Estimating atmospheric aerosol concentrations from weather satellites

MAST90107 - Data Science Project Pt2 - Final Report

Vilberto Noerjanto (553926)
Daniel Ye (898875)
Yuhan Zhang (988299)
Zhanchi Dong (1056582)
Fuhan Sun (1131339)

Supervised by:
Dr Jeremy Silver

A report submitted in partial fulfilment of the
requirements for the degree of ***Master of Data Science***
in the
School of Mathematics and Statistics
and
School of Computing and Information Systems

THE UNIVERSITY OF MELBOURNE

October 2022

Abstract

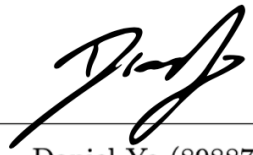
Estimates of atmospheric aerosols have many uses, such as monitoring and forecasting air quality, and one common measurement is aerosol optical depth (AOD). There are a number of established AOD retrievals from both orbiting satellites (MODIS) and surface-based observation sites (AERONET), however these data products are sparse. Either temporally sparse in the case of MODIS, or spatially sparse in the case of AERONET. Our work aims to develop an approach for AOD retrieval at high temporal and spatial resolution over Australia by leveraging high-res data from geostationary weather satellite Himawari-8. We applied machine learning techniques and used MODIS Deep Blue/Dark Target AOD product as response variable. We proposed a LightGBM model trained on 36-days worth of Himawari-8 data from 2019, with holdout R^2 of 0.41443 and RMSE of 0.05125. Benchmarking showed some agreement with a related work that evaluated MODIS DB AOD against AERONET (R^2 0.42641, RMSE 0.072), but noted that this was not a direct equivalent. Inference was also done by interpreting feature importance and dependence plots, and error analysis were also done through several case studies to better understand data nuances and uncertainties.

Declaration

I certify that this report does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

The report is 7,996 words in length (excluding text in images, tables, bibliographies and appendices).

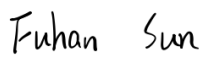
Vilberto Noerjanto (553926) — 29 OCTOBER 2022 — Date



Daniel Ye (898875) — 29 OCTOBER 2022 — Date

Yuhan Zhang (988299) — 29 OCTOBER 2022 — Date

Zhanchi Dong (1056582) — 29 OCTOBER 2022 — Date



Fuhan Sun (1131339) — 29 OCTOBER 2022 — Date

Acknowledgements

First and foremost we would like to thank our supervisor Dr Jeremy Silver, whose calm and pragmatic guidance kept us on track and allowed us to thrive. We greatly appreciate his insight and our work would not be where it is without his support.

To our families and friends, we cannot express enough gratitude for your ever-present care and companionship. The past few years were unprecedented times with unprecedented challenges, and you got us through.

Contents

1 Introduction

1.1 Problem Overview

Atmospheric aerosols, or particulate matter, are the suspension of fine liquid or solid matter in the air. These particulates originate from a wide variety of natural (e.g. fires, dust, pollen storms) and anthropogenic sources (e.g. industrial pollution, vehicular traffic).

Developing a granular understanding of the geographic spread of atmospheric aerosols is important for various environmental and health reasons. Aerosols directly and indirectly interact with Earth's radiation budget and affect the climate. Directly, they scatter sunlight back into space. Indirectly, they may offer increased opportunities for cloud seeding, indirectly affecting Earth's albedo this way and additionally impacting rainfall. Additionally, atmospheric particulates have negative effects on public respiratory and pulmonary health. This is especially true with fine particulates with diameter less than 2.5 µm.

Further motivation for modelling atmospheric aerosol levels is their functional role as atmospheric tracers. A typical characteristic of aerosols is they are more chemically inert than other chemical species. This makes them a good market of atmospheric movement which is invariant to concentration changes due to chemical reactions. Understanding their movement can be used to study how the Earth's atmosphere moves and inform climate modelling.

Particulates with sufficiently large size can be detected as they scatter and absorb light, often experienced as a reduction in visibility (haze) or reddening of sunrises and sunsets. In this study, we estimate the level of atmospheric aerosol by the quantity of 'aerosol optical depth' (AOD). AOD measures the extinction rate of a ray light as it passes through the atmosphere.

There already exist several monitoring networks for AOD. Ground-based networks such as the Aerosol Robotic Network (AERONET) measure columnar aerosol optical depth using specialised ground-based sun photometers. There exist just 24 such sensors within Australia, which are distributed predominantly around urban areas. The result is an incomplete picture for studying aerosol distribution at higher spatial granularity.

Satellite-based monitoring of atmospheric aerosol trades time-resolution to offer superior spatial resolution for AOD measurement. The Moderate Resolution Imaging Spectroradiometer (MODIS) aboard the Terra (EOS AM-1) and Aqua (EOS PM-1) satellites, has a 2,330-km-wide viewing swath for which it measures 36 discrete spectral bands. From this, we considered two AOD data products from MODIS: Deep Blue and Dark Target. As the designed purpose of MODIS is to monitor large-scale changes in the atmosphere, the satellites follow near-polar orbits which cover the Earth every day. Thus, AOD estimates for any localised region may only be obtained once per day per satellite: this AOD retrieval framework suffers from low temporal resolution.

1.2 Related Work

Previous work on AOD retrieval has been very focused on a physics-first approach to model the transmissivity of light through aerosol. This requires compensation for various factors influencing AOD, for example surface reflectance by considering vegetation index [1].

The Japan Meteorological Agency (JMA) provides their own aerosol optical depth product

(JMA AOD) using visible and near-infrared data, which is intended to be used for Asian dust monitoring [2]. Their method works by interpolating lookup tables (LUTs) representing the theoretical relationship between reflectances and aerosol properties to simultaneously estimate AOD and the Ångström exponent (a proxy for particle size). The lookup table originates from a physics-based radiative transfer model (RTM) following the method of Mano et al. (2009)[3]. The algorithm takes into account the radiances of bands 3,4,6, satellite/solar zenith/azimuth angles, and a cloud masks and land-sea masks. However, the working assumption is the particle type is dust, so this method does not work for estimating AOD influenced by other aerosol types such as haze. Furthermore, there is an issue with thick aerosol coverage which is excluded from the product due to these being included in the cloud mask. The resulting distribution of Ångström exponent estimations was also questioned by the authors. Another work by Ding et al. (2019) evaluated JMA's AOD product over Eastern China and concluded that even though subsequent revisions by JAXA did improve the product, their spatial distribution was still found to be different to that of MODIS [4].

A similar study of AOD over the city of Beijing was conducted by combining AHI with AERONET labels [1]. This model employed a superposing technique and linear regression to learn the AHI surface reflectance from AHI bands over three Normalised Difference Vegetation Index (NDVI) classifications in the area. A radiation transmission function is then used to retreive the predicted AOD. This method was shown to over-estimate AOD in dense aerosol conditions, and furthermore only validated on the three summer months. The NDVI would be lower during fall and winter and it was unclear the effect of this.

More recent work focused on a data-first approach using techniques such as deep neural networks. These may have the benefit of making fewer assumptions about the physical nature of the aerosol and environmental factors. She et al. 2020 proposed a deep neural network (DNN) to model the complicated relationship between AHI readings and AOD observations from AERONET [5]. Seventeen predictor variables are selected for the first layer as they are used in the radiative transfer model-based AOD retrieval algorithms. They are: six AHI top-of-atmosphere (TOA) reflectances, three TOA reflectance ratios derived based on dark-target (DT) assumptions, digital elevation map (DEM), AHI solar and viewing zenith, azimuth and scattering angles, and ERA-5 water vapor and ozone concentrations. Unlike the RTM-based JMA AOD product, the DNN does not make any assumptions about the surface reflectance and aerosol model. It was shown that this type of model outperforms JMA AOD and a baseline random regression forest model. However, this sort of model takes into account only the per-pixel relationship between the predictors and TOA AOD and does not make any use of the spatial temporal relationships available within the data.

Shaylor et. al. [6] presented an evaluation of AOD retrievals from MODIS over Australia over a two-decade period up to 2020. Their findings are of interest to our application, as a comparison was made between MODIS Deep Blue (DB) AOD retrievals and AERONET AOD which they treated as 'ground truth'. While not directly equivalent, this is still a relevant benchmark for our model results. They find 80% of DB retrievals fall within expected error envelope, but note there are notable discrepancies: DB is biased, systematically underestimating AOD, and there may be a sensitivity issue for AOD values below 0.02. They also note certain the spatiotemporal

variation in AOD retrievals: those based on seasonality and the characteristics of the underlying terrain. DB AOD depends on time of year, showing peaks between November and January, and minimums over June and August, coinciding with wildfire seasons and the Australian desert interior becoming dustier. They also show dependence on seasonal regions, for example cropland and heavily vegetated areas subject to biomass burning activity. We expect that these types of dependencies could be informative to a model.

1.3 Research Structure and Objectives

Our research aims to develop a mechanism for AOD retrieval at high temporal and spatial resolution over Australia. This involves leveraging data from geostationary satellite Himawari-8 and we intend to apply machine learning techniques to achieve this. Thus our key research objectives are twofold:

1. **Prediction**: Develop a model that can decently predict AOD using Himawari-8 data
2. **Inference**: Understand the relationships, nuances, and uncertainties of our predictors and response variables

On the prediction front, our work started with comparing the viability of several different modelling techniques, before pursuing the method we deemed most suitable and recommending the best configuration for it. Then we evaluated the performance of our proposed model and performed inference by interpreting relationships between variables and analysing prediction errors through several case studies.

1.4 Report Structure

Our report unfolds into the following sections. In Section 2, Exploratory Data Analysis, we present the MODIS and Himawari-8 datasets to expose the relationships and relevance of variables to our task. Section 3, Methodology, has purpose threefold: 1) it describes the methodology for the creation of training datasets, 2) it overviews the use of the mini dataset to evaluate different techniques, and 3) it details the experiments that lead to LightGBM being chosen as the most suitable model for the task. Section 4, Results and Discussion, presents our proposed LightGBM model and its performance evaluation, interpretation of variable relationships, and case study analysis of prediction errors. Finally, Section 5 concludes our work and suggests avenues for further research to iterate upon.

2 Exploratory Data Analysis

2.1 Overview of Datasets

There are two primary data sources that can be used for estimating AOD over Australia: geostationary satellite and orbiting satellites. The geostationary Himawari-8 satellite gives us multispectral image data at 16 different wavelengths, and these form our predictor variables. The orbiting Terra/Aqua satellites carry MODIS and provide us with AOD estimates, which forms our response variable.

One important first preprocessing step that would enable us to link our predictor and response variables is to spatiotemporally match the two datasets. That means filtering the Himawari-8 dataset geographically to Australia and matching them to the times that Terra and Aqua pass over Australia at daytime, at 10.30am and 1.30pm local time respectively. This preprocessing task has previously been done by our client for the period of calendar year 2019[1], so we will leverage these matched datasets throughout our project. For the purpose of EDA, we narrowed our analysis period to just a single day and we chose the first available date in the dataset which is 1 January 2019.

