiency [44]. As the name suggests the errors are propagated in the backward directions and the weights are updated accordingly this is illustrated in Figure 4.8. Since the traditional backpropagation algorithm has the drawback of getting stuck in a local minimum and converges slowly the choice is the Levenberg-Marquardt backpropagation (trainlm) as the main training algorithm [40]. It is best suited to work with MLP and loss functions that form a sum of squared errors [42]. It updates the weights and biases according to the Levenberg-Marquardt optimization that works without computing the exact Hessian matrix. Instead, it works with the gradient vector and the Jacobian matrix [45],[44]. Another training algorithm scaled conjugate gradient backpropagation (trainscg) is also being used to check if the performance is better for the real dataset. This algorithm updates weights and biases based on the scaled conjugate gradient method. However, trainlm is the first choice as it is the fastest algorithm and works best for regression problems.

The cost function measures how well parameters w and b are doing on the training set. [42] The function selected for optimization is MSE which takes first the difference between the estimated value and the original values, squared it and then average it over the number of samples as shown in Eq.4.3. Choosing a correct cost function is dependent on the datasets, its distribution, and scale [40].

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.3}$$

where:

N = number of samples

y_i = original value of y

$\hat{y}_i$ = estimated value of y

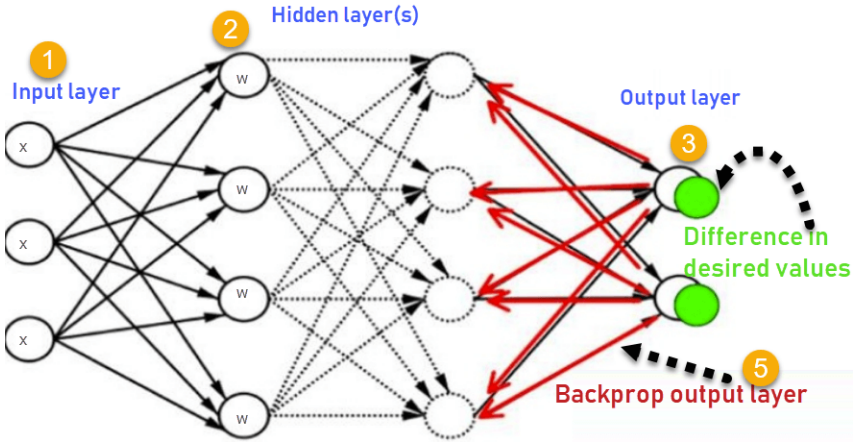


Figure 4.8: A schematic diagram of Backpropagation.
Source: https://www.guru99.com/images/1/030819_0937_BackPropaga1.png

A good practice to prevent overfitting and validate the model is to shuffle the labeled data and split it into training sets and validation sets, allowing the network to be trained and then tested on the data it never saw before, as to assess if the model is generalized or if it may be over or underfitting. It is also important to understand that a good dataset is representative of the most scenarios possible, to allow for generalization. Thus it is necessary to use a large number of training samples as well as a validation set to estimate the predictive capability of the network. Also, to obtain high generalization capability and unbiased estimates of the model, training and test set must be independent of each other [44], [42], [40].

5 Implementation

The retrieval of ozone total column is a non-linear inverse problem. To verify this, linear as well as non-linear ML schemes are tested for this experiment. This work proposes the workflow as shown in Figure 5.1. This includes the extraction of prepared dataset (see 3.3 and 3.4) based on specific cases, 1) synthetic data for all clusters, 2)synthetic data for one cluster and 3)real data for one day. This is then further altered and divided into train and test sets. These sets are then either passed directly to the linear regressor and NN or PCA is applied to these sets and then the resultant reduced set is passed to the linear regressor and the NN. Thereafter, histograms are plotted for each case and method and the efficiencies of different methods are compared (see Chapter 6).

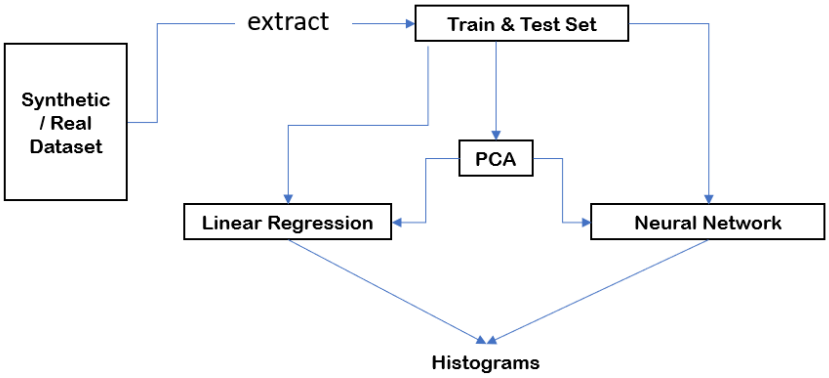


Figure 5.1: Workflow.

For this work, some of the common alterations (default settings) for both datasets before dividing into the train and test sets are:

1. All the angles (SZA, VZA, RAA) are converted to their cosine values
2. The SA values from 0.2 to 0.4 are chosen, other than explicitly mentioned
3. The fill values are filtered out for both datasets

The RAs, SZAs, VZAs, RAAs, and SAs are chosen as inputs and OTCs as outputs. The inputs from synthetic data consist of 85 spectra points (81 spectra points for RA, 4 spectra points including SZA, VZA, RAA & SA) and real data consist of 58 spectra points. All setups specific to the cases are described in their respective sections below. The implementation of this experiment is carried out in Python using scikit-learn for LR and NN toolbox from MATLAB for NNs.

The following sections explain in detail the two main algorithms mentioned in Figure 5.1

along with the various type of data used for each algorithm. Section 5.1 explains LR and 5.2 explains NN.

5.1 Linear Regression

For this method, two sub approaches are considered. The first approach called PCR is where the train and test set containing only the radiance values are passed to the PCA with 10 PCs for dimensionality reduction and this reduced set combined with other input parameters are then passed to train the linear regressor. The trained linear regressor is then applied to the reduced test set and errors are computed. The second approach is where the linear regressor is trained using the original train set and it is then applied to the original test set and then the errors are computed. For both the approaches, the test size is equal to 0.2. The following sections describe the various cases involved w.r.t the dataset on which the LR is applied.

5.1.1 Case 1: Synthetic Dataset for All Clusters

For case 1, all the 11 clusters of synthetic data are extracted consisting of 220,000 sample points, default settings are applied and then divided into train and test sets. The total number of train samples = 33,489 and test samples = 8,373. To these sets, PCR and LR is applied as explained above. Thereafter, the histogram of relative error of OTC is computed for both PCR and LR separately. These are illustrated and explained in 6.1.

5.1.2 Case 2: Synthetic Dataset for One Cluster

For case 2, one cluster of synthetic data is extracted consisting of 20,000 sample points, default settings are applied and then divided into train and test sets. For this case, two clusters are tested independently, cluster 1 and cluster 10. The total number of train samples varies between 3,066 to 6,119 and test samples vary between 767 to 1,530 depending on the selected cluster. To these sets of each cluster, PCR and LR is applied as explained in 5.1. Thereafter, the histogram of relative error of OTC for each cluster is computed for both PCR and LR separately. This is illustrated and explained in 6.2.

5.1.3 Case 3: Real Dataset

For case 3, the real data for one day is extracted, default settings are applied and then divided into train and test sets. For this case, all the four days (01/01/18, 04/04/18, 08/08/18 and 12/12/18) are tested independently. The total number of train samples varies between 92,688 to 354,745 and test samples vary between 23,173 to 88,687 depending on the selected day. To these sets of each day, again PCR and LR is applied as explained in 5.1

and the histogram of relative error of OTC for each day is computed for both PCR and LR separately. This is illustrated and explained in 6.3.

5.1.4 Case 4: Real Dataset with Albedo = (0.1 to 0.2)

For case 4, again the real data for one day is extracted but here the SA values between 0.1 to 0.2 are chosen. All other default settings remain the same. It is then divided into train and test sets. In this case, only three days are tested, independently. Since 01/01/18 data does not contain any IR component, the values do not resemble the true values of radiation and hence this data is discarded for further testing. The total number of train samples varies between 184,129 to 275,672 and test samples vary between 46,033 to 68,919 depending on the selected day. Here as well, PCR and LR is applied as explained in 5.1 to the sets of each day. Thereafter, the histogram of relative error of OTC for each day is computed for both PCR and LR separately. This is illustrated and explained in 6.4.

5.1.5 Case 5: Real Dataset with Solar Zenith Angle = (0° to 20°)

For case 5, again the real data for one day is extracted and here the SZA between 0° to 20° is chosen. All other default settings remain the same. It is then divided into train and test sets. For this case as well three days are tested independently. The total number of train samples varies between 184,129 to 275,672 and test samples vary between 46,033 to 68,919 depending on the selected day. To these sets of each day, PCR and LR is applied as explained in 5.1. Thereafter the histogram of relative error of OTC for each day is computed for both PCR and LR separately. This is illustrated and explained in 6.5.

5.2 Neural Network

For this method as well two sub approaches are considered. The first approach called principal component neural network (PCNN) is where the train and test set are passed to the PCA with 10 PCs for dimensionality reduction and this reduced set is then passed to the NN. This trained network is then applied to the reduced test set (unseen data). The second approach is where the original train set is passed to the NN and the trained network is then applied to the unseen original test set. For both the approaches, few steps are carried out before the actual training of the NN. These steps are as follows:

- Two *.txt* files are generated. The first file contains the input parameters (RAs, SZAs, VZAs, RAAs and the SAs) which are either shuffled or non-shuffled. The second file contains the output parameter OTCs these are also either shuffled or non-shuffled
- The train and test sets are created from the above two files
- The train set is then passed to the wrapper of the MATLAB library along with the number of hidden neurons for each hidden layer to be created

- The train set is further divided into 70% train and 15% validation set
- The type of training algorithm is also passed to the wrapper, for this work *trainlm* (see 4.4.3) training algorithm is used other than explicitly mentioned
- The transfer function for hidden layer is set to *tansig* (see 4.4.2) and *purelin* (see 4.4.2) for the output layer
- The batch size is set to the number of samples passed to the network
- The maximum validation fails are set to 40
- The epoch is set to 100000 initially

Once the train set is passed to the wrapper with all other parameters, the training is started by setting the weights and biases randomly for the first iteration. During each iteration the samples are passed, the weights and biases are updated based on the principle of the trainlm algorithm and the performance of each iteration is calculated. The validation set is also used simultaneously for each epoch without updating the weights and biases. The training stops if either the epoch is reached or the validation-fails reaches its maximum value. Thereafter, the best epoch is chosen and set as the new epoch value and the training is performed again till the new epoch value. After the end of the training, the training time and the best training performance is calculated. This trained network is then applied to the test set (unseen data).

The following sections describe the various cases involved w.r.t the dataset on which the PCNN and NN are applied. Each case data is fed to numerous NN configurations by fine-tuning the hyper-parameters to achieve the best NN configuration with the lowest error value.

5.2.1 Case 1: Synthetic Dataset for One Cluster

In this case, cluster 10 of the synthetic data is chosen, consisting of 85 inputs and a total of 3833 samples. The data is divided into train and test sets with 3000 train samples and 833 test samples. For the first sub approach, these sets are passed to the PCA as explained above and the reduced train set is then fed to the NN configuration as described in Table 5.4b. The trained network is then applied to the reduced unseen test set and the histogram of relative error for train and test set is plotted. This is illustrated and explained in Section 6.6.

For the second sub approach, the train samples are fed to the different NN configurations as described in Table 5.1. After training for each NN configuration it is applied to the test set (unseen data) and the histograms of relative and absolute errors for train and test set are plotted. This is illustrated and explained in Section 6.6.

5.2.2 Case 2: Synthetic Dataset for All Clusters

This is a special case, here both the sub approaches are trained on synthetic data for all 11 clusters and this trained network is then applied to the real data for one day (i.e 04/04/18). The particular real data is chosen randomly there is no specific reason. Since the spectral resolution and number of spectra points for the synthetic and real data are different, the RA value of real data (=54 spectra points) is interpolated to match the spectra points (=81) of the synthetic data. To this interpolated data, the other input parameters (i.e angles and albedo) are added before applying on the NN. The number of train samples = 41,862 (synthetic data) and test samples = 10,000 (real data) for both the approaches.

Here, the first approach is where the original train (synthetic data = 85 inputs) and test (interpolated real data = 85 inputs) sets are used. The train set is fed to different NN configurations as described in Table 5.2. For the second sub approach, the synthetic and interpolated real data consisting of 81inputs are passed to the PCA with 10 PCs for dimensionality reduction, then the angles and albedos are added and the train and test set with 14 inputs are formed. The reduced train set is then passed to different NN configurations as described in Table 5.4. After training for each NN configurations these trained networks are then applied to the original test set and reduced test set respectively. Thereafter for all the different NN configurations, the histograms of relative and absolute errors are plotted for both the approaches. This is illustrated and explained in Section 6.7.

5.2.3 Case 3: Real Dataset

In this case, one day (i.e 04/04/18) of the real data is chosen, consisting of 58 inputs and a total of 264,067 samples. The data is divided into train and test sets with 185,000 train samples and 79,067 test samples. For the first sub approach, these sets are passed to the PCA as explained in 5.2 and the reduced train set is then fed to the NN configuration described in Table 5.4b. The trained network is then applied to the reduced unseen test set and the histogram of relative error for train and test set is plotted. This is illustrated and explained in Section 6.8.

For the second sub approach, the train samples are fed to the different NN configurations as described in Table 5.3. After training for each NN configuration it is applied to the test set (unseen data). Here, the trained network on real data (04/04/18) is also applied to the real data for 08/08/18 to test the performance of the network when applied to a different season data. Thereafter, the histograms of relative and absolute errors for the train and each test set are plotted. This is illustrated and explained in Section 6.8.

NN Configuration 1	
Nature of Input	Non-shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	10, 5, 3

(a) NN configuration 1 for non-reduced synthetic data for one cluster (cluster 10).

NN Configuration 2	
Nature of Input	Non-shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	20, 10, 5

(b) NN configuration 2 for non-reduced synthetic data for one cluster (cluster 10).

NN Configuration 3	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	10, 5, 3

(c) NN configuration 3 for non-reduced synthetic data for one cluster (cluster 10).

NN Configuration 4	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	20, 10, 5
Training Algorithm	Trainscg

(d) NN configuration 4 for non-reduced synthetic data for one cluster (cluster 10).