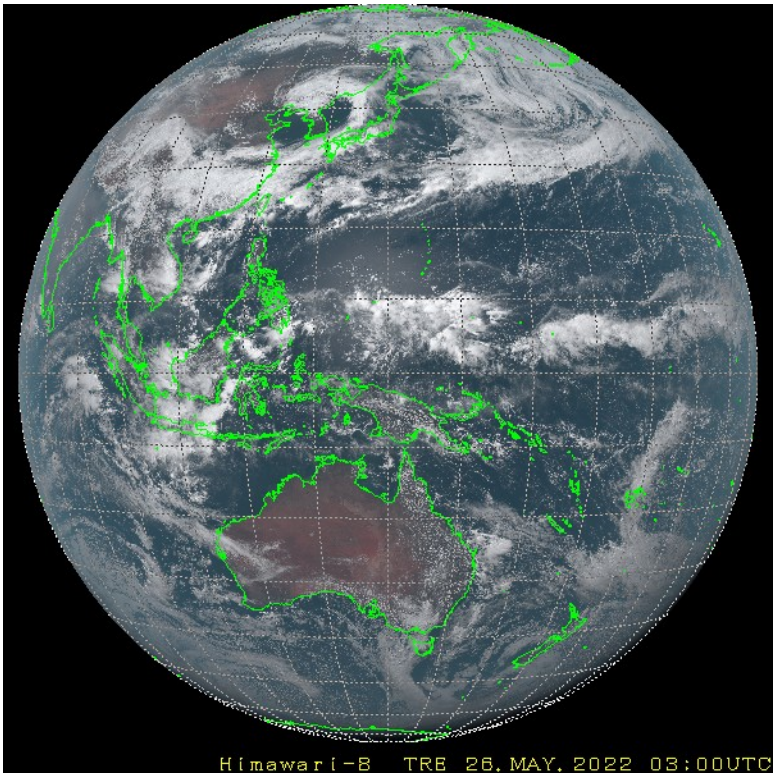


Figure 1: True colour RGB image of the Full Disk region captured by Himawari-8. Source: JMA Himawari Real-Time Image (https://www.data.jma.go.jp/mscweb/data/himawari/sat_img.php?area=fd_)

2.2 Geostationary - Himawari-8

Himawari-8 is a geostationary weather satellite that covers East Asia and Western Pacific region. Its main instrument, the Advanced Himawari Imager (AHI), is a multispectral imager that captures images at 16 different wavelengths, ranging from visible spectrum (channel 1-3), near-infrared (ch.4-6), to infrared (ch.7-16) [7]. Figure 1 shows an example of true colour RGB image of the entire observation region, referred to as 'Full Disk', which does include our region of interest Australia.

Himawari-8 captures these observations for the Full Disk region throughout the day at an interval of 10 minutes, so in its raw format it gives us 144 observations per day across 16 different wavelengths. However as previously discussed, the version of the Himawari-8 dataset we explored

[1]Data for 2020 were also provided but missing some Aqua data, and we found 2019 data to be sufficient for our use case so we only used 2019 data.

had been spatiotemporally matched to the overpass times of Terra and Aqua over Australia so that we can match these predictors to the AOD response variable.

To allow us to visually explore the data, we visualise data from one observation day (1 January 2019) in heatmaps for each channel. Figure 2 shows the heatmaps for the visible spectrum (VIS) channels 1-3, across two different processings: bidirectional reflectance factor (BRF) and scaled radiance. Figure 3 shows the heatmaps for the near-infrared (NIR) channels 4-6, again across two different processings. Figure 4 shows the heatmaps for the infrared (IR) channels 7-16, under a different processing called brightness temperature.

We can observe from Figure 2 and 3 that there seem to be different levels of mean and variance across the different channels, for example lower variance in channel 1, higher mean in channel 5. However there is a consistent trend across the 6 channels where values appear to be higher just east of the central Australia, suggesting positive correlation. The same trend cannot be seen across the infrared channels 7-16 in Figure 4, which suggests the correlations between IR channels and VIS/NIR channels could be weaker. Among the IR channels, channel 8-10 seem to have higher variance and show different characteristics to channels 11-16.