NN Configuration 5	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	20, 10, 5

(e) NN configuration 5 for non-reduced synthetic data for one cluster (cluster 10).

Table 5.1: Different NN configurations for non-reduced synthetic data for one cluster (cluster 10).

NN Configuration 1	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	20, 10, 5

(a) NN configuration 1 for non-reduced synthetic data for all clusters.

NN Configuration 2	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	40, 12, 3

(b) NN configuration 2 for non-reduced synthetic data for all clusters.

Table 5.2: Different NN configurations for non-reduced synthetic data for all clusters.

NN Configuration 1	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	20, 10, 5

(a) NN configuration 1 for non-reduced real data for one day (04/04/18).

NN Configuration 2	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	40, 12, 3

(b) NN configuration 2 for non-reduced real data for one day (04/04/18).

NN Configuration 3	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	40, 12, 3
Training Algorithm	Trainscg

(c) NN configuration 3 for non-reduced real data for one day (04/04/18).

NN Configuration 4	
Nature of Input	Shuffled
Number of Hidden Layers	4
Number of Hidden Neurons	40, 15, 8, 3

(d) NN configuration 4 for non-reduced real data for one day (04/04/18).

Table 5.3: Different NN configurations for non-reduced real data for one day (04/04/18).

NN Configuration 1	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	20, 10, 5

(a) NN configuration 1 for reduced synthetic or real data.

NN Configuration 2	
Nature of Input	Shuffled
Number of Hidden Layers	3
Number of Hidden Neurons	8, 5, 3

(b) NN configuration 2 for reduced synthetic or real data.

Table 5.4: Different NN configurations for reduced synthetic or real data.

6 Results and Discussion

In the previous chapter several cases were analyzed, to explore the capabilities and efficiencies of linear and non-linear ML algorithms when applied to real and synthetic data. This chapter would now focus on explaining the results achieved for the above experiments and compare linear and non-linear algorithm's efficiency. To represent how well the algorithm predicted the OTC values, different error plots are created. The threshold range of retrieval error for ozone total column is between 2% to 3% of an average ozone total column of 300 DU. A histogram of the percentage of relative error for train and test is created by using the formula:

$$\text{relative error} = \frac{\left(OTC_{original} - OTC_{predicted}\right)}{OTC_{original}} \times 100 \tag{6.1}$$

A histogram of absolute error for train and test is created by using the formula:

$$\text{absolute error} = \left(OTC_{original} - OTC_{predicted}\right) \tag{6.2}$$

The percentage of MAE is also calculated for both train and test set using the formula:

$$\text{MAE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{OTC_{original_t} - OTC_{predicted_t}}{OTC_{original_t}} \right| \tag{6.3}$$

where:

$OTC_{original}$ = Value of ozone total column by RTM

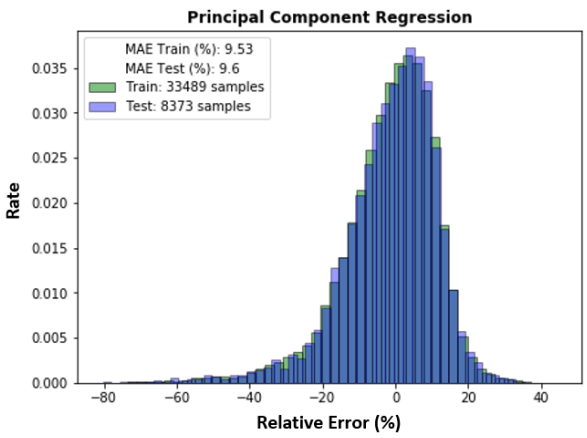
$OTC_{predicted}$ = Value of ozone total column by NN

n = total number of samples

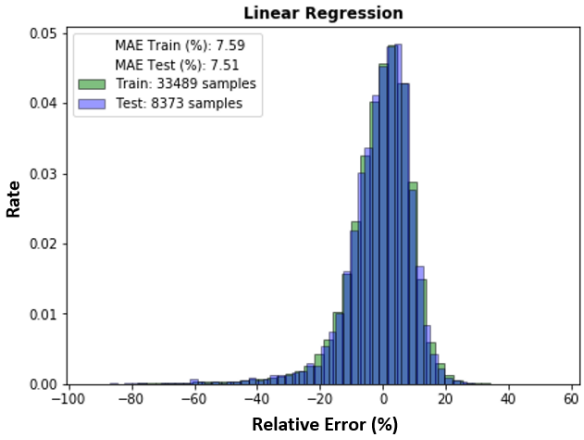
All the histograms are normalized. A third plot for the neural network is created to compare the performance of the NN for the predicted and actual OTC values. For this plot, first 20 samples are taken from the train set and the next 20 samples are taken from the test set. Section 6.1 to 6.5 highlight the results for linear scheme and 6.6 to 6.8 highlight the results for non-linear scheme.

6.1 Linear Regression for Synthetic Dataset: All Clusters

The first experiment is conducted on synthetic data considering all the 11 clusters for the linear scheme with reduced and actual spectra. Figure 6.1 shows the outcome of the two sub approaches carried out for LR. It is clearly seen from 6.1a & 6.1b that LR performs worst on reduced spectra (∼10% MAE i.e. 30 DU) than on the actual spectra (∼8% MAE i.e 24 DU). However, the result from both the cases are greater than the threshold error range (∼6 to 9 DU) for the OTC. Thus, the experiment for single cluster at a time is analyzed.



(a) Histogram of percentage of relative errors of the OTC from synthetic reduced spectra for all clusters.



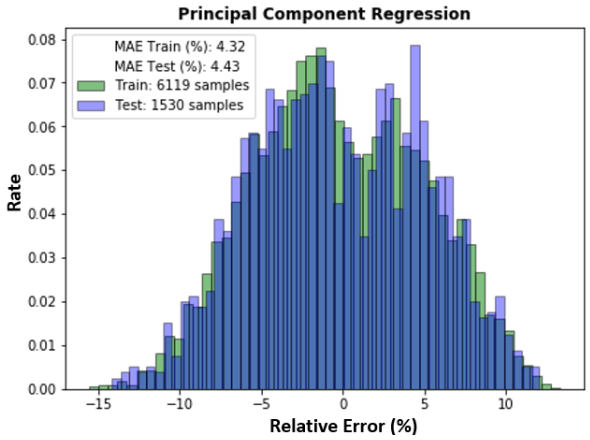
(b) Histogram of percentage of relative errors of the OTC from synthetic actual spectra for all clusters.

Figure 6.1: Linear regression results for synthetic data for all clusters.

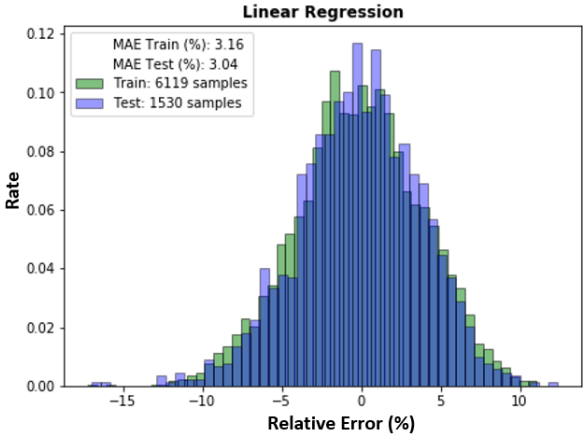
6.2 Linear Regression for Synthetic Dataset: One Cluster

For the experiment to test one cluster at a time, synthetic data for cluster 1 and cluster 10 are tested independently with the regressor. Figure 6.2 shows the outcome for the two LR sub approaches carried out for cluster 1 and 6.3 for cluster 10. This experiment provides very interesting results, as can be seen, the error rate for both the sub approaches (i.e LR & PCR) for cluster 1 & 10 has dropped significantly compared to the previous case where all the clusters are taken into consideration. This shows that the assumption to test for one cluster at a time is true. This also explains that due to large input space and differences in the properties of each cluster the regressor could not be trained well and hence the error is higher when tested on all clusters compared to a single cluster where the regressor trains itself on a smaller input space and for one specific property. Thus, clustering is important for the retrieval of OTC. However, considering this case only, the error rate for the reduced spectra is on the higher side (i.e ∼ 3% to 4% MAE) for this case as well than for the actual

spectra for either cluster. Moreover, the MAE for either cluster 1 or cluster 10 for actual spectra is just close to the higher limit (i.e $\sim$ 3% = 9 DU) of the range but not the best. This leads to further testing of LR on the real data to verify if the results are better or get worst when applied for real-time data.

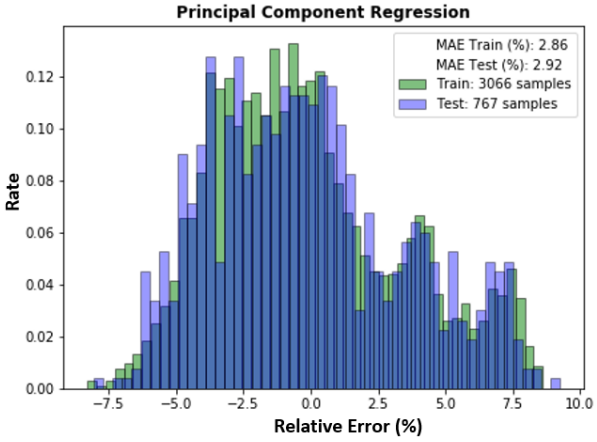


(a) Histogram of percentage of relative errors of the OTC from synthetic reduced spectra for one cluster (cluster 1).

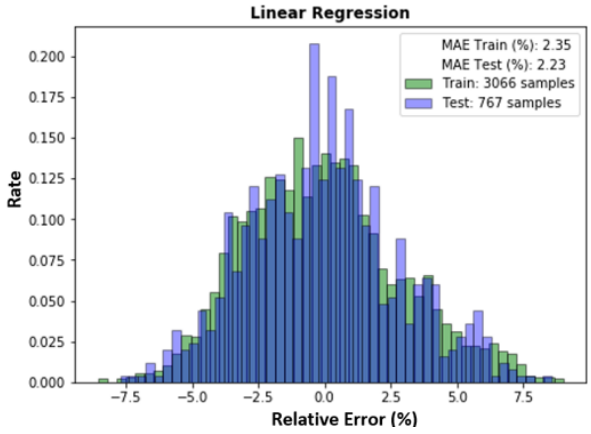


(b) Histogram of percentage of relative errors of the OTC from synthetic actual spectra for one cluster (cluster 1).

Figure 6.2: Linear regression results for synthetic data for one cluster (cluster 1).



(a) Histogram of percentage of relative errors of the OTC from synthetic reduced spectra for one cluster (cluster 10).

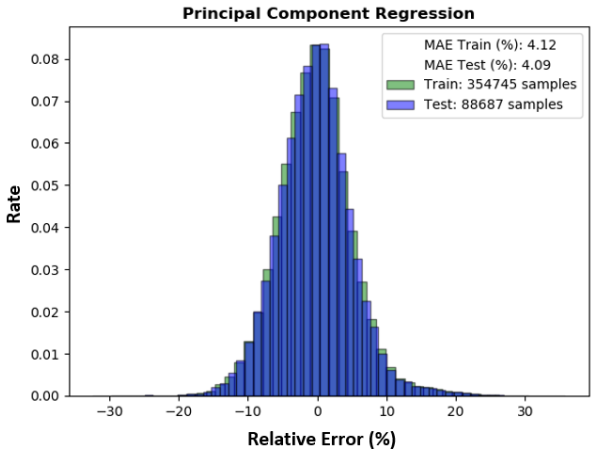


(b) Histogram of percentage of relative errors of the OTC from synthetic actual spectra for one cluster (cluster 10).

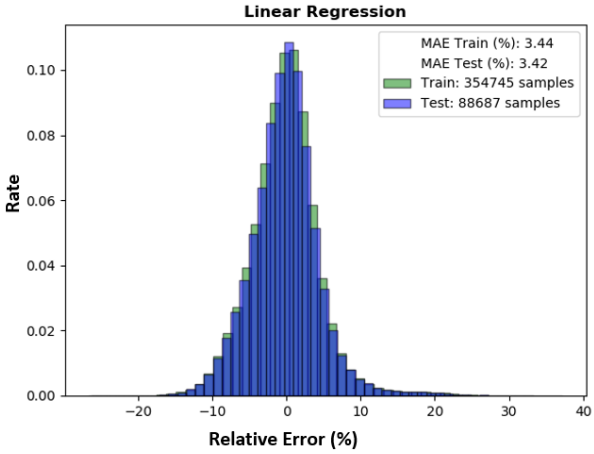
Figure 6.3: Linear regression results for synthetic data for one cluster (cluster 10).

6.3 Linear Regression for Real Dataset

For the experiment on real data, four different days (01/01/18, 04/04/18, 08/08/18 and 12/12/18) are chosen and tested independently for LR. Figure 6.4 to 6.7 shows the outcome for the two LR sub approaches carried out for each day respectively. As seen in these figures the MAE for the actual spectra for either of the days is between 3% to 7% (i.e 9 to 21 DU) which is a huge error compared to the threshold error in terms of real-time application. This shows that when testing a linear scheme on real data it gets worst compared to synthetic data for one cluster. Also, the reduced spectra are tested and it is seen that for this approach as well the results had no improvement. This, lead to further testing by tweaking the values of SA and SZA for the real data and check the behavior of LR.

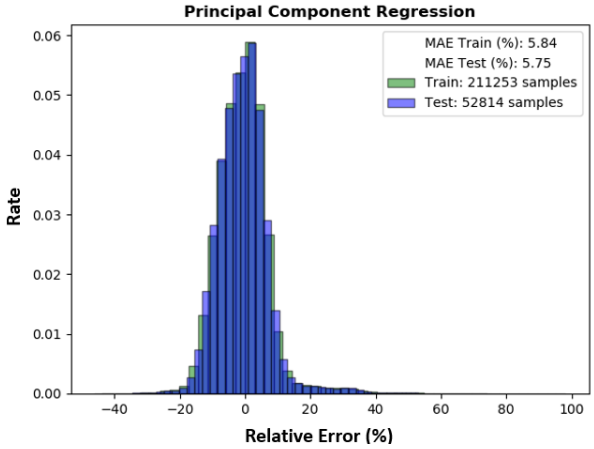


(a) Histogram of percentage of relative errors of the OTC from real reduced spectra for 01/01/18.

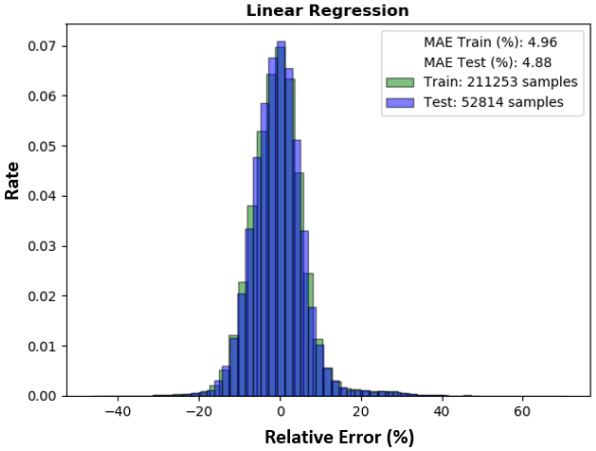


(b) Histogram of percentage of relative errors of the OTC from real actual spectra for 01/01/18.

Figure 6.4: Linear regression results for real data for 01/01/18.

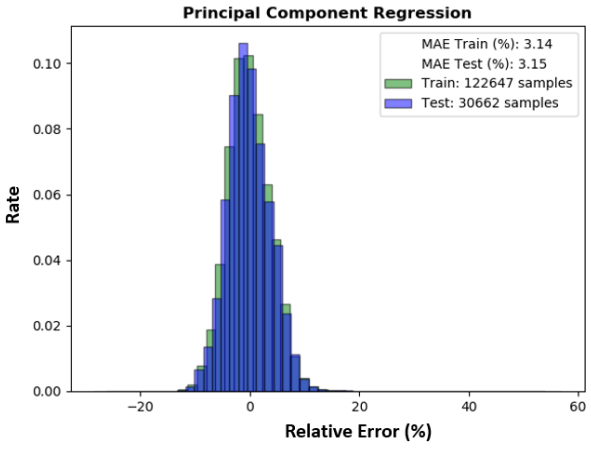


(a) Histogram of percentage of relative errors of the OTC from real reduced spectra for 04/04/18.

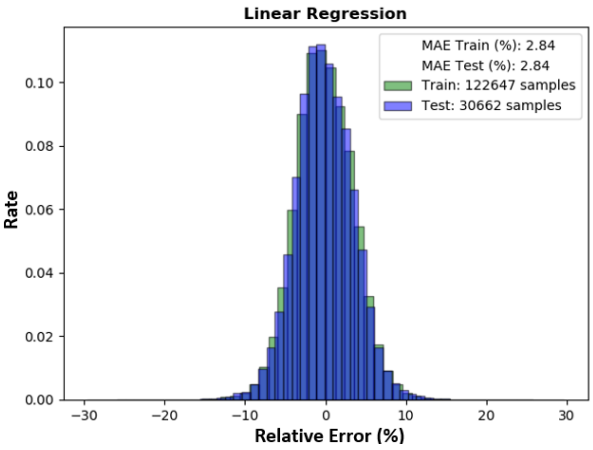


(b) Histogram of percentage of relative errors of the OTC from real actual spectra for 04/04/18.

Figure 6.5: Linear regression results for real data for 04/04/18.

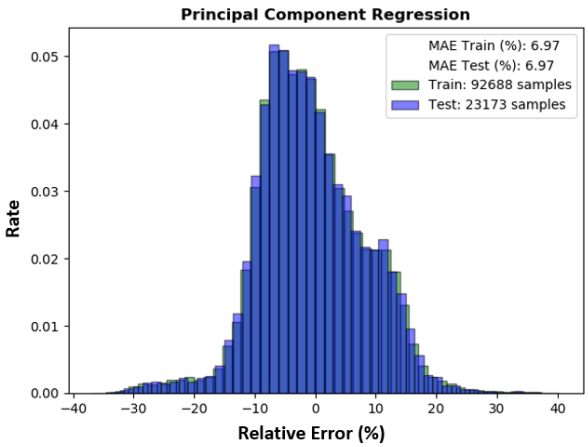


(a) Histogram of percentage of relative errors of the OTC from real reduced spectra for 08/08/18.

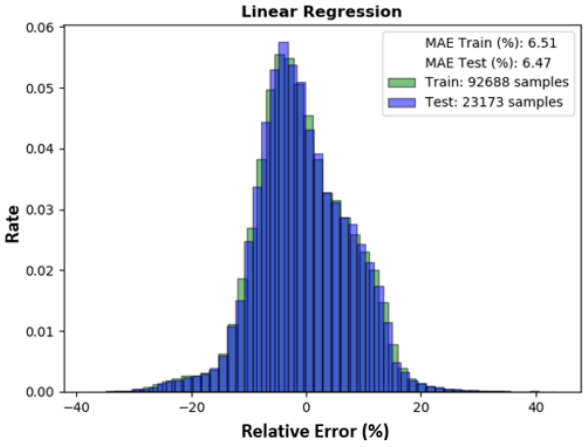


(b) Histogram of percentage of relative errors of the OTC from real actual spectra for 08/08/18.

Figure 6.6: Linear regression results for real data for 08/08/18.



(a) Histogram of percentage of relative errors of the OTC from real reduced spectra for 12/12/18.

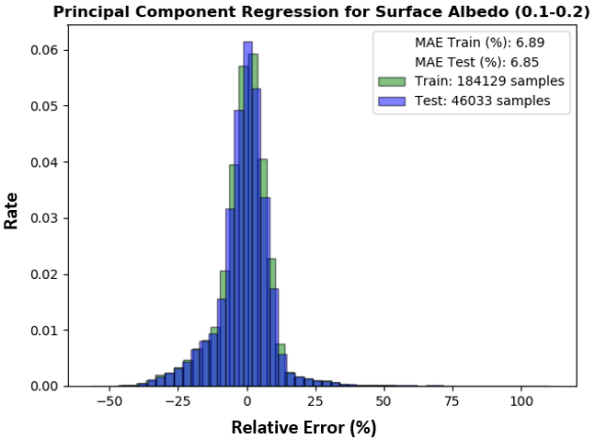


(b) Histogram of percentage of relative errors of the OTC from real actual spectra for 12/12/18.

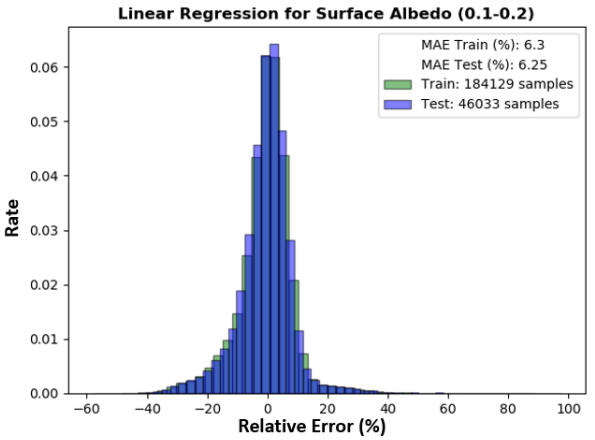
Figure 6.7: Linear regression results for real data for 12/12/18.

6.4 Linear Regression for Real Dataset: Albedo = (0.1 to 0.2)

Since the change in albedo affects the values of OTC directly this case is implemented to test if the SA is restricted to 0.1 to 0.2 instead of the default values (0.2-0.4) will there be any significant improvement in the results of LR. For this experiment, the three days are tested independently. The results for both the sub approaches for LR are shown in Figure 6.8 to 6.10 for 04/04/18, 08/08/18 and 12/12/18 respectively. It is seen from the results of actual spectra and reduced spectra for all the three days that with the decrease in SA values the MAE increases largely (i.e ranges between $\sim$ 9 to 21 DU). Thus, with this experiment as well there is no improvement in prediction error for retrieval of OTC for the real data.

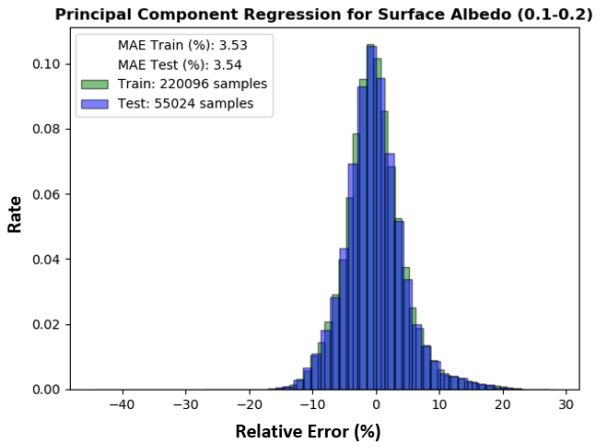


(a) Histogram of percentage of relative errors of the OTC from real reduced spectra with albedo (0.1-0.2) for 04/04/18.

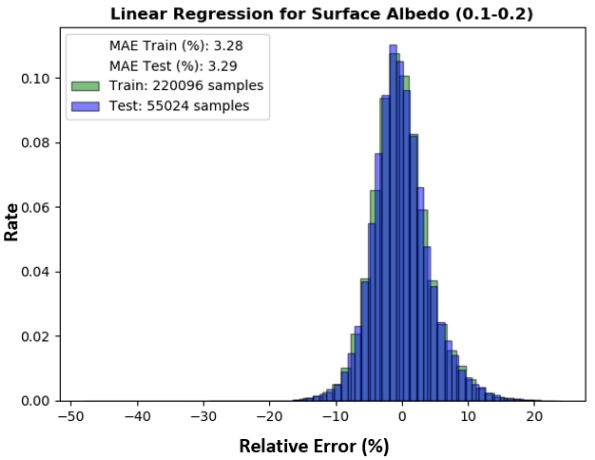


(b) Histogram of percentage of relative errors of the OTC from real actual spectra with albedo (0.1-0.2) for 04/04/18.

Figure 6.8: Linear regression results for real data with SA (0.1-0.2) for 04/04/18.

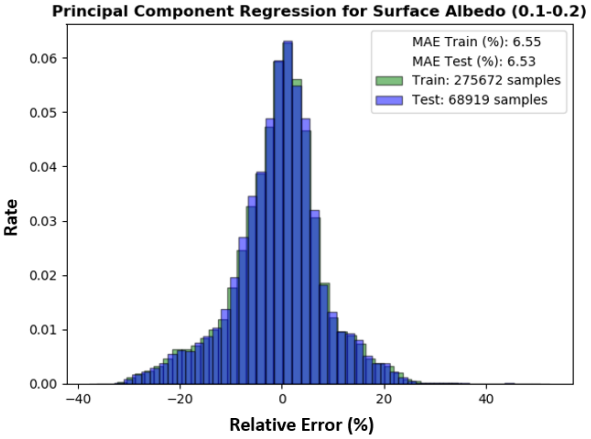


(a) Histogram of percentage of relative errors of the OTC from real reduced spectra with albedo (0.1-0.2) for 08/08/18.

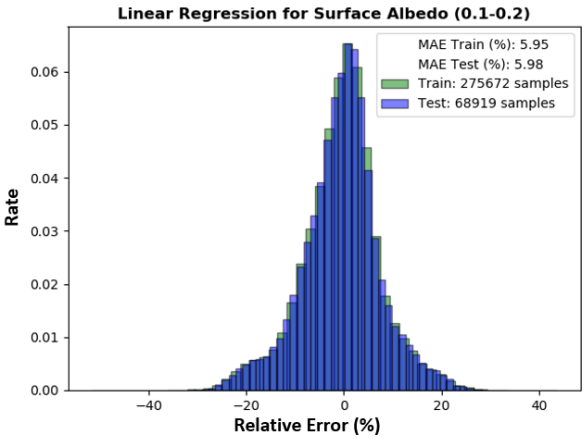


(b) Histogram of percentage of relative errors of the OTC from real actual spectra with albedo (0.1-0.2) for 08/08/18.

Figure 6.9: Linear regression results for real data with SA (0.1-0.2) for 08/08/18.



(a) Histogram of percentage of relative errors of the OTC from real reduced spectra with albedo (0.1-0.2) for 12/12/18.

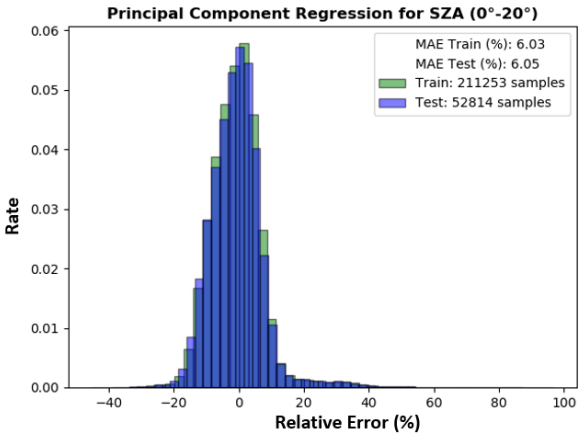


(b) Histogram of percentage of relative errors of the OTC from real actual spectra with albedo (0.1-0.2) for 12/12/18.

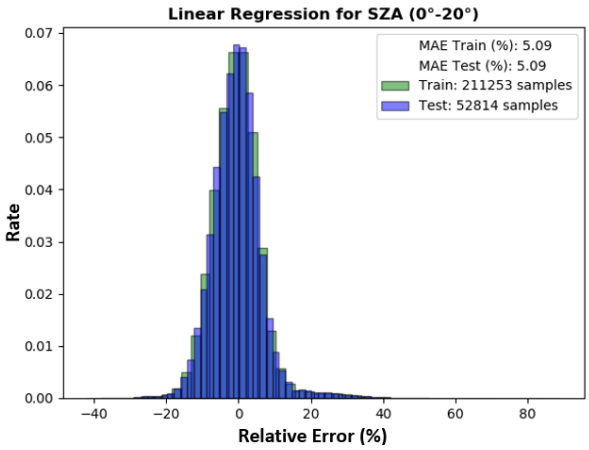
Figure 6.10: Linear regression results for real data with SA (0.1-0.2) for 12/12/18.

6.5 Linear Regression for Real Dataset: Solar Zenith Angle = (0° to 20°)

The change in SZA also affects the values of OTC directly, for this case SZA is restricted to 0° to 20° for all three days. As seen from the histograms of reduced and actual spectra in Figure 6.11 to 6.13 the errors are increased up to 2% compared to the default test in 6.3. Thus, this emphasizes that after testing all the different configurations of the real dataset as well as the synthetic dataset the percentage of error from LR to predict OTC is not improved significantly. The best prediction achieved so far for real or synthetic data is with a MAE error of $\sim$ 3% i.e 9 DU. This leads to the understanding that the datasets are not linearly related and further testing of the datasets with a non-linear scheme could provide better results according to universal approximation theorem as explained in 4.4.1.