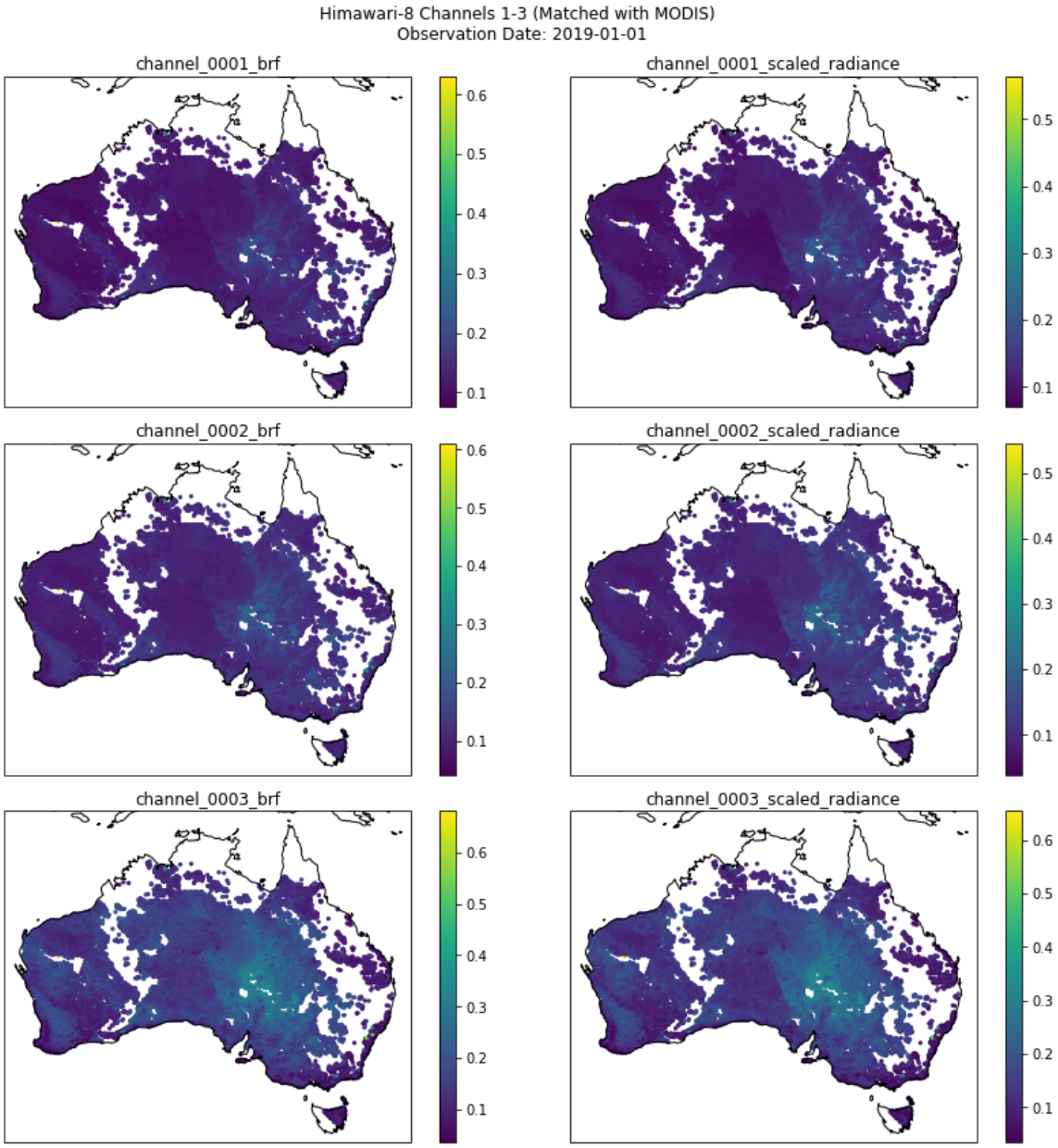


Figure 2: Heatmaps for AHI visible spectrum (VIS) channels 1-3

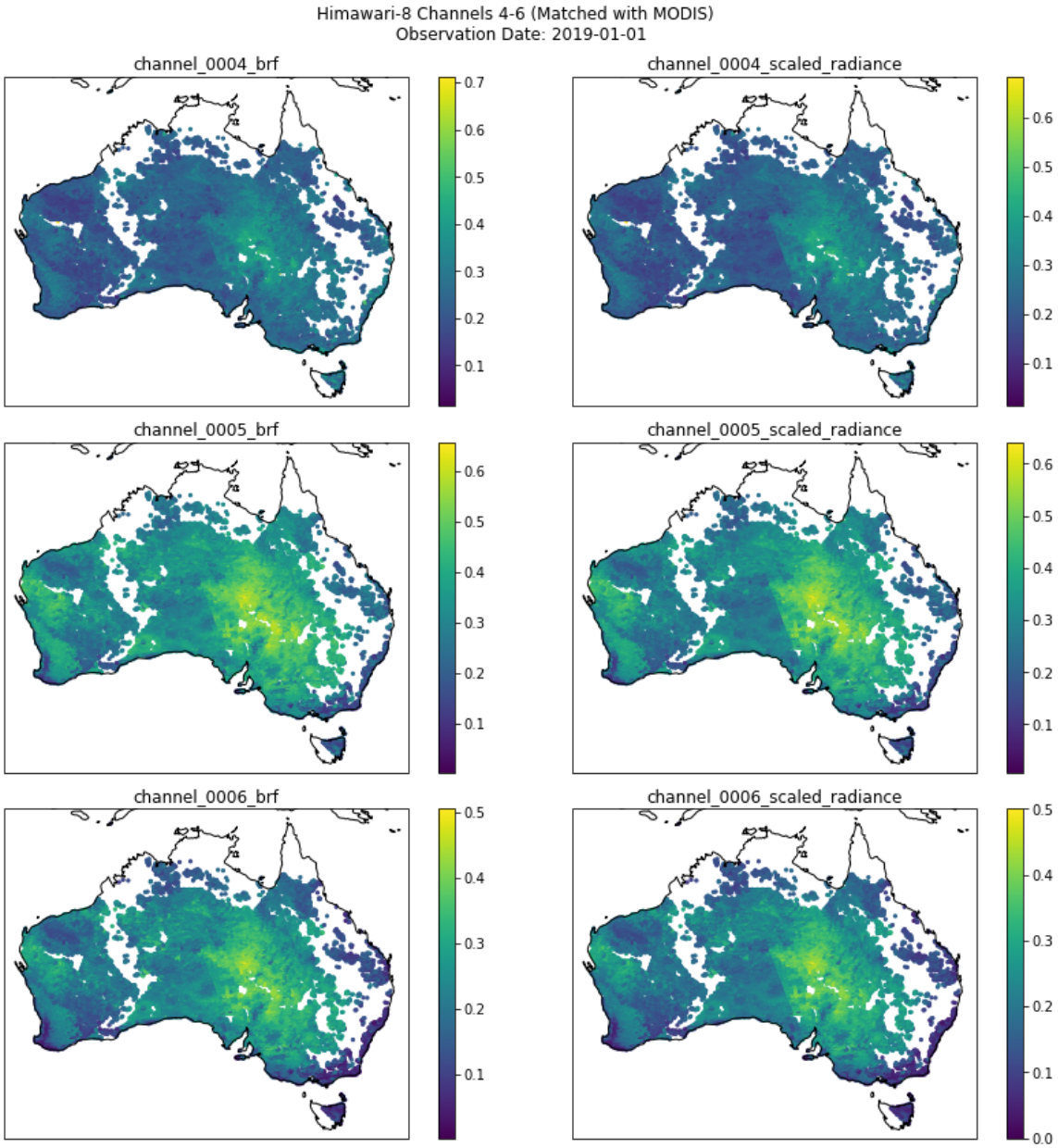


Figure 3: Heatmaps for AHI near-infrared (NIR) channels 4-6

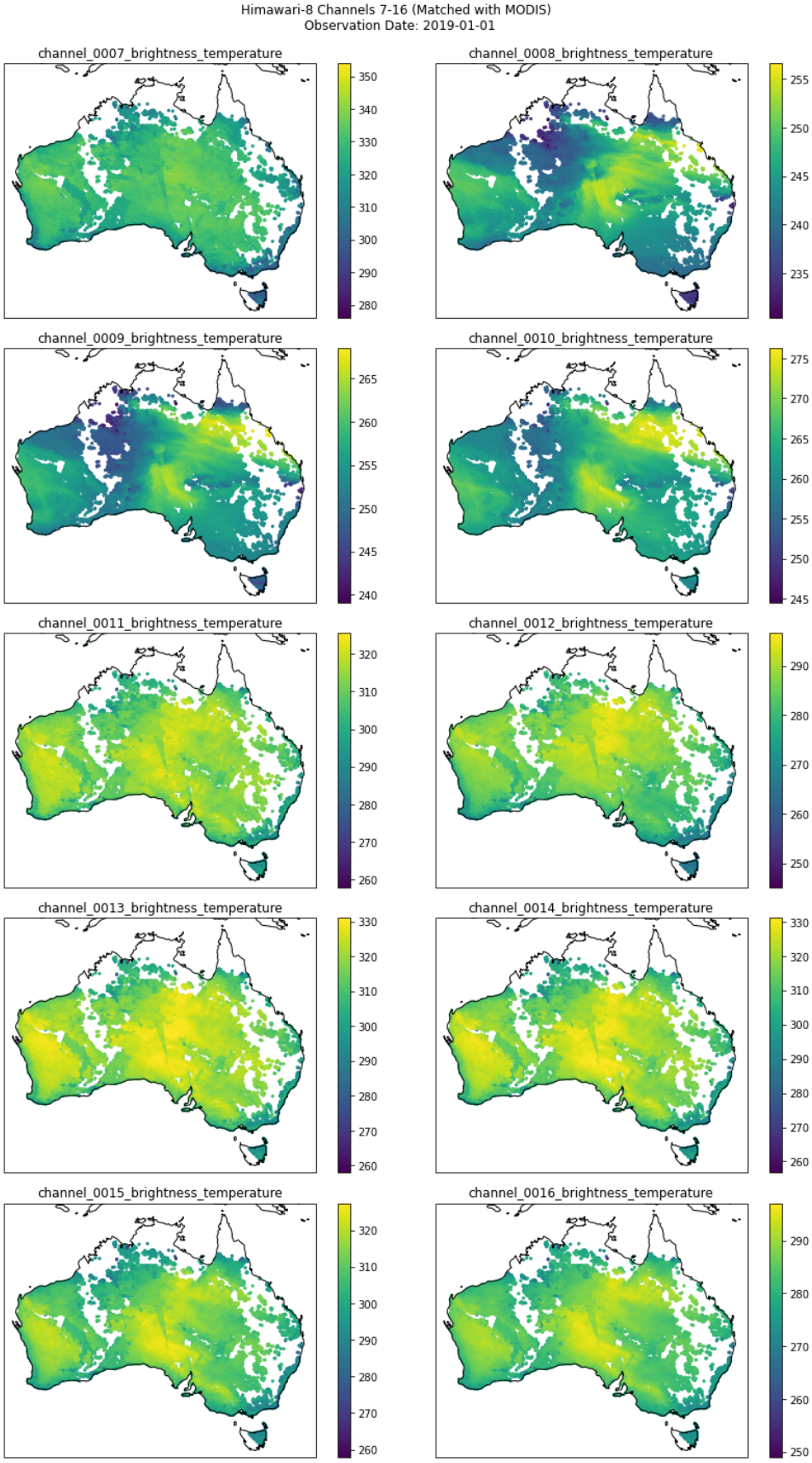


Figure 4: Heatmaps for AHI infrared (IR) channels 7-16

2.3 Orbiting - MODIS

MODIS is a sensor payload carried by two orbiting satellites operated by NASA: Terra and Aqua. In its rawest format MODIS observes data across 36 spectral bands, but on top of that NASA also provides MODIS Aerosol products which apply multiple algorithms on different spectral observations to produce AOD estimates [8]. We will use these AOD estimates as the response variable in our analysis.

The version of MODIS dataset we explored was from 2019, and was already spatially filtered to the Australia region. We again visualise data from one observation day (1 January 2019) in a heatmap as seen in Figure 5.

We can observe that while there are mostly low values with little variance in the western half of the region, the values vary a lot more in the eastern half. We can also see that there is a faint patch of higher values in the same region (east of central Australia) as the heatmaps for Himawari-8 VIS and NIR channels, even though we also see a smaller pocket of higher values further east of that patch. Visually, this heatmap bears some similarities in trend/variance with the heatmaps of VIS/NIR channels in Figure 2 and 3, which is a good sign as it suggests correlation might exist between our response and predictors as we will see in the next section.

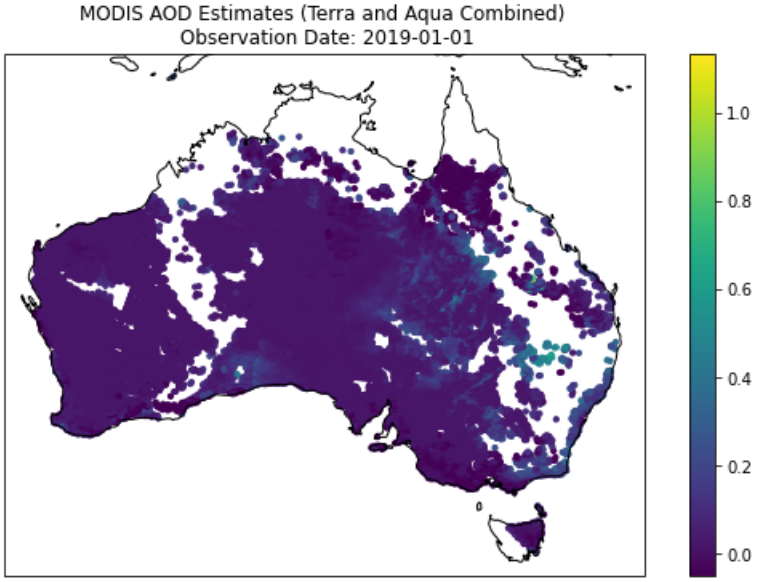


Figure 5: Heatmap for MODIS AOD estimates (Terra and Aqua combined)

2.4 Correlation Matrix

As we saw in the heatmaps, there are signs that our predictors and response variables correlate with each other. We plot the correlation matrix using data from a single observation day (1 January 2019) in Figure 6 to allow us to have a better look.

Our response variable, the MODIS AOD estimate, is located in the last row/column. We can see that AOD appears to positively correlate the most with VIS channels 1-3, followed by some weaker positive correlation with the NIR channels 4-6. These findings are somewhat consistent with what we can visually see in the heatmaps. With regard to its correlations with the IR channels, they are mostly weak but do vary. It does show some weak positive correlation with

channels 8-9 but weak negative correlation with higher-wavelength channels 11-16.

When it comes to correlations among our predictors, we can visually see two obvious 'blocks' in the chart. The VIS and NIR channels 1-6 have strong positive correlation with each other, while the IR channels also positively correlate with each other. There appears to be two different behaviour among the VIS and NIR channels, where channels 1,2,4 have weaker or negative correlation with IR channels, while channels 3,5,6 correlate positively with IR channels. Also of note is IR channels 8-10 showed weaker correlation with the other IR channels.

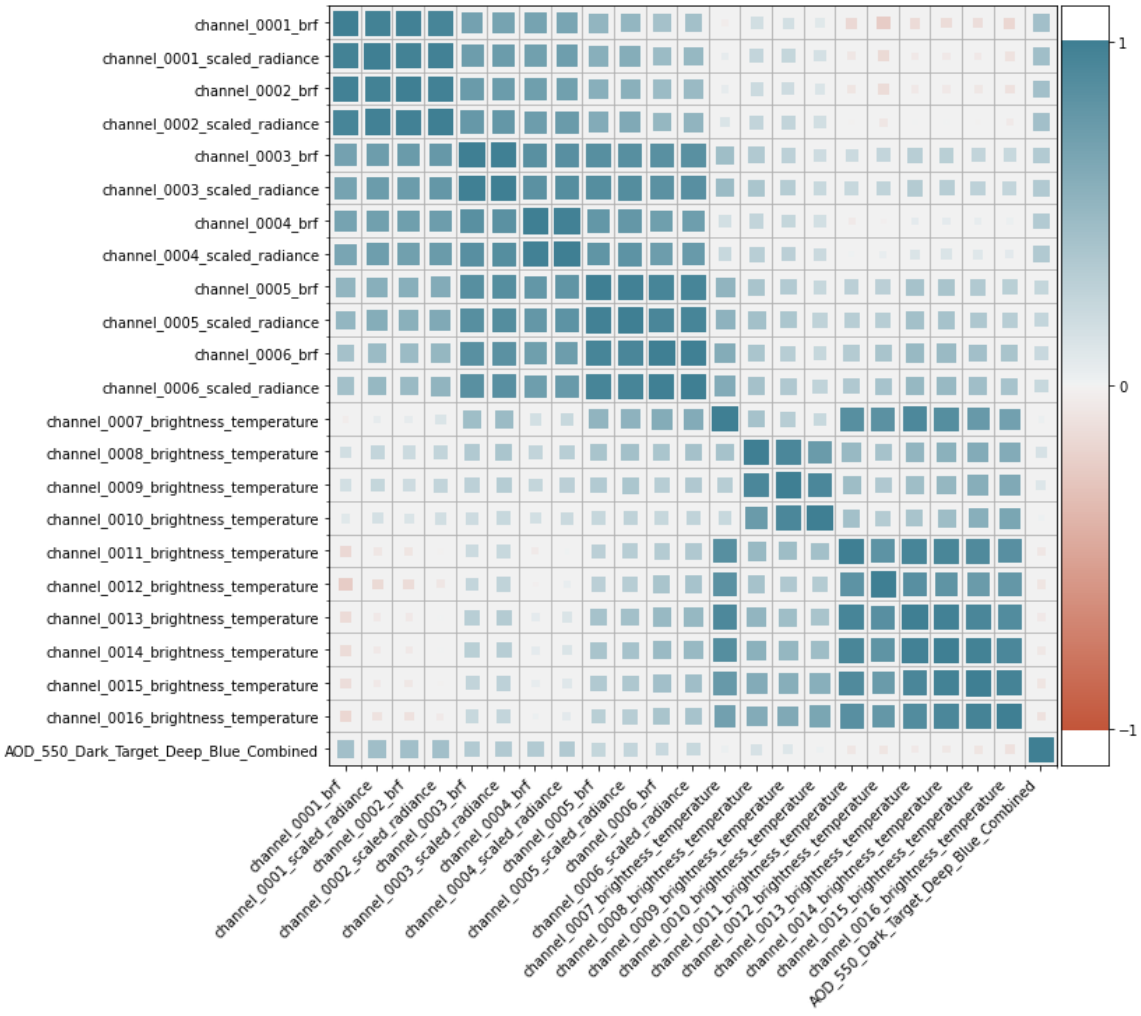


Figure 6: Correlation matrix between Himawari-8 channels and MODIS AOD estimate. Observation date: 1 January 2019.

2.5 Trend Analysis

Correlation analysis is useful in exploring the linear relationships (or lack thereof) between our variables, however it doesn't allow us to uncover trends and one-way relationships between our predictors against our response variable, especially if there exists a non-linear relationship. To explore these further, we plot the histograms for each predictor variable from Himawari-8, and at the same time overlay a line graph showing the average AOD estimate (sometimes also called 'response rate') for each bin in the histogram. This should allow us to better understand how each predictor relates to our response variable. It is worth noting ahead that the line graphs are

expectedly jagged on the tail due to low sample size, so we should focus more around the fatter part of the histograms. Similar to how we split the figures for heatmaps into three figures, we also split these charts into three. Figure 7, 8, 9 shows the trends for VIS, NIR, and IR channels respectively.