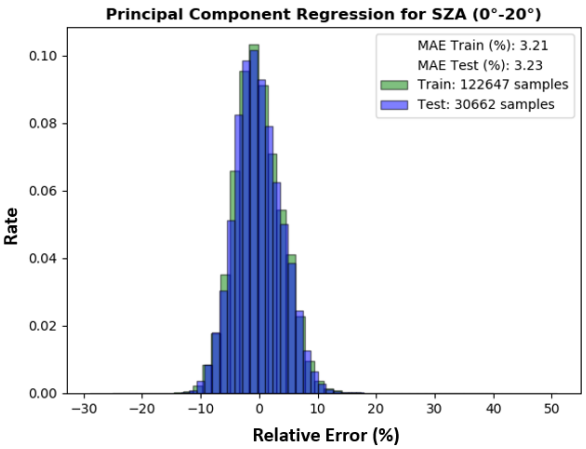


(a) Histogram of percentage of relative errors of the OTC from real reduced spectra with SZA (0° to 20°) for 04/04/18.

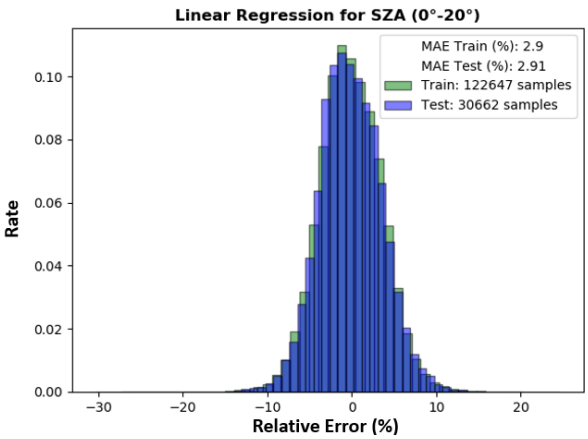


(b) Histogram of percentage of relative errors of the OTC from real actual spectra with SZA (0° to 20°) for 04/04/18.

Figure 6.11: Linear regression results for real data with SZA (0° to 20°) for 04/04/18.

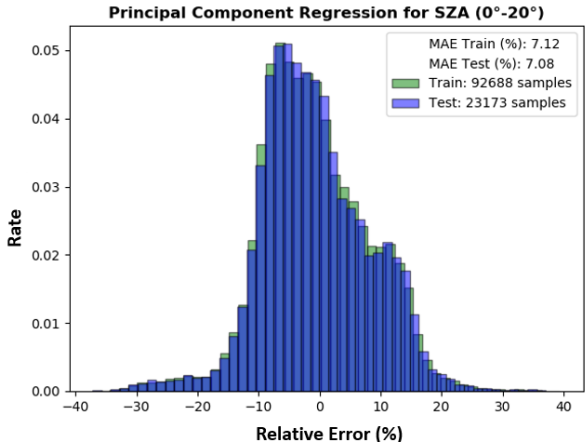


(a) Histogram of percentage of relative errors of the OTC from real reduced spectra with SZA (0° to 20°) for 08/08/18.

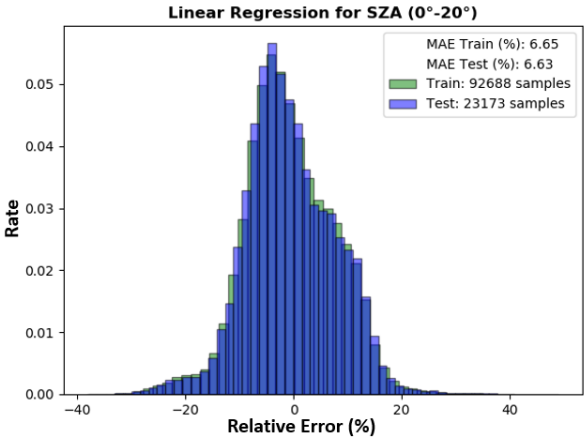


(b) Histogram of percentage of relative errors of the OTC from real actual spectra with SZA (0° to 20°) for 08/08/18.

Figure 6.12: Linear regression results for real data with SZA (0° to 20°) for 08/08/18.



(a) Histogram of percentage of relative errors of the OTC from real reduced spectra with SZA (0° to 20°) for 12/12/18.



(b) Histogram of percentage of relative errors of the OTC from real actual spectra with SZA (0° to 20°) for 12/12/18.

Figure 6.13: Linear regression results for real data with SZA (0° to 20°) for 12/12/18.

6.6 Neural Network for Synthetic Dataset: One Cluster

For the non-linear scheme, both synthetic and real datasets along with their reduced spectra are passed to different NN configurations. First, the synthetic data for cluster 10 is tested for reduced and actual spectra. Figure 6.14 shows the result for reduced spectra and 6.15 to 6.19 shows the results for all the 5 different NN configurations tested for the actual spectra. It is seen that when the reduced data is passed to the NN with [8, 5, 3] hidden neurons the error for this approach is slightly lower compared to its counterpart when passed to LR (see Fig. 6.3a) Through this approach the MAE achieved is $\sim$ 2% i.e 6 DU. The actual data is also passed to the NN with 3 hidden layers and [10, 5, 3] neurons respectively. As seen in Figure 6.15 the relative error for train set has reduced considerably and the MAE is reduced to 0.26% i.e 0.78 DU. However, for unseen test data, the error is very high (i.e $\sim$ 108 DU). The actual spectra is then further tested with hidden neurons [20, 10, 5] (i.e NN configuration 2) and it is observed in Figure 6.16 that with increase in the number of hidden neurons from [10, 5, 3] to [20, 10 5] their is significant drop in errors for both train (MAE = 7e-06%) and test (MAE = 7%) set however, the error for unseen test data is still on a higher range. This result gives an understanding that the network is over-fitting on the train data. This is illustrated through the performance of the network by using a comparison plot for train and test set as shown in Figure 6.15c. In this graph, the predicted and original OTC values of these sets are plotted together. Through this plot, it is clear that the network is predicting very well during training but during testing, there is a discontinuity in the plot due to over-fitting. To overcome this problem the dataset is shuffled, then divided into train and test set. This is then retested with NN configuration with [10, 5, 3] hidden neurons. Figure 6.17 illustrates the results for this configuration. It is observed that for this configuration the errors have

dropped significantly for the test set as well (i.e MAE $\sim$ 0.6%) and the performance of the network (see 6.17c) is also much better compared to previous two configurations. However, the difference between the predicted and original OTC is still quite large. Thus, two other configurations were tested, first is where the default training algorithm is changed to *trainscg* (NN configuration 4) and hidden neurons to [20, 10, 5] and second where only the number of hidden neurons is changed to [20, 10, 5] with default training algorithm (NN configuration 5). Figure 6.18 and 6.19 illustrates results for these two configuration. It is observed that for the configuration where trainscg is the training algorithm the prediction for both train and test set is much better and quite close to the actual values (i.e MAE between 0.09% to 0.1%, 0.27 to 0.3 DU). However, for the last configuration, the predicted and original values fit in completely together with MAE of 5e-06% (i.e 0.15e-04 DU) for unseen test data. Through this experiment few conclusions are drawn, shuffling of data is very important to achieve accurate results. The NN configuration 5 with trainlm provides results with high precision compared to NN configuration 4 with the trainscg training algorithm. Thus trainlm seems to be the optimal choice of training algorithm for this problem. Also, error with high precision is preferred for the retrieval of OTC since the concentration of ozone is very small as explained in Section 1.1.1 and due to high signal to noise ratio of tropomi (see 3.1.1).

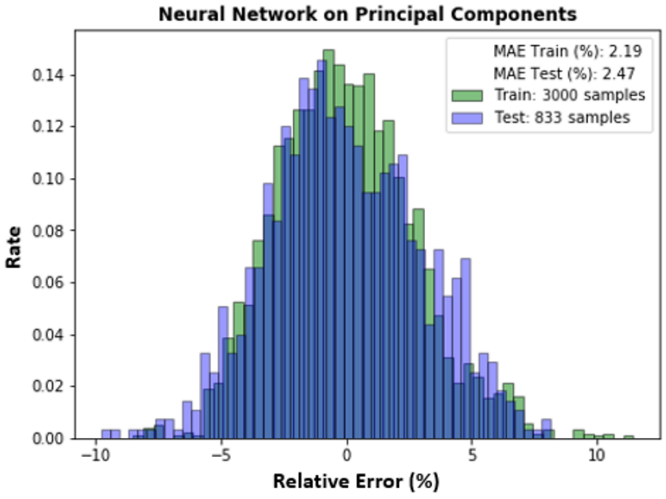
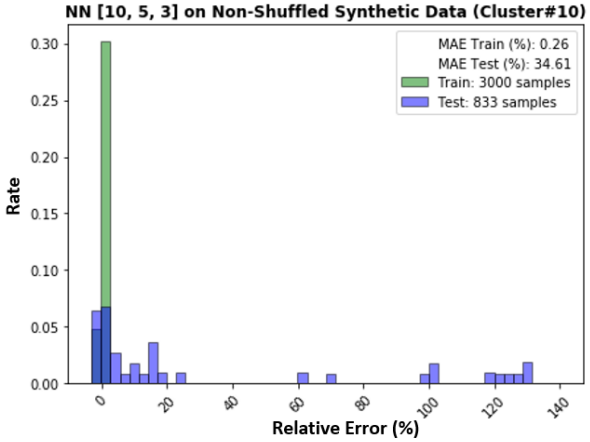
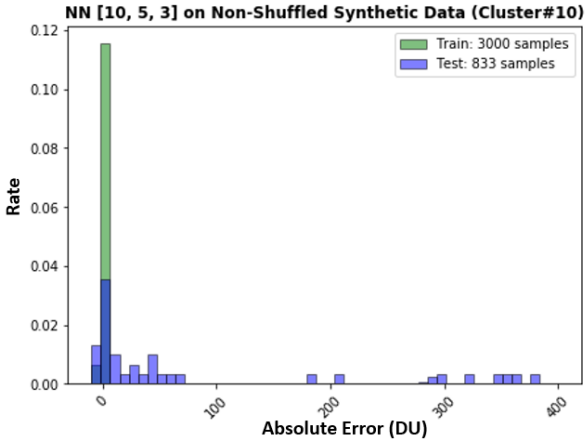


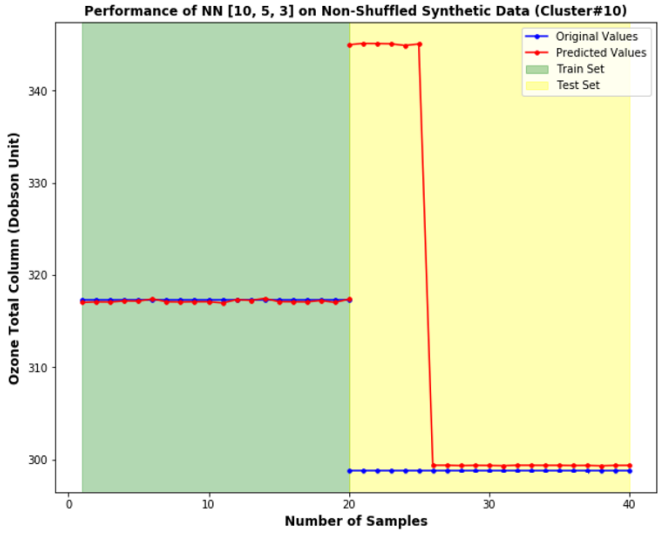
Figure 6.14: Histogram of percentage of relative errors of the OTC for NN with reduced synthetic spectra for one cluster (cluster 10).



(a) Histogram of percentage of relative errors of the OTC for synthetic data cluster 10 using NN configuration 1.

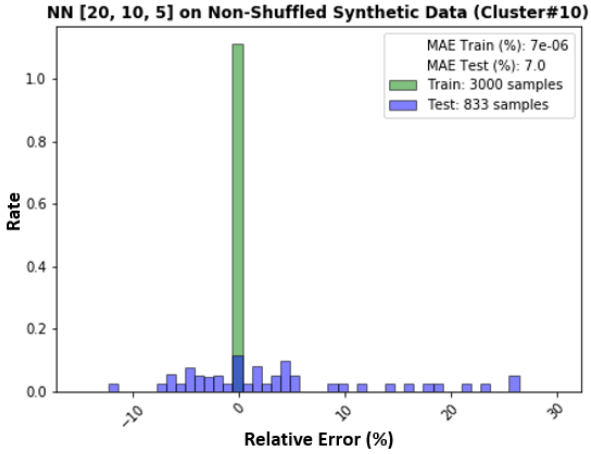


(b) Histogram of absolute error in DU of the OTC for synthetic data cluster 10 using NN configuration 1.

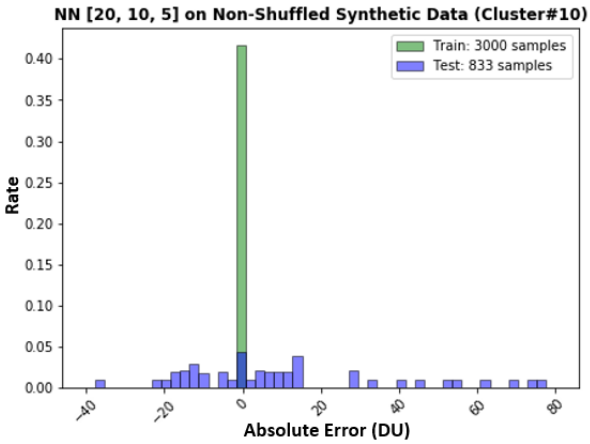


(c) Comparison plot for predicted and actual OTC values for synthetic data cluster 10 using NN configuration 1.

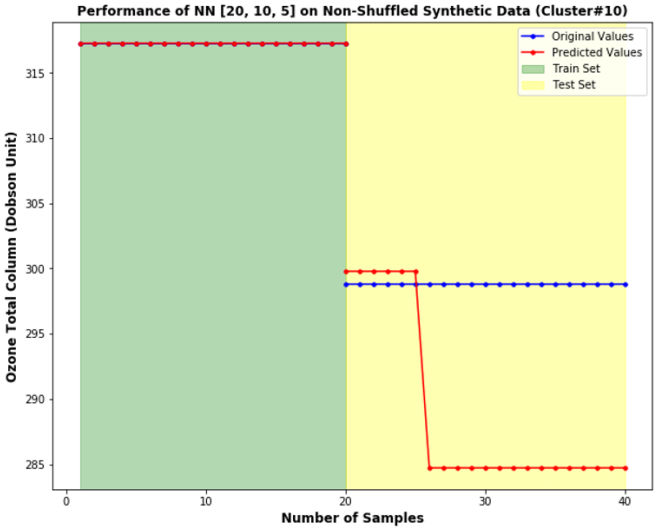
Figure 6.15: Results for synthetic data cluster 10 using NN configuration 1.



(a) Histogram of percentage of relative errors of the OTC for synthetic data cluster 10 using NN configuration 2.

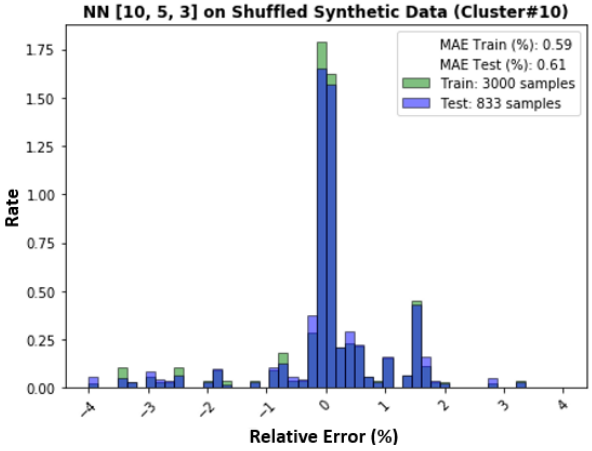


(b) Histogram of absolute error in DU of the OTC for synthetic data cluster 10 using NN configuration 2.

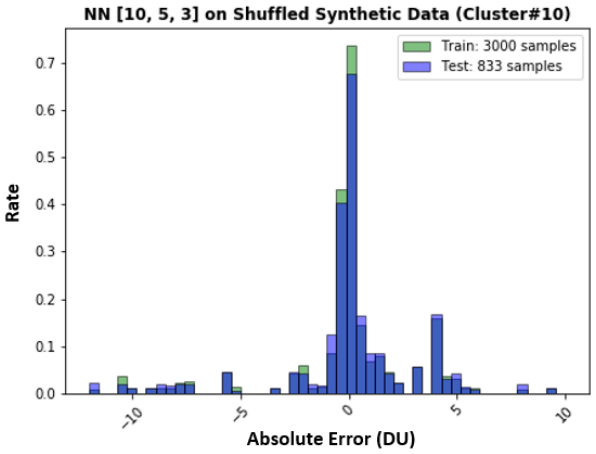


(c) Comparison plot for predicted and actual OTC values for synthetic data cluster 10 using NN configuration 2.

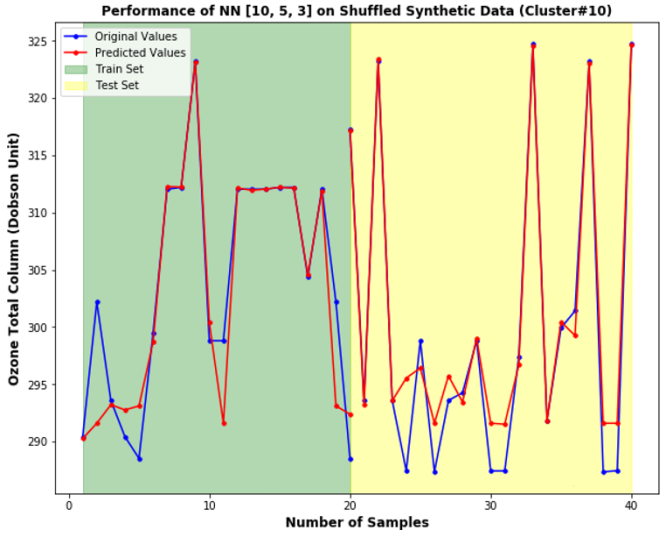
Figure 6.16: Results for synthetic data cluster 10 using NN configuration 2.



(a) Histogram of percentage of relative errors of the OTC for synthetic data cluster 10 using NN configuration 3.

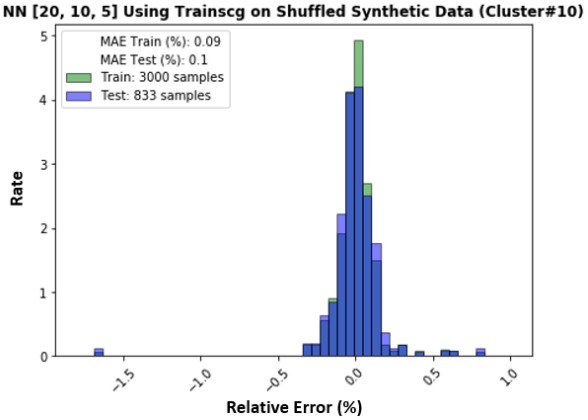


(b) Histogram of absolute error in DU of the OTC for synthetic data cluster 10 using NN configuration 3.

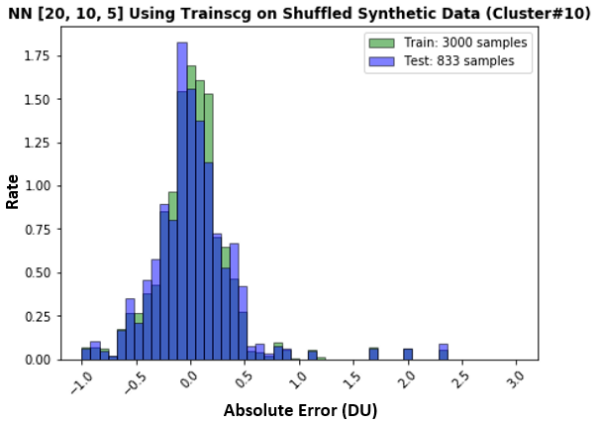


(c) Comparison plot for predicted and actual OTC values for synthetic data cluster 10 using NN configuration 3.

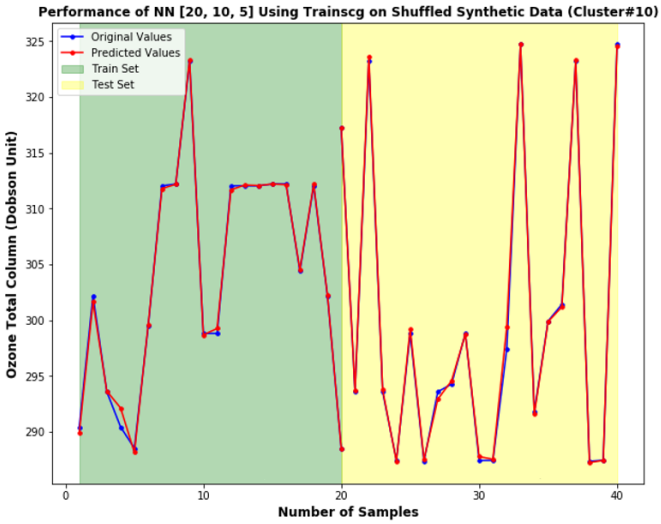
Figure 6.17: Results for synthetic data cluster 10 using NN configuration 3.



(a) Histogram of percentage of relative errors of the OTC for synthetic data cluster 10 using NN configuration 4.

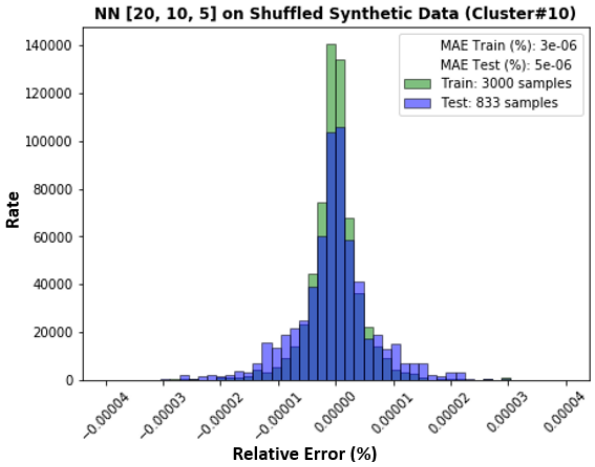


(b) Histogram of absolute error in DU of the OTC for synthetic data cluster 10 using NN configuration 4.

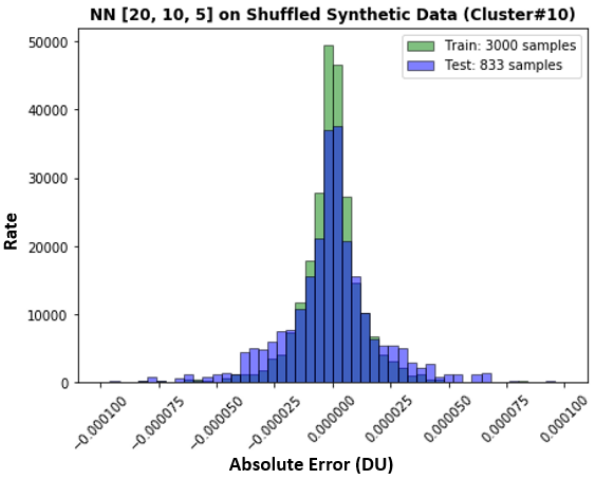


(c) Comparison plot for predicted and actual OTC values for synthetic data cluster 10 using NN configuration 4.

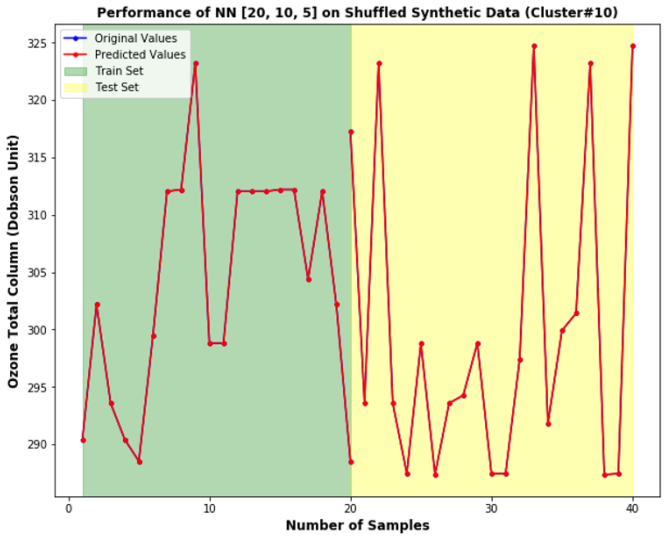
Figure 6.18: Results for synthetic data cluster 10 using NN configuration 4.



(a) Histogram of percentage of relative errors of the OTC for synthetic data cluster 10 using NN configuration 5.



(b) Histogram of absolute error in DU of the OTC for synthetic data cluster 10 using NN configuration 5.



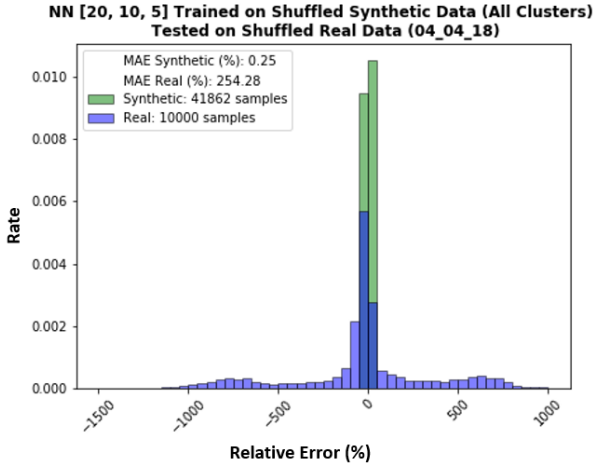
(c) Comparison plot for predicted and actual OTC values for synthetic data cluster 10 using NN configuration 5.

Figure 6.19: Results for synthetic data cluster 10 using NN configuration 5.

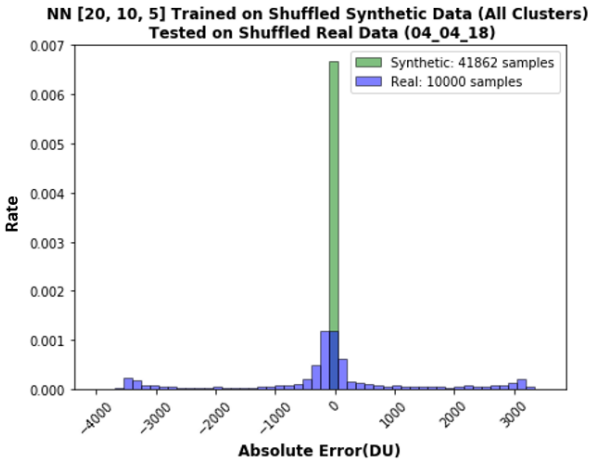
6.7 Neural Network for Synthetic Dataset: All Clusters

On achieving exceptional results for NN trained and tested on synthetic data for one cluster. It now makes sense to check the efficiency of NN when applied to a real dataset. Thus, the NN is trained on synthetic data for all clusters and tested on real data for one day. Here, as well 4 different configurations are tested and the input data is always shuffled. The first two

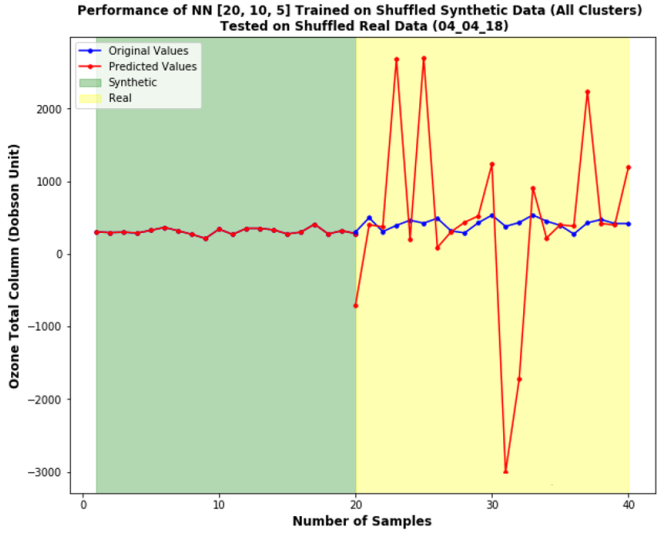
configurations are with 3 hidden layers consisting of [20, 10, 5] neurons (NN configuration 1) and [40, 12, 3] neurons (NN configuration 2) respectively. Figure 6.20 & 6.21 illustrates the results for the above two configurations. On analyzing the results it is seen that the MAE is too large from 254% (i.e $\sim$ 762 DU) & 484% (i.e $\sim$ 1452 DU) respectively when the trained network is tested on real data. This result is quite explanatory as both the datasets vary in terms of spectral resolution, thus to test the network trained on synthetic data the real data is interpolated which induces a great amount of noise. Moreover, the synthetic data does not account for measurement noise during training. To reduce the effect of noise, the approach is to consider the PCs instead of the actual spectra for synthetic and real data for training and testing the network. The efficiency of this approach is illustrated in Figure 6.22 & 6.23. Here, as well two NN configurations are tested one with [20, 10, 5] hidden neurons and another with [8, 5, 3] hidden neurons. As observed in Figure 6.23 the MAE for the configuration with [8, 5, 3] hidden neurons has a significant drop from 250% (i.e $\sim$ 762 DU) to 89.9% (i.e $\sim$ 269 DU). However, the error for prediction of NN for OTC is still very large as noise and other factors (e.g degradation of the sensor, actual aerosol parameters, etc.) are not included in the RTM and hence the real data does not belong to the range of RTM [11]. Thus, the idea was to train the NN directly on an already processed real dataset. The results for this are explained in next section.