If we firstly look at the VIS and NIR trends in Figure 7 and 8, just as we saw in the correlation matrix, we can almost split these 6 channels into two groups. Channels 1,2,4 have similar trends, and channels 3,5,6 also have trends that look similar to each other. In the first group (ch1,2,4), we can see that there is a strong increasing trend on the fat side of the histograms, even though the shape is not exactly linear. However in the second group (ch3,5,6), there appears to be a distinct non-linear shape where the average AOD appear to be higher in the extreme ends, even though we still see a strong increasing trend for the most part. This observation explains why we saw that channels 5 and 6 have weak positive correlation with AOD, but as we can see in their trends they do appear to hold a non-linear relationship with AOD.

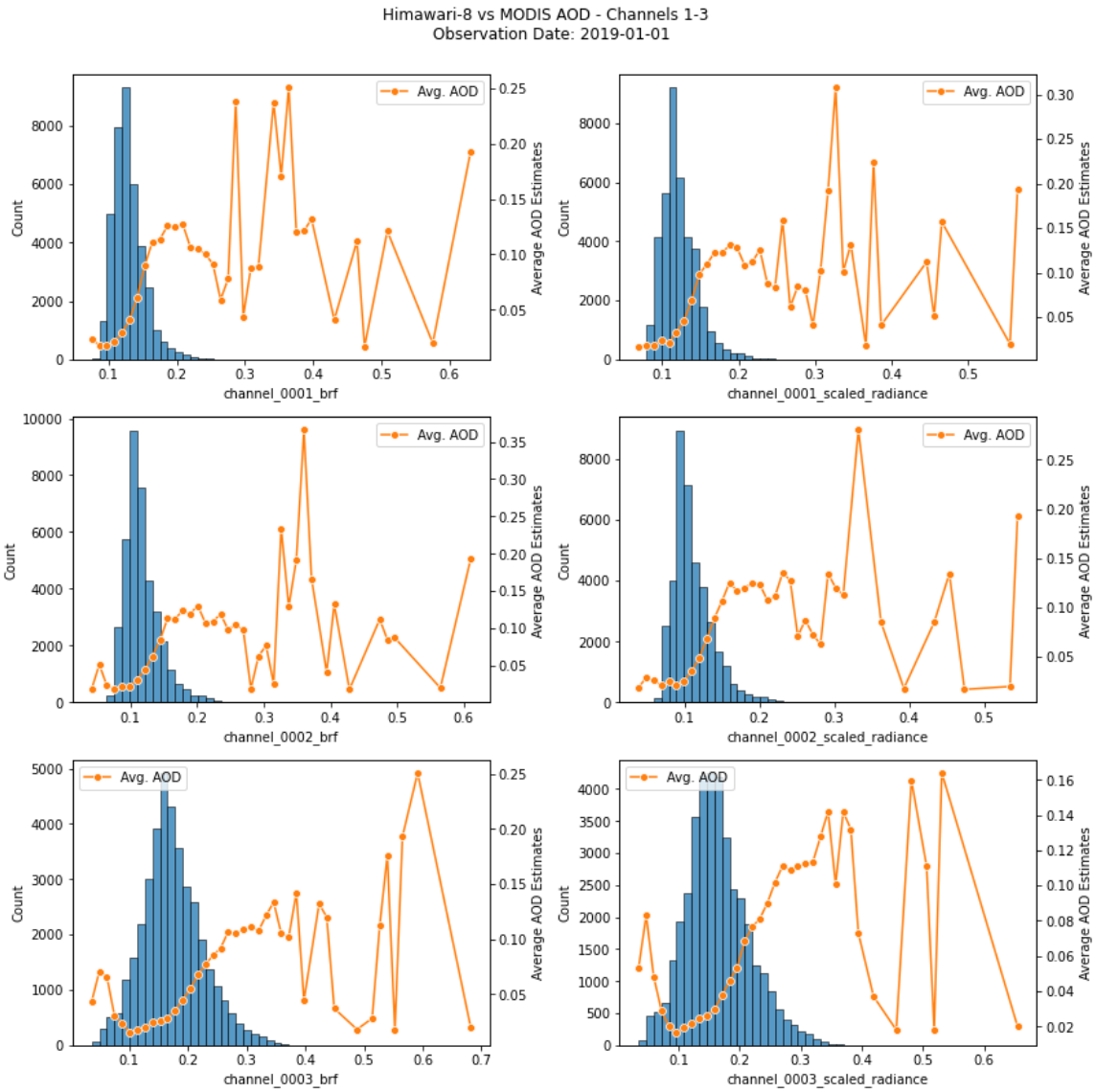


Figure 7: Trends of VIS channels 1-3 against AOD

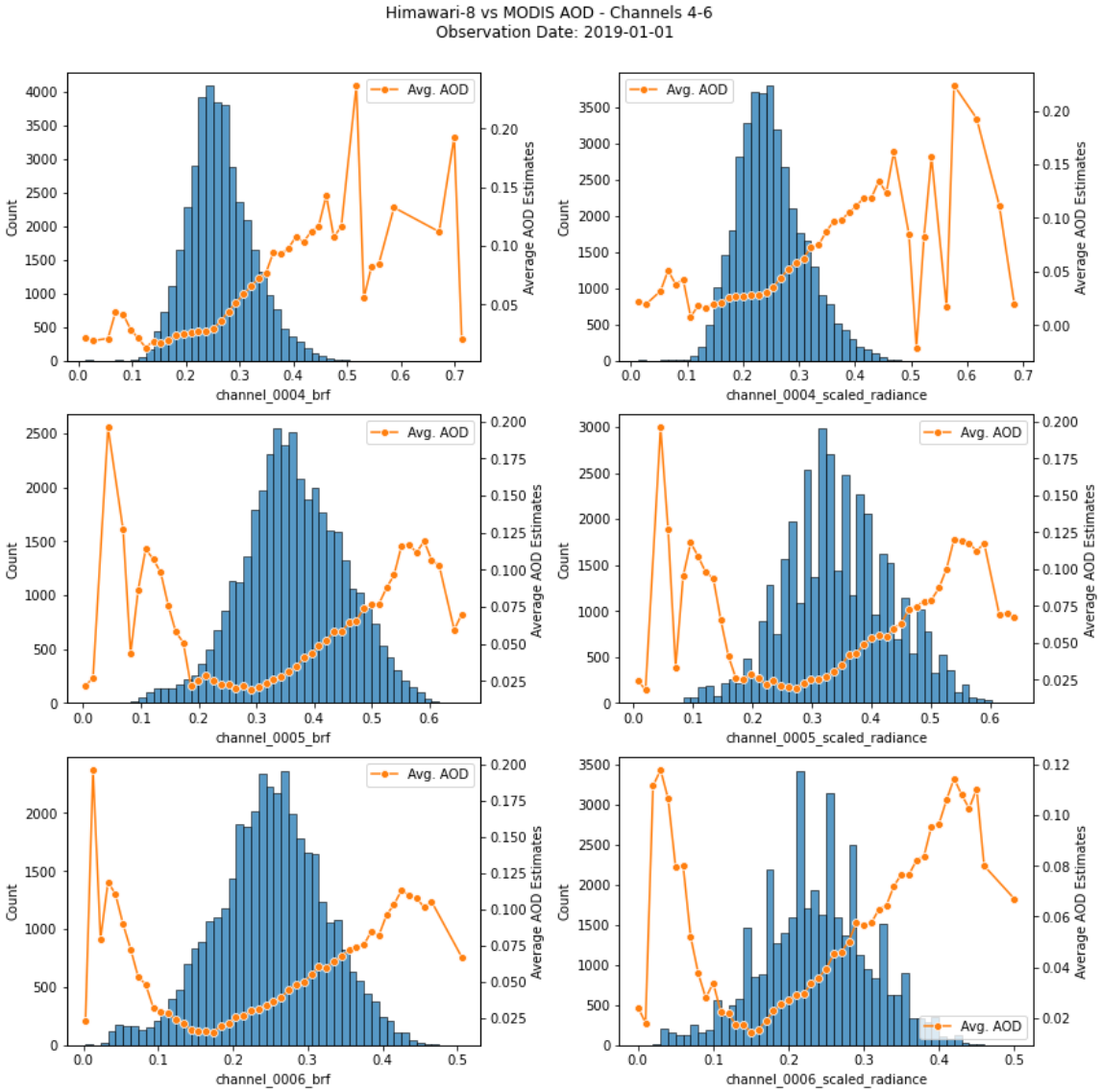


Figure 8: Trends of NIR channels 4-6 against AOD

We then look at the trends for IR channels in Figure 9, and just as we saw in the heatmaps, the relationship is starkly different to those of VIS and NIR channels. Channnels 11-16 seem to have non-linear trends that are similar in nature, with a slightly decreasing trend on the centre of the histograms which explains their weak negative correlation with AOD. Channels 8-10 shows high variance (as also seen in heatmaps), but do not seem to hold any distinct trend which could explain their almost-zero correlation with AOD. Lastly channel 7 can also be observed to have a non-linear relationship with AOD.

These observations indicate to us that some non-linear relationships do exist between our response AOD and Himawari predictors, and therefore we should explore modelling techniques that are capable in capturing such non-linear relationships.