(a) Histogram of percentage of relative errors of the OTC trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 1.

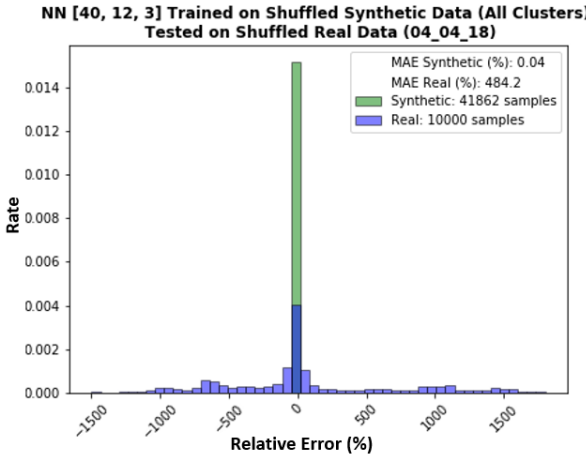


(b) Histogram of absolute error in DU of the OTC trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 1.

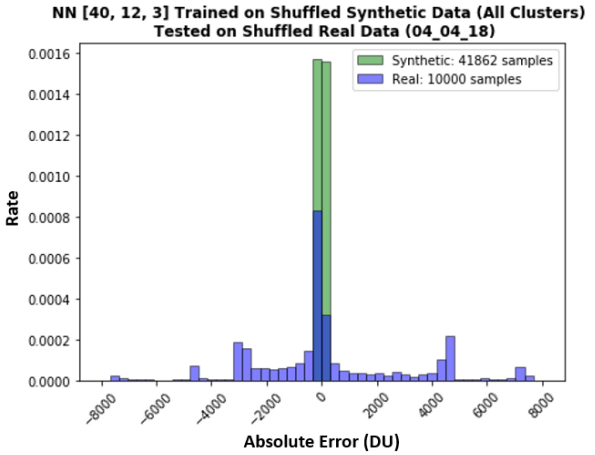


(c) Comparison plot for predicted and actual OTC values trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 1.

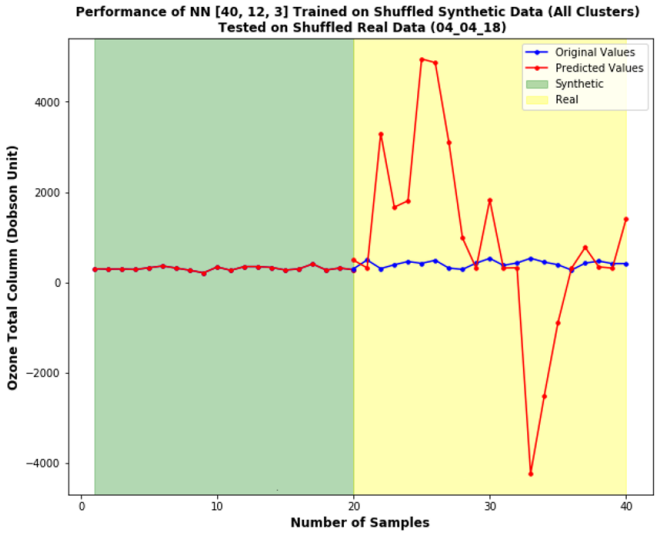
Figure 6.20: Results for NN trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 1.



(a) Histogram of percentage of relative errors of the OTC trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 2.

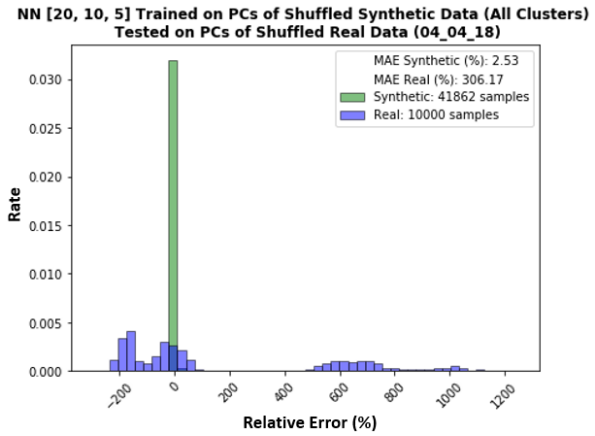


(b) Histogram of absolute error in DU of the OTC trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 2.

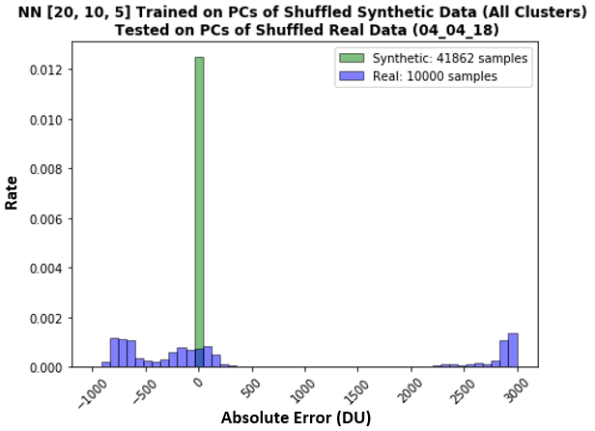


(c) Comparison plot for predicted and actual OTC values trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 2.

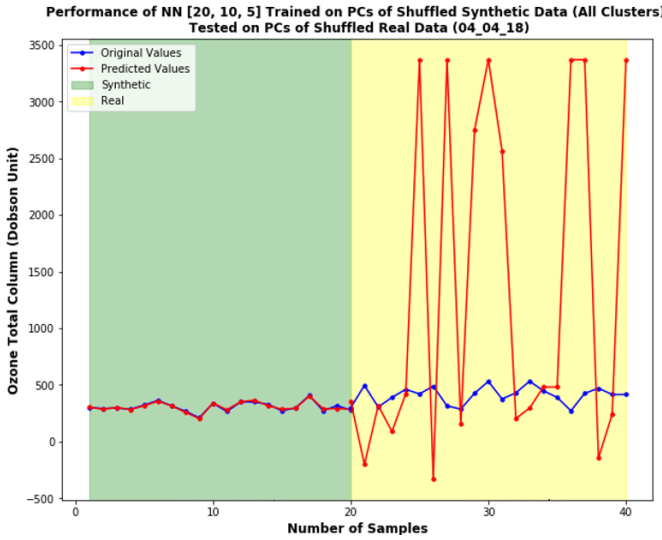
Figure 6.21: Results for NN trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 2.



(a) Histogram of percentage of relative errors of the OTC trained on reduced synthetic data (all clusters) and tested on reduced real data (04/04/18) using NN configuration 1.

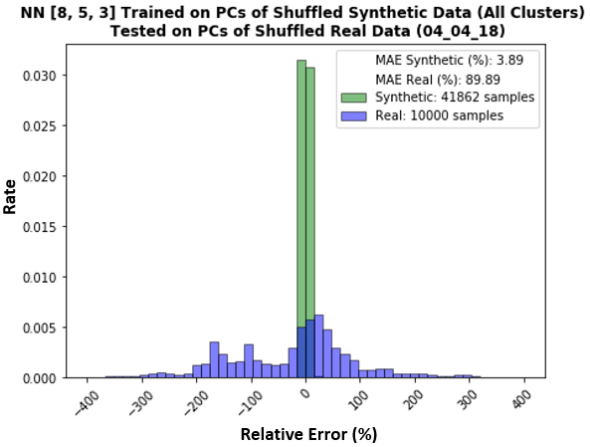


(b) Histogram of absolute error in DU of the OTC trained on reduced synthetic data (all clusters) and tested on reduced real data (04/04/18) using NN configuration 1.

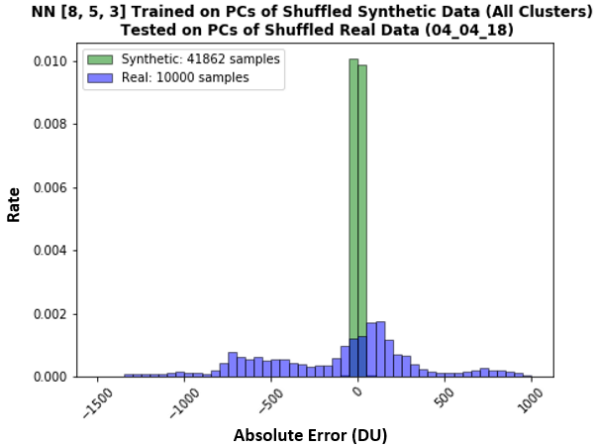


(c) Comparison plot for predicted and actual OTC values trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 1.

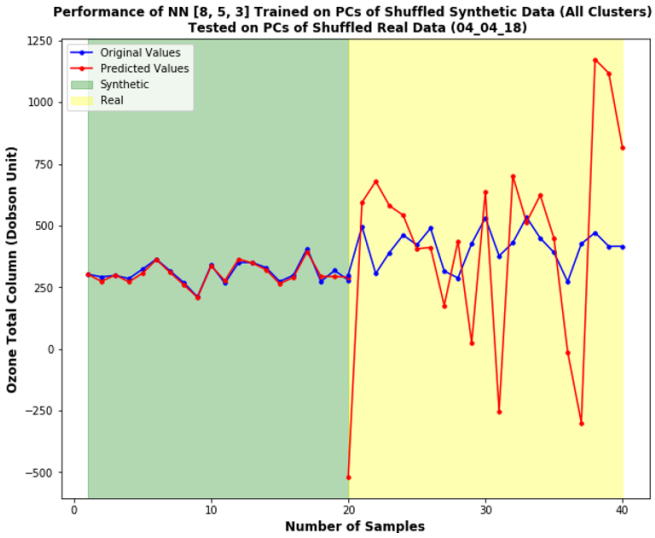
Figure 6.22: Results for NN trained on reduced synthetic data (all clusters) and tested on reduced real data (04/04/18) using NN configuration 1.



(a) Histogram of percentage of relative errors of the OTC trained on reduced synthetic data (all clusters) and tested on reduced real data (04/04/18) using NN configuration 2.



(b) Histogram of absolute error in DU of the OTC trained on reduced synthetic data (all clusters) and tested on reduced real data (04/04/18) using NN configuration 2.



(c) Comparison plot for predicted and actual OTC values trained on synthetic data (all clusters) and tested on real data (04/04/18) using NN configuration 2.

Figure 6.23: Results for NN trained on reduced synthetic data (all clusters) and tested on reduced real data (04/04/18) using NN configuration 2.

6.8 Neural Network for Real Dataset

Since the synthetic and real dataset is different, the idea is to train & test the NN on the real data itself. For this experiment 4 different configurations of NN are tested and it's reduced spectra is also tested. Here, the real data of 04/04/18 is chosen and is shuffled. Figure 6.24 shows the result for reduced spectra when applied to NN. Figure 6.25 to 6.27 represents

the results for the actual spectra when applied to NN. For the PCNN with [8, 5, 3] hidden neurons, it is observed that the MAE is much higher (∼12 DU for train set, ∼18 DU for test set) and hence reducing the spectra is not a good option. For the NN results, with actual spectra of real data the errors for configuration 1 with [20, 10, 5] hidden neurons is found to be quite low (∼0.8% MAE). Moreover, on testing the second configuration the results are much better and the lowest error of 0.7% is obtained which is just 2 DU. On analyzing the performance of the network for predicted and original values in Figure 6.26c it is observed that there is less offset from original value than compared to the previous configuration 1. Two more tests are implemented to check if the results can be improved further, first NN configuration 3 using the trainscg training algorithm for [40, 12, 3] hidden neurons and second NN configuration 4 with 4 hidden layers with [40, 15, 8, 3] neurons respectively. However, as seen in 6.27a the error increases when using the trainscg training algorithm. Also, for the last configuration with 4 hidden layers, the results as seen in Figure 6.28a has no significant improvement in the error. Thus, the best result achieved for the real dataset is with NN configuration 2 with 3 hidden layers with [40, 12, 3] neurons. The MAE is of 0.7% = 2 DU for real dataset and the total time taken is ∼30 minutes. This trained network on the real data (04/04/18) is also tested on real data (08/08/18) to check for the stability of this network. Figure 6.29 illustrates this case. It is seen, as expected that the MAE increases up to 4% when tested on a different day which is understood as the values of input parameters especially, the albedos are dependent on the season. A slight change in weather affects albedo value and OTC. Thus it makes sense to train the network for a specific case. This shows that the trained network is very stable, fast and accurate and can be used in conjunction with the conventional approach. 1.2.

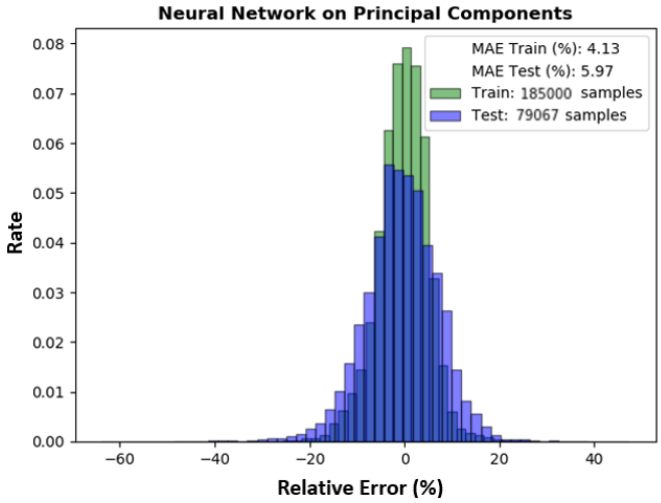
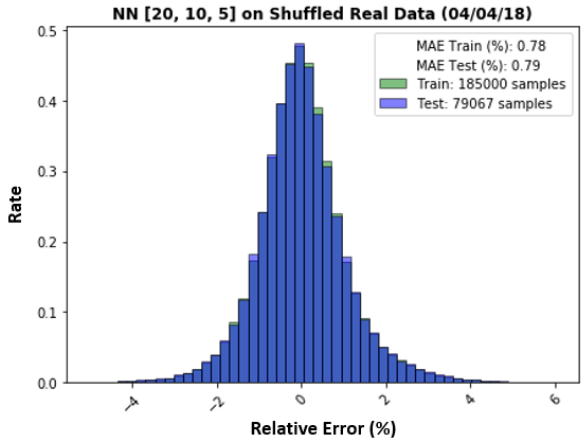
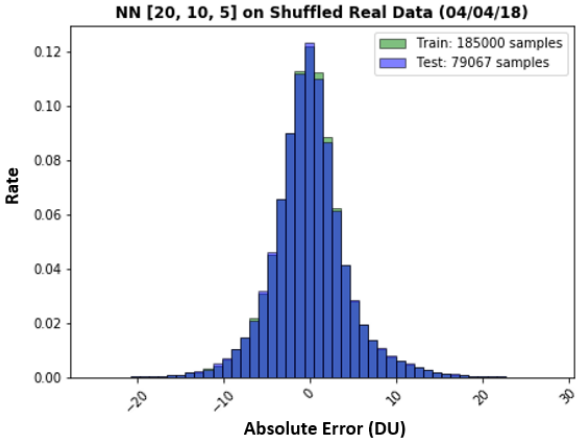


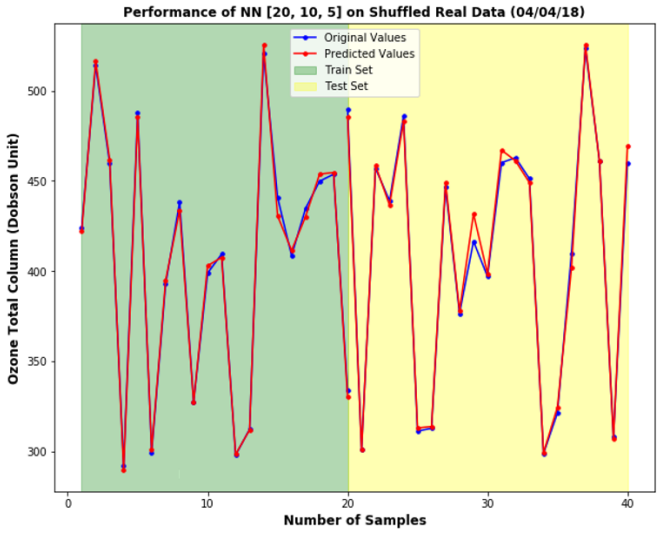
Figure 6.24: Histogram of percentage of relative errors of the OTC for NN with reduced real spectra (04/04/18).



(a) Histogram of percentage of relative errors of the OTC for real data (04/04/18) using NN configuration 1.

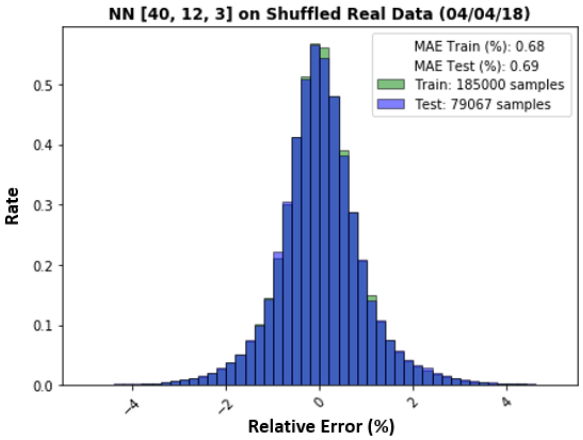


(b) Histogram of absolute error in DU of the OTC for real data (04/04/18) using NN configuration 1.

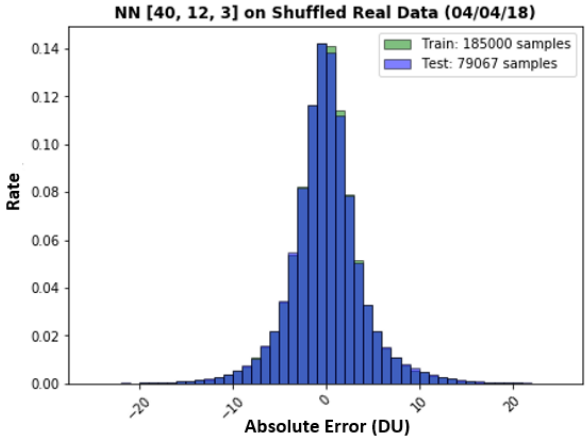


(c) Comparison plot for predicted and actual OTC values for real data (04/04/18) using NN configuration 1.

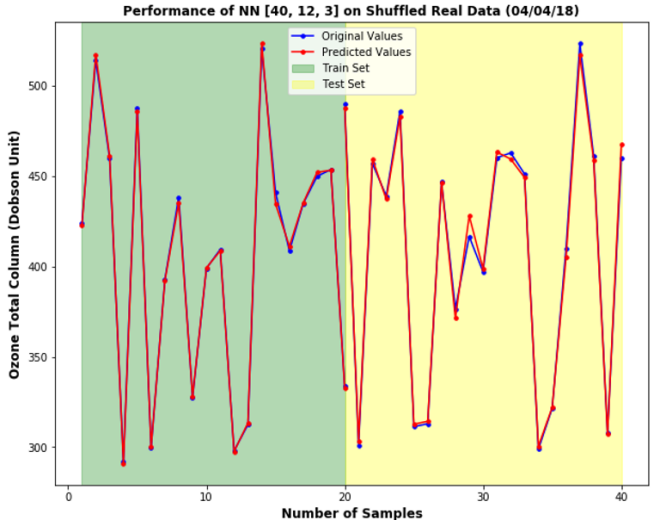
Figure 6.25: Results for real data (04/04/18) using NN configuration 1.



(a) Histogram of percentage of relative errors of the OTC for real data (04/04/18) using NN configuration 2.

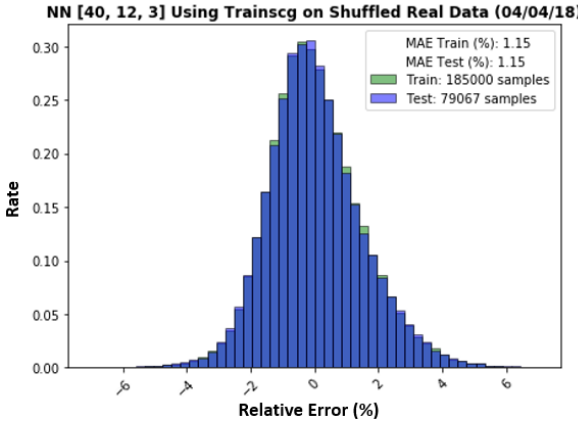


(b) Histogram of absolute error in DU of the OTC for real data (04/04/18) using NN configuration 2.

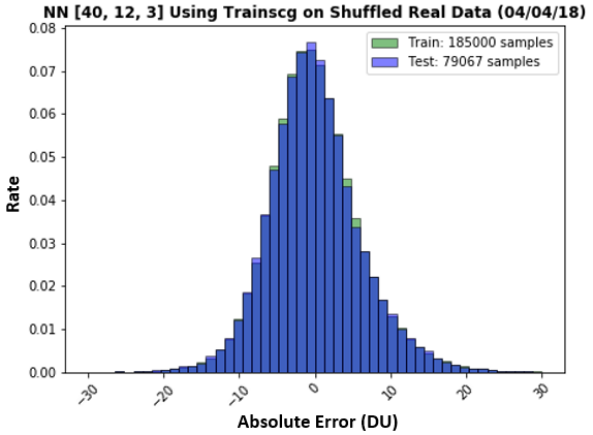


(c) Comparison plot for predicted and actual OTC values for real data (04/04/18) using NN configuration 2.

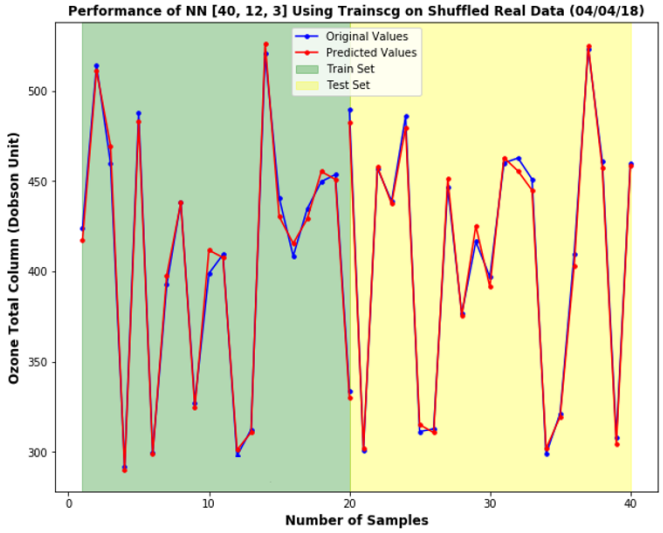
Figure 6.26: Results for real data (04/04/18) using NN configuration 2.



(a) Histogram of percentage of relative errors of the OTC for real data (04/04/18) using NN configuration 3.

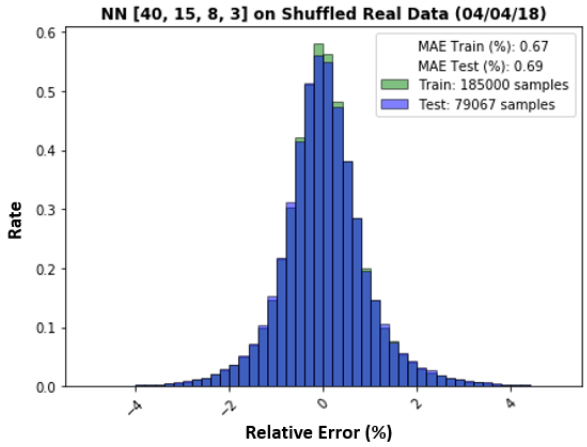


(b) Histogram of absolute error in DU of the OTC for real data (04/04/18) using NN configuration 3.

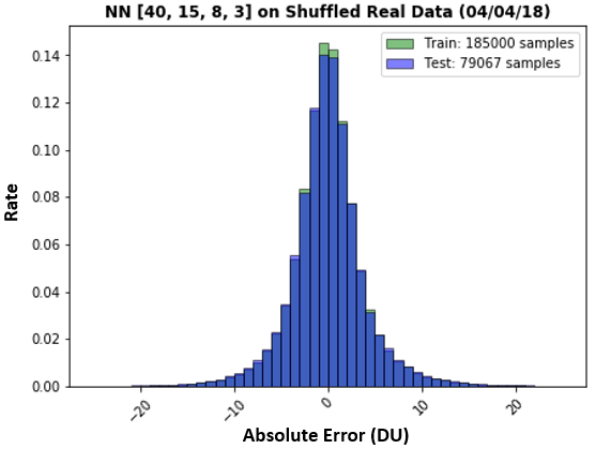


(c) Comparison plot for predicted and actual OTC values for real data (04/04/18) using NN configuration 3.

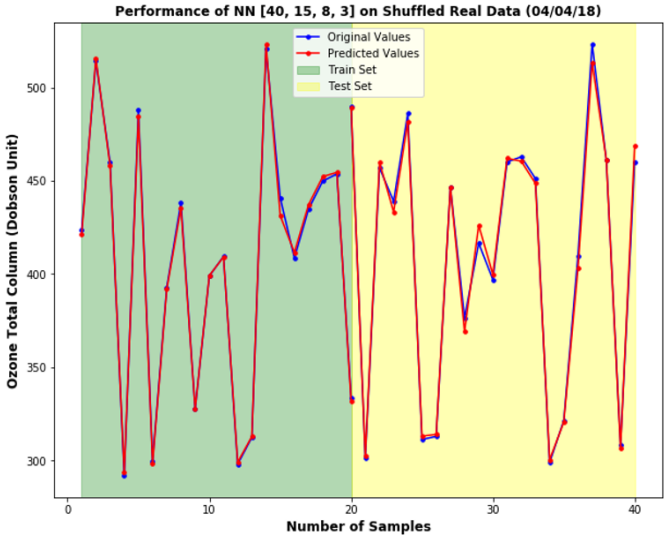
Figure 6.27: Results for real data (04/04/18) using NN configuration 3.



(a) Histogram of percentage of relative errors of the OTC for real data (04/04/18) using NN configuration 3.

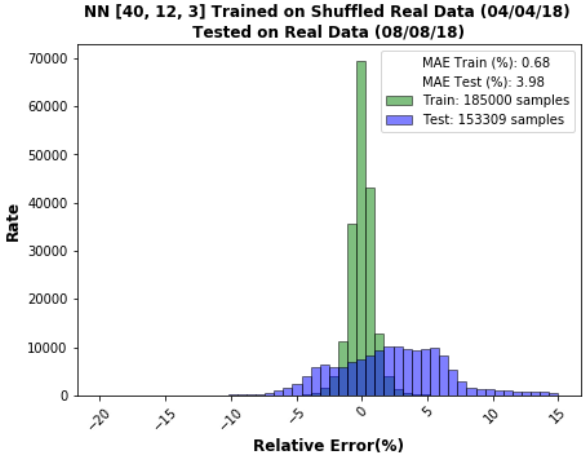


(b) Histogram of absolute error in DU of the OTC for real data (04/04/18) using NN configuration 3.

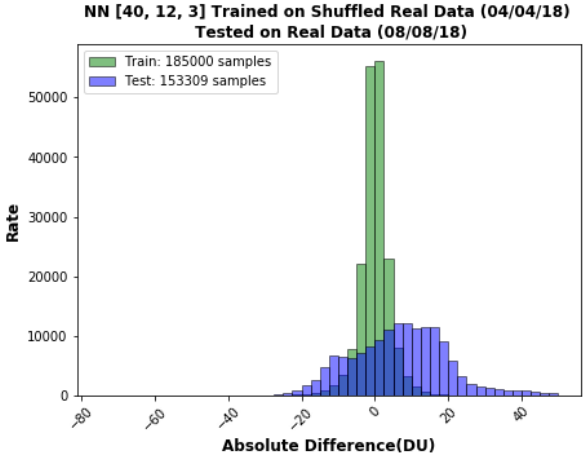


(c) Comparison plot for predicted and actual OTC values for real data (04/04/18) using NN configuration 3.