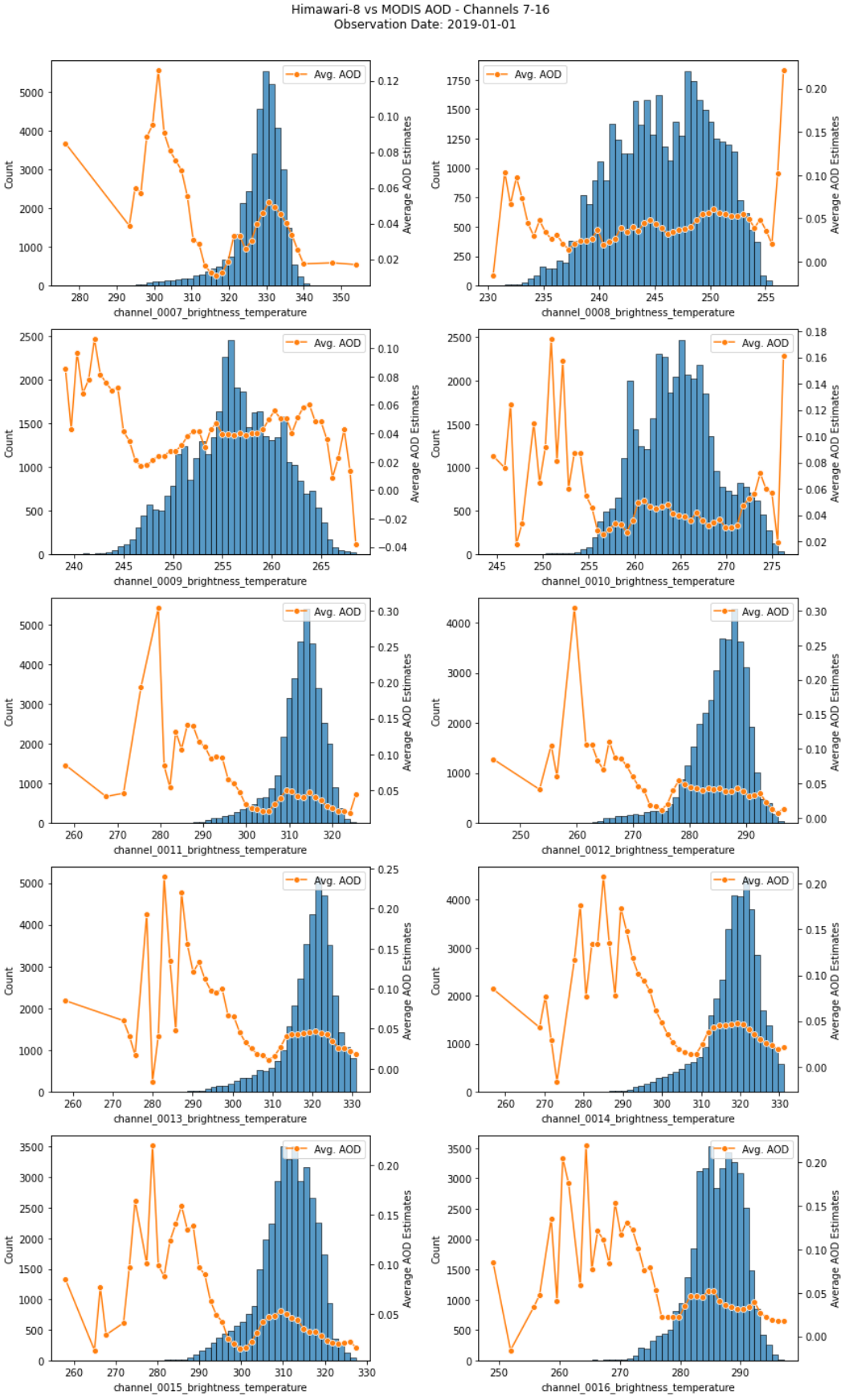


Figure 9: Trends of IR channels 7-16 against AOD

3 Methodology

3.1 Training Data Creation

We will firstly describe how we created the train/validation/holdout sets that will be used for modelling. We were provided with data from calendar year 2019 (we discarded 2020 data in this project because it's missing some data from Aqua), which gives us approximately 16 million rows across 16 predictors. Taking into consideration the volume of data and limited computing resource, whereby we trained models locally on personal computers, we consulted our client and decided to downsample to keep runtime and resource usage manageable. We agreed that training on smaller data should be sufficient for the scope of our project, and the model could be trained on larger data and productionised in future work.

		Train	Validation	Holdout
Day of month		1^{st},10^{th},20^{th}	8^{th},16^{th}	28^{th}
Main Set	Sampling frac	100%	100%	100%
	Sample size	1,640,398	1,088,946	496,836
Mini Set	Sampling frac	3%	3%	3%
	Sample size	49,193	32,656	14,900

Table 1: Summary of training data creation sampling methodology

When sampling the data, it is important that we still maintain coverage for the entire year to account for variations that occur at different months or seasons. To enable this, we used *day-of-month sampling* to create our *Main Set* and chose $1^{st}/10^{th}/20^{th}$ for train set, $8^{th}/16^{th}$ for validation set, and 28^{th} for holdout set. These dates were chosen to maintain an approximate 3:2:1 ratio between train/validation/holdout, while also keeping coverage on the different weeks of the month. It is also worth mentioning that we intentionally made sure that dates from the 2019-20 Black Summer Bushfire, e.g. 20 December 2019, were included in our training data. (We will discuss the bushfire case study further in section 4.3.3.)

Throughout our experiments, we found that even after using *day-of-month sampling* some models with higher time complexity still took too long to train (as we will see in section 3.2.6). While this would naturally eliminate some modelling techniques from being chosen, we would still like to be able to compare their performance on the same training data, so we created the *Mini Set* solely for this purpose by randomly sampling 3% from *Main Set*.

3.2 Modelling Technique Selection

We explored several popular regression techniques, and there are two important considerations in choosing one that best suits our use case. First, the model must be able to capture non-linear relationships given we had seen from our EDA that our predictor-response relationships are non-linear. And secondly, given the size of our *Main Set* and limited computing resource, the model must be able to handle larger data locally and train within reasonable time to allow for further experimentation.

3.2.1 Random Forest

Random forest (RF) is the first regression technique we explored in this project. According to Breiman (2001), random forest regression is a supervised learning algorithm that uses an ensemble learning method based on the decision tree [9]; it could be described as the combination of decision tree and bagging. In the ensemble step, the mean value of decision trees would be calculated as the predicted value of RF regression. Moreover, all the calculations by decision-trees are run in parallel due to the usage of bagging. Compared with the simpler decision trees, RF regression improves accuracy and reduces overfitting.

From our EDA, it is clear that there exists non-linear relationships between our predictors and response variables, and as a tree-based method, random forest could capture these relationships including the interactions between features. However, this technique can be quite impractical on larger data due to the time complexity so it will be a challenge to use this technique on our *Main Set*. We trained a random forest model on our *Mini Set*, and the results will be discussed in section 3.2.6.

3.2.2 Support Vector Machine

Support vector regression (SVR) [10] is an application of SVM (Support Vector Machine) for regression problems. SVM has the objective of maximising the "distance" to the nearest sample point in the hyperplane, while SVR has the objective of minimising the "distance" to the farthest sample point in the hyperplane. SVM is traditionally a popular technique for classification tasks and it has the ability to capture non-linear relationships through the use of non-linear kernels, hence we chose to include this technique in our analysis.

Before training the model, we firstly normalised the data. This is a pre-processing method that is commonly used when the range of feature values vary quite widely, or when they each use different units. Since SVMs try to maximise the distance between the decision plane and the support vector, features with disproportionately large values could affect the calculation result more than other features. Thus, normalisation puts all features on the same scale. In choosing the appropriate kernels, we tried a couple non-linear kernels and landed on using the radial basis function (RBF) kernel.

We ran into a few issues while testing out this technique. The first issue is the time it takes to train SVR models. SVMs are known to have a relatively high time complexity and can be impractical on larger data, so we trained the model on the *Mini Set* which still took more than 1 hour. The slow training time also meant that it makes it harder for us to perform hyperparameter tuning. We initially intended to tune using grid search, however this ended up taking way too long so we just manually tuned some of the parameter values. These impracticalities meant that we know this technique was unlikely to be selected because it would take too long to experiment on, however we would still like to compare the performance on the *Mini Set* against other techniques, which are summarised in section 3.2.6.

The second issue we ran into is for a reason that is unknown, our SVR models always produced an R^2 that is negative. While mathematically it is not impossible to get this result, it is still very unusual. We attempted to investigate and resolve this issue, but did not manage to find a model that produced a positive R^2. Considering we won't be pursuing this technique due

to its slow training time, we eventually decided to deprioritise this investigation and redirect our effort elsewhere.

3.2.3 Generalised Additive Models

Generalized Additive Model (GAM) is the extension of Generalized Linear Model (GLM) which combines the properties of the generalized linear model and additive model[11]. As for the relationship which must be a simple weighted sum, Generalized Additive Model relaxes this limitation and assumes that the results can be modeled by the sum of an arbitrary function of each feature[12]. As a result, Generalized Additive Model allows us to model non-linear data and learn non-linear features while still maintaining interpretability.

In the GAM, the coefficients of the predictor variables in the linear predictors would be replaced by a smoothing function. This smoothing function is called a spline which is a flexible and complex function for modelling the non-linear relationships for each feature. A spline is a piecewise polynomial curve that connects more than two polynomial curves[13]. And the GAM is composed of the sum of all splines. According to this parameter, the wiggles of model prediction line would be affected which means the more the splines, the wigglier the prediction line for all feature in the model[14]. While too many splines would lead to overfitting to the training data. Another parameter is λ which is a smoothing parameter to control the smoothness of the prediction function. λ can penalize the splines, so we can adjust the λ simply to control the wiggles of prediction function and avoid overfitting. Here the higher λ is, the less wiggly our line will be, until it reaches a straight line.

GAM poses several advantages. The first one is interpretability[15]. Due to its additive property, the interpretation of marginal effects of individual variables does not depend on the values of the other variables in the model. Here we can output the partial dependence plots to interpret the marginal effects of individual feature on the prediction results. Secondly, unlike when training fully parametric regression models, the selection of the best model in GAMs does not require constructing a large number of transformations. GAM also applies regularisation, done through the smoothing parameter λ that allows us to control the smoothness of prediction function to avoid overfitting. Despite these strengths, one big drawback of GAM is its inability to capture interactions between features, unlike tree-based methods. In scenarios such as ours where the relationships between variables are complex, this would mean that GAM might not be able to predict as well as the tree-based methods.

To compare its performance against other techniques, we trained GAM on the *Mini Set*, testing several different splines, λ and shape constraints. We compare the results against other techniques in section 3.2.6.

3.2.4 Neural Network

Deep Neural Networks (DNN) have been previously proposed in related works [5] for the task of estimating AOD. DNN are known to be universal function approximators [16], and thus allows us to model the non-linear relationships between the features and response. A series of experiments help us determine best parameters for this model, and we will describe the final architecture and the process in this section.

The architecture we use is multilayer perceptron (MLP). The final model chosen with hyperparameter tuning has 16 input layers, two hidden dense layers with 96 and 192 neurons respectively, and a single output layer outputting the final AOD. The number of hidden layers are independently selected from a range between 32 and 512 dense units in 32 unit increments. We note that this architecture is one deeper than that of She et. al. [5], as the single hidden layer MLP showed initial performance wanting of our other methods.

As seen previously, each of the 16 input features have different ranges. Normalisation is first applied before the input layer to centre the distribution around 0 with a standard deviation of 1. This helps stabilise model training as the weights will then be of similar scale.

On the topic of regularisation, we apply batch normalisation which also reduces the problem of internal covariate shift[17]. Random dropout of 10% of the neurons is done within each layer (except the output layer) to help generalise the model and avoid overfitting. The level of dropout is a tuned quantity between 10% and 20%.

The network is optimised with the adaptive moment estimation optimiser (Adam) observing the mean absolute error loss function. Default Tensorflow hyperparameters are kept for Adam except the initial learning rate, which is tuned between 0.01, 0.001, and 0.0001. We use the rectified linear unit (ReLU) activation function, which is chosen to avoid the vanishing gradients and exploding gradients problem in DNN that may arise when using other activation functions such as the sigmoid activation function [18].

The hyperparameter search space here is large, so we have opted to use the Hyperband Tuner [19] which is a multiarmed-bandit approach to hyperparameter tuning. Combining our small hyperparameter space across the width of the two hidden layers, learning rate and dropout values entails a classic grid search through 1536 hyperparameter combination options. Hyperband speeds up this search by adaptively exploring randomly sampled configurations and exploiting those with good initial validation loss.

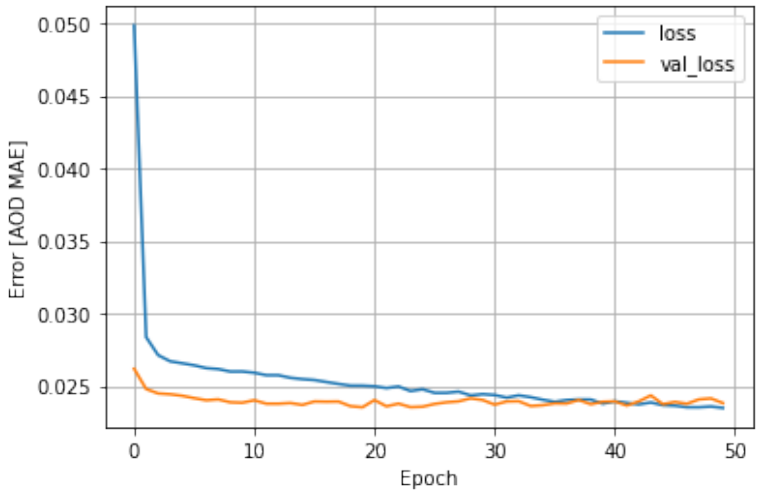


Figure 10: Learning curves best Multi-layered Perception tuned with Hyperband: 96 and 192 wide hidden layers, learning rate of 0.01 and dropout of 10%.

We trained the network on the *Mini Set* over 50 epochs and observe the validation set to detect overfitting and stop learning 10. It achieves a holdout R^2 of 0.140 on the *Mini Set*.

3.2.5 Gradient Boosting - LightGBM

Gradient boosting machines (GBM) [20] is an ensemble modelling technique that uses boosting to create a strong learner from a set of weak learners, in this case simple decision trees, and it builds its model stage-wise by adding weak learners using a gradient descent like procedure, hence the name. It has become a very popular technique [21] for classification and regression tasks due to its ability to produce state-of-the-art results on many benchmarks [22]. As a tree-based method, GBM is naturally able to capture non-linear relationships, including interactions between predictors, which is suitable for our problem. It is for these reasons we included GBM as one of the techniques to explore. One drawback it has is lack of intrinsic interpretability which is to be expected of a tree-based ensemble method, but as we will see later in section 4.2.2 there are methods to mitigate this through post-hoc interpretability and this should help us in our inference task.

One of the most popular [21] implementations of GBM is LightGBM, which aimed to address the efficiency and scalability issues of other GBM implementations by sacrificing a small degree of accuracy [23]. Some studies [24, 25] have found LightGBM to be faster than other methods such as XGBoost[26] and CatBoost[27], but it is worth noting that these performance gains were likely tested on datasets much larger than ours so in our case the difference might not be too noticeable. Rather, we chose LightGBM over XGBoost due to prior knowledge and familiarity, and also for its simpler interface.

For the purpose of performance comparison, we trained a LightGBM model on the *Mini Set* using all 16 baseline predictors while keeping parameter values as default. The resulting performance will be discussed in the next section.

3.2.6 Comparison of Performance

Technique	**Package**	R^2	**MSE**	**RMSE**	**Training Time**
Random Forest	scikit-learn	0.292	0.00248	0.0498	$<$ 10mins
SVM	scikit-learn	*-0.001	0.13886	0.3691	$\sim$ 1.5hrs
GAM	pyGAM	0.115	0.00311	0.0557	$<$ 30secs
Multilayer Perceptron	Keras	0.140	0.00302	0.0550	$<$ 5mins
Gradient Boosting	LightGBM	0.156	0.00296	0.0544	$<$ 30secs

Table 2: Comparison of *holdout* error metrics from different techniques trained on the *Mini Set*. *Issue with SVM metrics discussed in section 3.2.2

For the purpose of performance comparison, we trained all 5 models on the *Mini Set* because SVM and random forest (RF) both took too long to train on the *Main Set*. This meant that we eliminated both of them from being selected even though RF seems to have done pretty well on the *Mini Holdout Set*. From the results we saw in Table 2, we ended up selecting LightGBM for its efficiency and predictive power, and it outperforms the other techniques when we take into account both of those factors. From our subsequent experiments (discussed in next section), we saw that we could reliably train LightGBM models on the *Main Set* which has more than 30 times the amount of data within less than 5 minutes. This is a very important consideration for us as it enables us to iterate quickly on much larger data, while the other methods either took

a long time (SVM, RF, MLP) or unable to capture interactions and hence is not as predictive (GAM).

3.3 LightGBM Experiments

We did further experiments on LightGBM after selecting it as our chosen modelling technique, experimenting on different feature sets and tuning hyperparameter values to find the best configuration.

3.3.1 Feature Sets Experiments

We firstly set a baseline model that uses all 16 Himawari-8 channels as its feature set, before we experiment by adding and removing features.

In the first experiment, we added a categorical ancillary predictor for land cover classification from MODIS/Terra+Aqua Land Cover Type Yearly L3 Global 0.05Deg CMG product (short name: MCD12C1). The MCD12C1 data product provides us with majority land cover type at 0.05 deg resolution based on International Geosphere-Biosphere Programme (IGBP) classification scheme, which was derived from a year's input of Terra and Aqua MODIS observations [28]. We spatially matched this dataset to our training data, and trained a model using this new feature plus our 16 baseline features. We could see in Table 3 that there were small improvement in error metrics, however we also note that the model dimensionality doubled because it now includes a 17-class categorical feature so we decided to stick with the simpler baseline model and not pursue the land cover predictor in this project.

Feature Set	**Notes**	**Validation RMSE**	**Holdout RMSE**
All 16 channels[#]	Baseline	0.06124	0.05125
All 16 channels + land cover	Ancillary categorical predictor for land cover from MCD12C1	**0.06078**	**0.05071**
Ch.1-4,6,16	Top 6 from Gini importance	0.06306	0.05193
Ch.1-3,7,12,13	Top 6 from SHAP importance	0.06241	0.05425

Table 3: Results from LightGBM feature sets experiments. [#] denotes chosen

We then experimented with making our models simpler by using just the top 6 features according to feature importance, and we do this twice using two different importance methods, Gini and SHAP. In both experiments, the models seem to slightly underfit and reducing feature set did not improve performance. Therefore we again chose the baseline model.

3.3.2 Hyperparameter Tuning

We performed hyperparameter tuning on 3 parameters that were suggested by LightGBM's documentation to tune for better accuracy [29]. The first one we looked at was number of boosting iterations where the default 100 was not enough and the model underfit, while the error flattens out after 1000 iterations and therefore we chose 1000. We then looked at learning rate where lowering it actually increased errors so we chose the default 0.1. The last hyperparameter we tuned was `max_bin`, which is the maximum number of bins that continuous features get

binned into. The theory is that higher number of bins may increase training accuracy but may cause overfitting. Here we saw that increasing `max_bin` doesn't necessarily improve our validation RMSE even though eventually it did when we increased it to 1000, while reducing it to 125 also did not improve performance. For these reasons we kept `max_bin` to the default value of 255. Results from these tuning experiments are summarised in Tables 4, 5, and 6.

No. iterations	**Validation RMSE**	**Holdout RMSE**
100*	0.06443	0.05243
200	0.06318	0.05209
500	0.06183	0.05134
1000#	0.06124	**0.05125**
2000	0.06093	0.05156
3000	**0.06080**	0.05217

Table 4: Results from tuning no. of iterations. * denotes default, # denotes chosen

Learning rate	**Validation RMSE**	**Holdout RMSE**
0.2	0.06259	0.0528
0.15	0.06154	0.05175
0.1*#	**0.06124**	**0.05125**
0.05	0.06184	0.05130
0.01	0.06402	0.05215
0.001	0.06792	0.05758

Table 5: Results from tuning learning rate. * denotes default, # denotes chosen

max_bin	**Validation RMSE**	**Holdout RMSE**
125	0.06159	0.05089
255*#	0.06124	**0.05125**
500	0.06145	0.05155
750	0.06177	0.05153
1000	**0.06073**	0.05139

Table 6: Results from tuning max_bin. * denotes default, # denotes chosen

4 Results and Discussion

4.1 Proposed Model

In previous sections we have discussed why we chose LightGBM and the subsequent experiments we performed using that technique, and here we propose our final model which is specified in Table 7. We will discuss the results and performance of this model in the next sections.

Method		Gradient Boosting - LightGBM
Feature set		All 16 Himawari-8 AHI channels
Response		MODIS Terra+Aqua AOD estimates
Data period		Calendar year 2019
Day-of-month sample	Train	1^{st},10^{th},20^{th}
	Validation	8^{th},16^{th}
	Holdout	28^{th}
Parameters		No. of iterations: 1000 Default for everything else
Holdout error metrics	R^2	0.41443
	MSE	0.00263
	RMSE	0.05125

Table 7: High-level specification of our proposed model

4.2 Diagnostics

4.2.1 Performance Evaluation

Our proposed model has a holdout R^2 of 0.41443 which means, in a rough sense, that our model was able to recognise around 41% of the variance that occurs in the holdout set, while achieving an RMSE of 0.05125. Table 8 summarises the rest of the metrics.

Metric	Train	Validation	Holdout
R^2	0.73071	0.33989	0.41443
MSE	0.00159	0.00375	0.00263
RMSE	0.03984	0.06124	0.05125

Table 8: Error metrics summary of our proposed model

In order to benchmark these results, we looked at a 2022 paper by Shaylor et al. that evaluated two decades' worth of MODIS AOD retrievals [6]. More specifically, we were interested in their evaluation of MODIS Deep Blue (DB) AOD, which forms a significant part of our response variable (for reference, our response variable combines both Deep Blue (DB) and Dark Target (DT) AOD retrievals). The authors evaluated DB AOD by comparing them against surface-based AOD data from AERONET which they treated as 'ground-truth', and from the results summarised in Figure 8 of their paper, this comparison yielded an R^2 of approximately 0.42641 and RMSE of 0.072. We recognise that their comparison of DB against AERONET is not directly equivalent to our error metrics which compare our predicted AOD against DB/DT AOD. However, we believe that it is still a relevant benchmark to compare against and the agreement gives us a little indication that our model is on the right track. In future work, we

could look at evaluating our predictions against AERONET which would yield results that are more applicable to those of Shaylor et al (2022).

Figure 8 of Shaylor et al. (2022) also showed a density scatter plot for DB vs AERONET, so we produced a similar plot that compared our Predicted vs Observed for the holdout set (see Figure 11 below). We could observe that our scatter plot holds a similar shape to theirs which is what we want, however it is worth noting that for now our plot doesn't allow us to compare density. We didn't manage to generate the density overlay because it was too resource intensive for the volume of holdout data we had, so one option for future work is to sample from the holdout and reproduce the scatter plot with density.

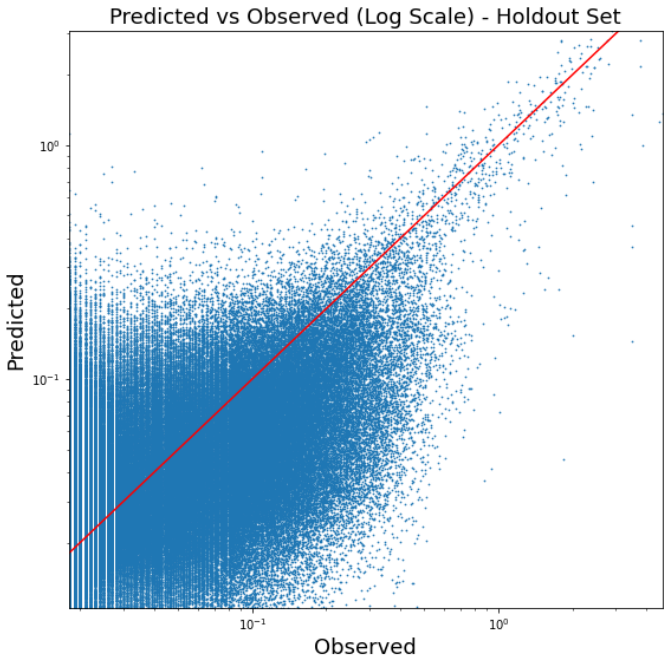


Figure 11: Scatter plot of holdout Predicted vs Observed AOD using our proposed model

Another diagnostic we looked at is the progression of our training and validation errors throughout model training iterations to check for overfitting. We noticed from Table 8 that our training errors are considerably lower than validation and holdout, which is quite commonly seen in machine learning models, however we wanted to make sure that our model doesn't severely overfit. We could see from Figure 12 that even though the two errors are diverging, at 1000 iterations the validation errors were still slowly declining and had not risen so it wasn't severely overfitting.