Figure 6.28: Results for real data (04/04/18) using NN configuration 4.



(a) Histogram of percentage of relative errors of the OTC for real data (04/04/18) tested on (08/08/18) using NN configuration 2.



(b) Histogram of absolute errors of the OTC for real data (04/04/18) tested on (08/08/18) using NN configuration 2.

Figure 6.29: Results for NN trained on real data (04/04/18) and tested on real data (08/08/18) using NN configuration 2.

7 Conclusion

7.1 Summary

The main contribution of this thesis was the investigation of the potential of machine learning algorithms for ozone total column retrieval from the newest sensor TROPOMI onboard S5P. The threshold range of retrieval error for ozone total column is between 2% to 3% of an average ozone total column of 300 DU. In this work, the efficiency of linear and non-linear ML schemes, as well as dimensionality reduction techniques for real and synthetic measurements of ozone total column, were tested. For this, several scenarios were designed w.r.t synthetic and real measurements.

It was found that when linear regression was applied to the synthetic measurements consisting of all the 11 clusters of ozone profiles the mean absolute error of the regressor for retrieving ozone total column was up to 8% (i.e $\sim$ 24 DU) whereas when linear regression was applied to the synthetic measurements consisting of a single cluster the mean absolute error was reduced to 3% (i.e $\sim$ 9DU). This infers that the clustering of ozone profiles was important to retrieve ozone total column.

The linear regression was also applied to the real measurements. However, the performance of the regressor deteriorated further compared to synthetic measurements. It was seen that the mean absolute error for a specific day was up to 6.5% i.e 19.5 DU (12/12/18). It was also tested on real measurements with surface albedo between 0.1-0.2 and solar zenith angles between 0° to 20°. However, the performance was not improved even with tweaking of input parameters for the regressor.

Dimensionality reduction techniques in specific principal component analysis capabilities in conjunction with linear regressors were also explored for the retrieval problem, with application to both synthetic and real measurements. It was seen that when a reduced spectrum of synthetic measurements of a specific cluster was used for linear regression the performance of the regressor deteriorated significantly (i.e MAE of 4.4% = $\sim$ 13 DU) compared to when the regressor was applied to the actual spectrum (i.e MAE of 3% = $\sim$ 9 DU). Also, when the regressor was applied to the reduced spectrum of real measurements for a specific day the error was up to 6.97% i.e 20.9 DU (12/12/18) compared to when the regressor applied on the actual spectrum (MAE of 6.5% = 19.5 DU). This concluded that the dimensionality reduction technique, when used in conjunction with linear regressor, performed worst than when linear regressor was applied directly to the original data for either synthetic or real data. The results for linear regression also stated that the retrieval problem was not linearly separable.

Finally, the MLP NN with 3 hidden layers were applied to the actual and reduced spectrum of synthetic and real measurements and its performance was analyzed. It was seen that when the neural network was trained on the reduced spectrum of synthetic measurements for one specific cluster and tested on the unseen data of the same cluster the performance of the network was less accurate (i.e MAE of 2% = $\sim$ 6 DU) compared to when it was trained/tested on the actual spectrum of the synthetic measurements for a specific cluster (i.e MAE of 5e-06% = 0.15e-04 DU). The accuracy was almost 100% for the last scenario.

The NN was also trained on the synthetic measurements for all clusters and then applied to the real data for a specific day. In this scenario, the error was too high up to 250% i.e $\sim$ 762 DU which was justified as the real and synthetic measurements were of different spectral resolution due to which the real data was interpolated which induced a great amount of noise. Thus, an attempt was made to minimize this effect by considering the reduced spectrum of the measurements for the NN. However, the error was still up to 89.9% i.e 269 DU. This summarized that since the RTM does not account for measurement noise and other factors such as degradation of the sensor, actual aerosol parameters, etc. the real spectra does not belong to the range of RTM. Thus, the network was trained using an already processed real dataset.

Lastly, NN was trained on the reduced and actual spectrum of real measurements for a specific day and tested on unseen data of the same day. For the setup, where the reduced spectrum was considered the mean absolute error was $\sim$ 6% i.e 18 DU. However, with the actual spectrum the mean absolute error of 0.7% (i.e $\sim$ 2.1 DU) was achieved. The training time of NN applied to the actual spectrum of the real dataset was $\sim$ 30 minutes for retrieving the entire spectrum which is $\sim$8 times lesser than the conventional approach (takes $\sim$4 hours for retrieving the entire spectrum). The real dataset was also tested with 4 hidden layer network and the mean absolute error was the same as achieved by 3 hidden layer network. Thus, the best result for retrieving ozone total column using the real dataset was achieved using a 3 hidden layer MLP network with a mean absolute error of 0.7% i.e $\sim$ 2 DU. This network was also tested on a different day's data and no instability was observed. Thus concludes that the ANNs trained on real measurements which have been already processed via conventional retrieval algorithms with full RTM simulations, already captures the physics behind the measurement process (expressed by the RTM), as well as instrument-related features and, are a very fast yet stable operator for the retrieval problem. This can thus be used in conjunction with a conventional approach (Eq.1.2). However, a big amount of data is required for the training procedure (such as that of S5P) [11].

7.2 Future Work

With this study it is proved that ANN can be used for the retrieval of ozone total column in real-time at a faster rate than RTM and with promising accuracy. Several further improvements can be done to make this system fail-proof and adopt it as a new retrieval approach for satellite missions with Big Data. The future work can focus on:

1. Testing other NN architectures such as CNN to check if the error can be reduced

further

2. Use the model to train on a full range of solar zenith angles and surface albedo
3. Apply the model to retrieve other trace gases
4. Hybrid usage of this model with the conventional model for a priori.

Bibliography

[1] R. Pu, *Hyperspectral remote sensing fundamentals and practices*, 2017.

[2] W. H. Bakker *et al.*, *Principles of remote sensing*. The international institute for aerospace survey and earth sciences (ITC), 2001.

[3] D. P. Lusch, *Introduction to environmental remote sensing*, 1999.

[4] (1999) Remote sensing: Introduction and history. [Online]. Available: https://earthobservatory.nasa.gov/features/RemoteSensing

[5] J. Veefkind *et al.*, "TROPOMI on the ESA Sentinel-5 Precursor: A GMES mission for global observations of the atmospheric composition for climate, air quality, and ozone layer applications," *RSE*, 2012.

[6] J. Burrows, A. Goede, C. Muller, and H. Bovensmann, "Sciamachy – the need for atmospheric research from space," *SCIAMACHY - Exploring the Changing Earth's Atmosphere*, 2010. [Online]. Available: https://atmos.eoc.dlr.de/projects/scops/sciamachy_book/sciamachy_book_ch1_springer.pdf

[7] J. P. Burrows and U. P. P. Borrell, *The remote sensing of tropospheric composition from space*. Springer, 2011.

[8] S. Deshpande, B. D. Bue, D. R. Thompson, V. Natraj, and M. Parente, "Learning radiative transfer models for climate change applications in imaging spectroscopy," 2019.

[9] M. Kataev and A. Lukyanov, "Simulation of reflected solar radiation for atmosphere gas composition evaluation for optical remote sensing from space," *Light and Engineering*, vol. 26, no. 3, pp. 14–21, 2018.

[10] B. Wyman and H. L. Stevenson, *The facts on file dictionary of environmental science*. Facts on File, 2007.

[11] D. S. Efremenko, H. Jain, D. Loyola, V. M. García, and J. Xu, "Comparison of three machine learning based schemes for solving direct and inverse problems of radiative transfer."

[12] A. Wassmann, "Ozone retrieval from satellite measurements," 2016.

[13] J. Xu, O. Schussler, D. G. L. Rodriguez, F. Romahn, and A. Doicu, "A novel ozone profile shape retrieval using full-physics inverse learning machine (FP-ILM)," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 12, p. 5442–5457, 2017.

[14] (2011) Radiative transfer. [Online]. Available: https://en.wikipedia.org/wiki/Radiative_transfer#The_equation_of_radiative_transfer

[15] R.Spurr, "Simultaneous derivation of intensities and weighting functions in a general pseudo-spherical discrete ordinate radiative transfer treatment," *J Quant Spectrosc Radiat Transfer*, vol. 75, no. 2, pp. 129–175, 2002.

[16] A. H. Alavi, A. H. Gandomi, and D. J. Lary, "Progress of machine learning in geosciences: Preface," *Geoscience Frontiers*, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.gsf.2015.10.006

[17] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks: a review," *Pattern Recognition*, vol. 35, no. 10, p. 2279–2301, 2002.

[18] A. del Águila, D. S. Efremenko, V. M. García, and J. Xu, "Analysis of two dimensionality reduction techniques for fast simulation of the spectral radiances in the Hartley-Huggins band," *Atmosphere*, vol. 10, no. 3, p. 142, 2019.

[19] A. del Águila, D. S. Efremenko, and T. Trautmann, "A review of dimensionality reduction techniques for processing hyper-spectral optical signal," *Light and Engineering*, vol. 27, no. 3, pp. 85–98, 2019.

[20] T. M. MITCHELL, *Machine learning*, ser. McGraw-Hill series in computer science. McGraw-Hill, 1997.

[21] B. D. Bue *et al.*, "Neural network radiative transfer for imaging spectroscopy," *Atmospheric Measurement Techniques*, vol. 12, no. 4, p. 2567–2578, 2019.

[22] D. Faggella. (2019) What is machine learning? [Online]. Available: https://emerj.com/ai-glossary-terms/what-is-machine-learning/

[23] M. Rouse. (2018) Machine learning. [Online]. Available: https://searchenterpriseai.techtarget.com/definition/machine-learning-ML

[24] P. Hedelt, D. S. Efremenko, D. G. Loyola, R. Spurr, and L. Clarisse, "SO_2 layer height retrieval from Sentinel-5 Precursor/TROPOMI using FP-ILM," *Atmospheric Measurement Techniques Discussions*, pp. 1–23, 2019.

[25] N. Rozemeijer and Q. Kleipool, "S5p mission performance centre level 1b readme," Royal Netherlands Meteorological Institute (KNMI), Tech. Rep., 2018. [Online]. Available: https://sentinel.esa.int/documents/247904/3541451/Sentinel-5P-Level-1b-Product-Readme-File

[26] F. Vonk, "Input/output data specification for the TROPOMI l01b data processor," Royal Netherlands Meteorological Institute (KNMI), Tech. Rep., 2018. [Online]. Available: https://sentinel.esa.int/documents/247904/3119978/Sentinel-5P-Level-01B-input-output-data-specification

[27] S. Hanna. Gsp 216 introduction to remote sensing. [Online]. Available: http://gsp.humboldt.edu/OLM/Courses/GSP_216_Online/lesson2-1/surface.html

[28] Support to Aviation Control Service. Products: Solar zenith angle. [Online]. Available: http://sacs.aeronomie.be/info/sza.php

[29] L. Su, Y. Huang, M. J. Chopping, A. Rango, and J. V. Martonchik, “An empirical study on the utility of BRDF model parameters and topographic parameters for mapping vegetation in semi-arid region with MISR imagery,” *International Journal of Remote Sensing*, 2009.

[30] J. A. Coakley. Reflectance and albedo, surface. [Online]. Available: http://curry.eas.gatech.edu/Courses/6140/ency/Chapter9/Ency_Atmos/Reflectance_Albedo_Surface.pdf

[31] NOAA Earth System Research Laboratory. (1995) Ozone measurements and distribution. [Online]. Available: https://www.esrl.noaa.gov/gmd/ozwv/dobson/papers/wmobro/ozone.html

[32] J. Xu *et al.*, “S5P/TROPOMI total ozone atbd,” Deutsches Zentrum für Luft- und Raumfahrt (DLR), Tech. Rep., 2018. [Online]. Available: https://sentinel.esa.int/documents/247904/2476257/Sentinel-5P-TROPOMI-ATBD-Total-Ozone

[33] M. V. Roozendael *et al.*, “Sixteen years of GOME/ERS-2 total ozone data: The new direct-fitting gome data processor (GDP) version 5—algorithm description,” *JOURNAL OF GEOPHYSICAL RESEARCH*, vol. 117, no. D03305, 2012.

[34] B. Boehmke and B. Greenwell, *Hands-On Machine Learning with R*. Taylor and Francis Group, 2019.

[35] C. Sorzano, J. Vargas, and A. Pascual-Montano, “A survey of dimensionality reduction techniques.” [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1403/1403.2877.pdf

[36] Z. Lateef. (2019) All you need to know about principal component analysis (PCA). [Online]. Available: https://www.edureka.co/blog/principal-component-analysis/#Principal%20Component%20Analysis%20With%20Python

[37] S. Ray. (2017) Commonly used machine learning algorithms (with python and r codes). [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/

[38] J. Brownlee. (2016) Linear regression for machine learning. [Online]. Available: https://machinelearningmastery.com/linear-regression-for-machine-learning/

[39] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning with application in R*. Springer, 2017.

[40] D. Fumo. (2017) A gentle introduction to neural networks series. [Online]. Available: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc

[41] N. Richárd. (2018) The differences between artificial and biological neural networks. [Online]. Available: https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7

[42] S. S. Haykin, *Neural networks: A comprehensive foundation*. Macmillan, 1994.

[43] Y. Upadhyay. (2019) Introduction to feedforward neural networks. [Online]. Available: https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620

[44] S.Sapna, Dr.A.Tamilarasi, and M. Kumar, “Backpropagation learning algorithm based on levenberg marquardt algorithm,” *computer science and information technology*, 2012.

[45] T. Ryder, A. Golightly, A. S. McGough, and D. Prangle, “Black-box variational inference for stochastic differential equations,” 2018. [Online]. Available: https://www.researchgate.net/publication/323118851_Black-box_Variational_Inference_for_Stochastic_Differential_Equations

[46] G. S. Levit, W. E. Krumbein, and R. Grübel, “Space and time in the works of v.i. vernadsky,” *Environmental Ethics*, vol. 22, no. 4, pp. 377–396, 2000.

[47] K. N. Liou, *An introduction to atmospheric radiation: second edition*, 2002.

[48] M. Kataev, A. Lukyanov, and A. Bekerov, “Modification of the empirical orthogonal functions method for solving the inverse task of retrieving of the CO_2 total content from satellite data,” *Journal of Siberian Federal University Engineering and Technologies*, vol. 11, no. 1, p. 77–85, 2018.