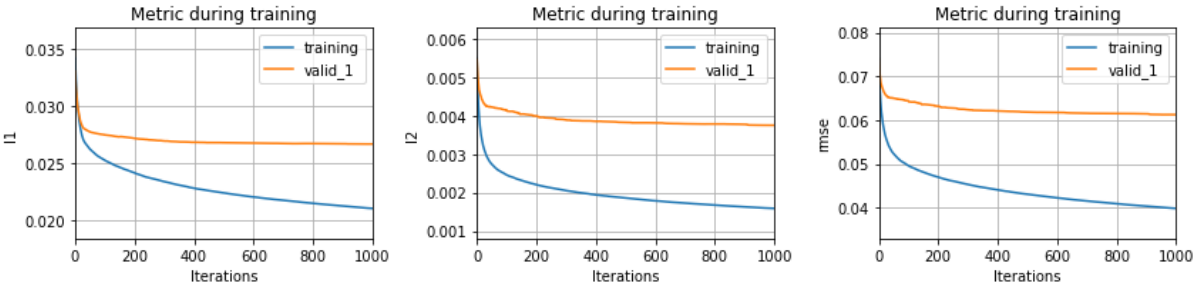


Figure 12: L1, L2, RMSE error curves over training iterations

4.2.2 Interpretation - Feature Importance and Dependence Plots

Our decision to use gradient boosting machines was primarily motivated by its predictive power and practicality in handling larger data, however tree-based ensemble methods are traditionally considered to have low interpretability [30]. There are many post-hoc interpretability methods, from the more traditional methods such as Gini, to the more recent ones such as SHAP (SHapley Additive exPlanation) [31]. We chose SHAP for a number of reasons. Firstly, SHAP addresses the consistency problems of Gini, whereby it is possible for Gini to lower a feature's importance when its true impact on the model increases [32]. SHAP also has a well-established integration with LightGBM which lowers coding and implementation effort. Lastly and importantly, SHAP assigns local attribution values to each individual point which enables it to provide us with feature importance, dependence plots, and also interaction visualisation.

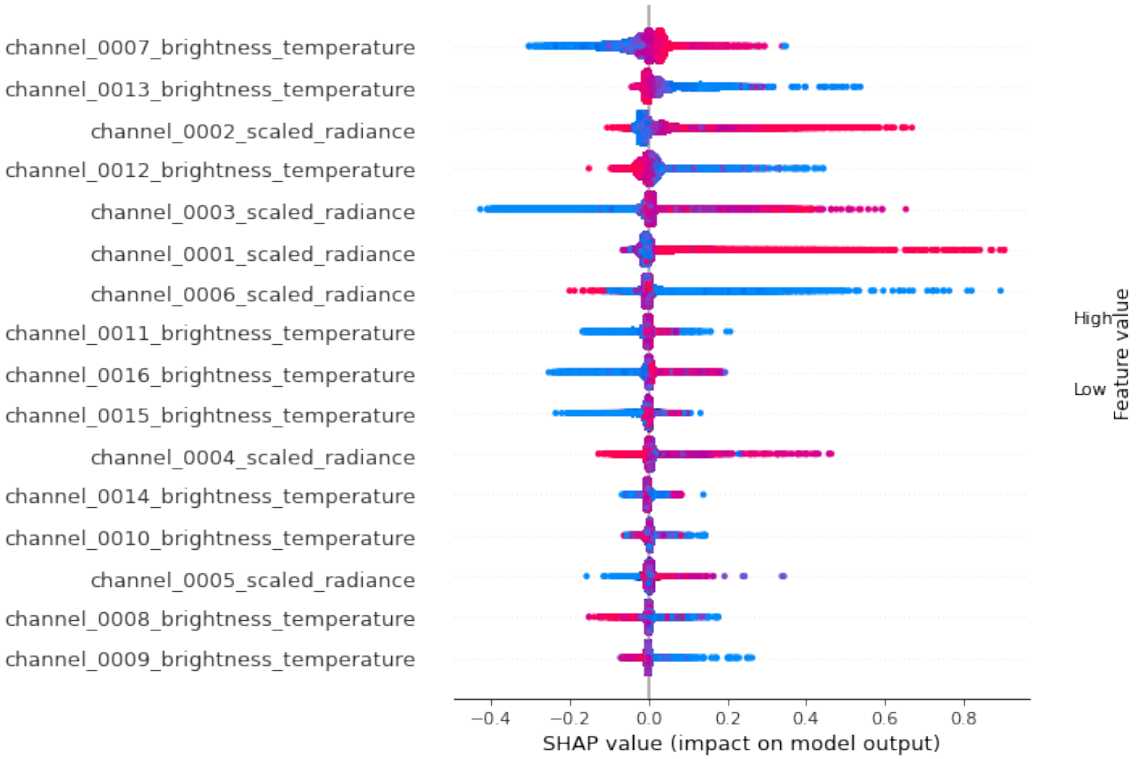


Figure 13: List of features ordered by importance, and a summary of their SHAP values

We firstly used SHAP to look at feature importance. Figure 13 shows a list of features

ordered by importance, and also summary of their SHAP values which is a measure of each feature's impact on each individual prediction. From the figure we can see that channels 7 and 13 are our two most important features.

The importance summary plot is very useful in telling us which features are important, but it can be quite difficult to see why that is the case just by looking at that figure. To do that we looked at SHAP dependence plots that show how a feature impacts model output. They are quite similar to the classic PDPs (Partial Dependence Plots) but with one key addition of interaction visualisation, where SHAP chooses a secondary feature that it deems has the most interaction with the primary feature, and colours the points accordingly.

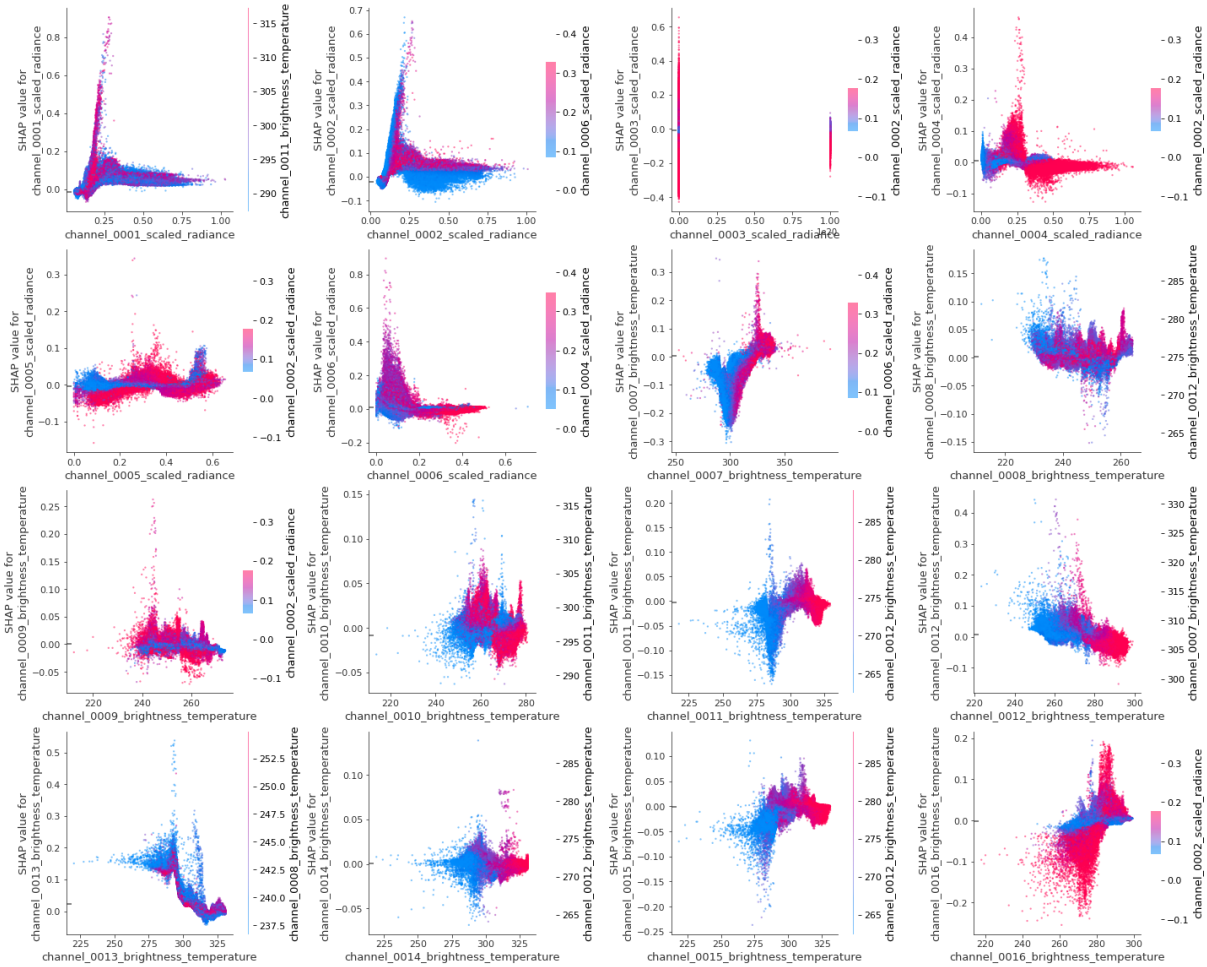


Figure 14: SHAP dependence plots of all 16 features, ordered by channel 1 to 16

From the dependence plots in Figure 14 [2] [3] , we can see why important features such as channel 7 and 13 are deemed important due to their strong relationships with the response, while channels 8 and 9 on the lower end of importance have visibly flatter shape. These plots also enable us to identify interactions between features if they are present. For example, we can see from the dependence plot of channel 7 that the feature does interact with channel 6,

[2]Plot size is scaled down for report conciseness, full size plots available in our GitHub repository which we've linked in the Appendix.

[3]Ch3 plot highlights a possible data issue for that feature. Issue was not there initially so it was not picked up during EDA, and it occurred after our dataset was updated halfway through Semester 2 to fix for nulls and missing values. Future works should look into fixing this issue.

where ch6 values can help separate the SHAP values in cases where there is a vertical dispersion of different SHAP values for a same ch7 value. This means that even though it has lower importance, the values of channel 6 could be interacting with the values of our most important feature. The complex dependence shapes and feature interactions we're seeing from our variables could be a potential explanation on why the simpler models with reduced number of features we experimented on ended up performing worse, as we saw in section 3.3.1.

4.3 Error Analysis

In addition to evaluating our model's performance through error metrics and diagnostics, we also wanted to gain a better understanding on our prediction errors so that we get a handle on some of the uncertainties and nuances of our data. We looked closer at the errors through three different lenses: validation vs holdout difference, seasonal spatial variation, and bushfire impact.

4.3.1 Validation Error vs Holdout Error

We could see from Table 8 that our proposed model seems to perform slightly better on the holdout set than on the validation set. While it is not that uncommon for holdout error to be slightly lower than validation error, it is still worth reflecting on how we created those sets and check for any potential data leakage.

Recall that we used *day-of-month sampling* to create our validation and holdout sets, so one notable difference between these two sets is their temporal distance to the training set. The validation set that contains data from the 8^{th} and 16^{th} days of the month has temporal distance of 2 and 4 days to the train days respectively, while the holdout set with data from the 28^{th} days has temporal distance of between 1 to 4 days (depending on the month). This leads to us hypothesising the possibility that error gets lower the closer the test data is to training days (in temporal distance). To check this, we split the validation set to separate data from 8^{th} and 16^{th}, and recalculated their errors. From Table 9 we can see that while predictions for the furthest day (16^{th}) has the highest errors, there is no obvious pattern on which one has the lowest errors. It is also difficult to test for the significance of these differences because we cannot use two-sample statistical tests that require the assumptions of independence and/or normality, such as the two-sample t-tests or Mann-Whitney U test.

Set	Day of month	Closest distance to train day	MSE	MAE
Validation	8^{th}	2 (to 10^{th})	0.003483	**0.024584**
Validation	16^{th}	4 (to 20^{th})	0.004018	0.028750
Holdout	28^{th}	Between 1-4 (to 1^{st})	**0.002627**	0.026621

Table 9: Validation and holdout errors, split by day of month and its distance to train days

From these observations, we don't believe that there are obvious data leakage that should be addressed, however we think there are a few suggestions to mitigate this risk further in future work. Firstly, there may be merit in making the size of both validation and holdout sets to be the same so their metrics become more comparable. Secondly, training on larger data such as the full year would increase the coverage of training days and enable us to eliminate this difference in temporal distance. Lastly we could also align closer to a truer 'out-of-time validation' by

using 2020 data as validation and holdout so we can be more assured that the model did not receive information it's not supposed to know during training.

4.3.2 Seasonal Spatial Variation of Errors

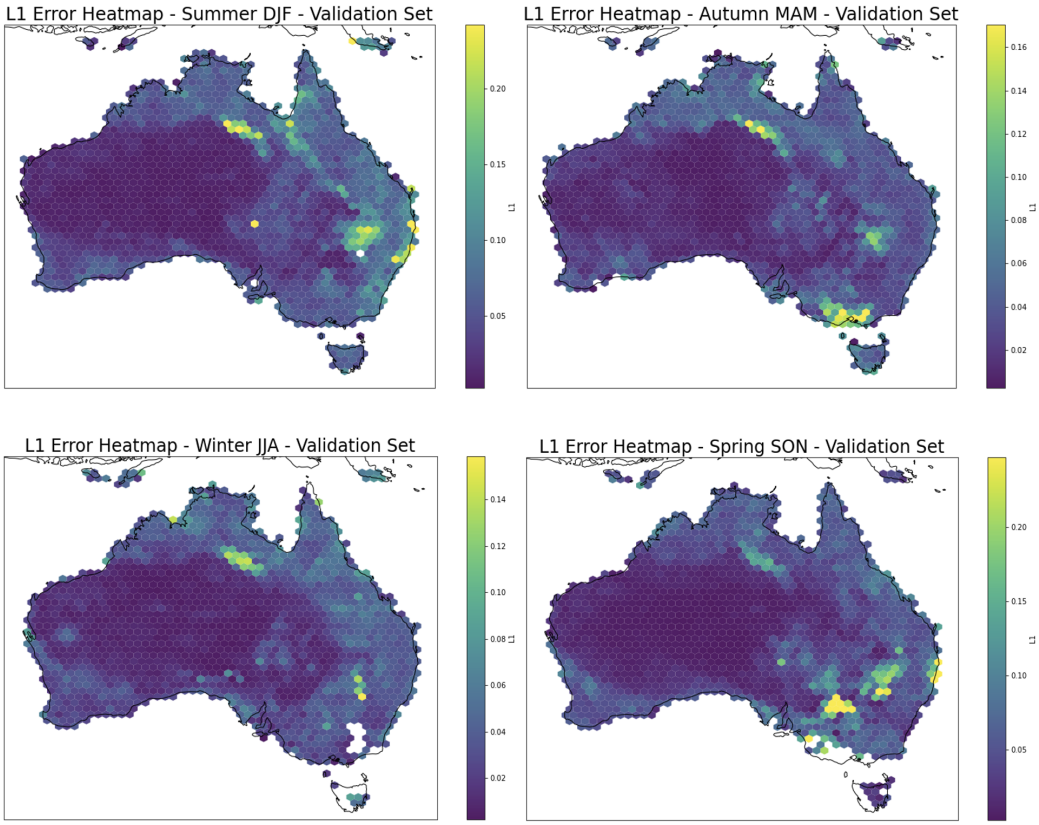


Figure 15: L1 validation error heatmaps for the different seasons

We also analysed the seasonal spatial variatons of errors by separating our validation set by season and plotting the L1 errors as spatial heatmaps. We can observe a number of variations across the seasons. Firstly, the large area around Tarrabool Lake in Northern Territory have relatively higher errors across all seasons, with the highest being in the summer. This could be due to changes in land cover where the lakes could potentially dry up during those months. A similar thing could be observed in spring, where a group of lakes near Mungo National Park in Southwestern New South Wales also have higher errors likely also due to changes in land cover. We also see that coastal areas tend to have higher errors, and this is likely due to cloud cover being more prominent in coastal areas which often leads to low sample size.

From these observations we could see that there are spatial variations throughout the seasons and they potentially also interact with land cover properties. One possibility for future work could be to revisit the land cover predictor, while also adding a temporal predictor for season or time of year.

4.3.3 Case Study - 2019-20 Black Summer Bushfire

We also did a case study to see how haze from bushfire affects our model predictions. We selected two dates from the 2019-20 Black Summer Bushfire period for this analysis, 17 and 20 December 2019, where satellite images from MODIS in Figure 16 showed significant haze on both days. Furthermore, these dates were chosen as such because data from 20$^{\text{th}}$ is included in our training data while data from 17$^{\text{th}}$ is not, and it is of interest to analyse this difference.

We plotted the L1 errors as spatial heatmaps for both dates, and we can make a general observation that areas covered by thick haze seemed to have relatively higher errors, possibly due to sparser data, cloud/haze ambiguity, and/or higher AOD range. However it is also worth noticing that the errors are lower on the 20$^{\text{th}}$ despite the higher level of haze seen on that day. This is likely due to the fact that data from 20$^{\text{th}}$ is in our training data, and this highlights the need to include bushfire impacted dates in our training data.

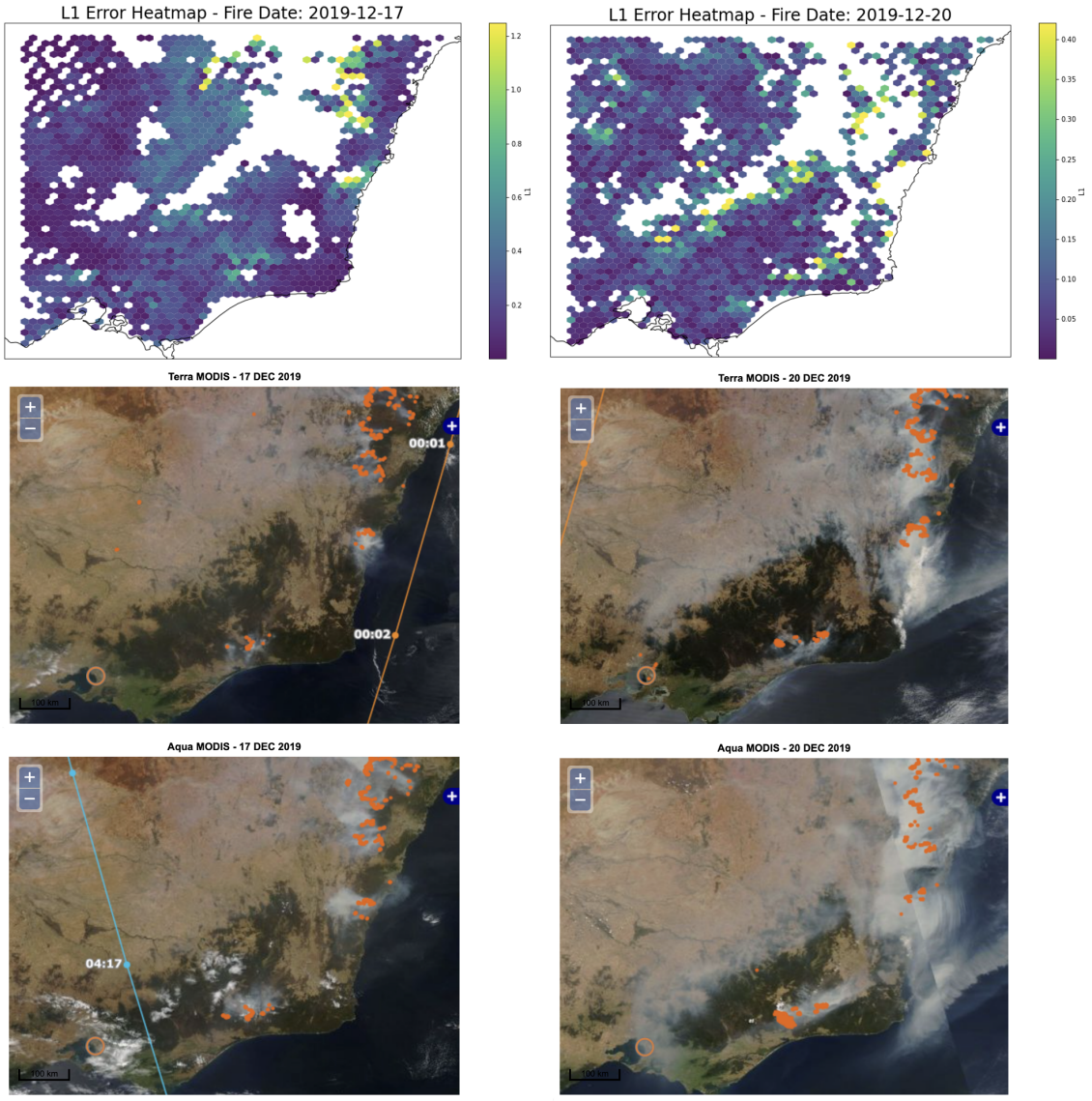


Figure 16: L1 error heatmaps and MODIS images from bushfire dates 17 and 20 December 2019 Source: AERONET AOD Data Display Interface, Aspendale site (https://aeronet.gsfc.nasa.gov/cgi-bin/data_display_aod_v3?site=Aspendale_Mel_AU)

5 Conclusion and Future Directions

AOD retrievals have many use cases within the industry, such as monitoring and forecasting of air quality, however the majority of established AOD retrievals either come from orbiting satellites (MODIS) or surfaced-based observation sites (AERONET). Those data sources are sparse, either temporally sparse in the case of MODIS, or spatially sparse in the case of AERONET. This motivates the problem of retrieving AOD using denser data sources, such as that of geostationary weather satellite Himawari-8.

In this work, we had the key objectives of developing a model that can decently predict AOD using Himawari-8 data, and also the inference task to better understand the relationships and uncertainties around our predictor and response variables. We proposed a model in section 4.1, whose performance compares well against a relevant benchmark presented by Shaylor et al. (2022). Our proposed model architecture is also scalable − training took less than 5 minutes on the *Main Set* which has 1.6mill training rows. With productionisation effort and slightly more computing resource, future works should be able to retrain the model with much more data and score within reasonable time frame.

On the inference front, our model's feature importance and dependence plots give us an insight into the relationships and interactions between our variables, and this would also help with future prediction and inference work. Our error analysis and case study explained some of the errors and helped us in understanding some of the nuances and uncertainties of our data. This would enable future works to mitigate these errors by accounting for these nuances, such as creating new features or expanding our training data.

We do believe that there are a few ways to further enhance the work that is already delivered. The proposed model could be further optimised in a few ways: 1) re-train it on much bigger data which should increase its predictive power, 2) revisit the land cover predictor to see if there is a better way to use it, and 3) use methods like PCA to reduce dimensionality. The way we evaluate our results could also be improved by validating against AERONET, which would involve spatiotemporal matching our data with AERONET sites.

We could also explore the alternative of switching our response variable from MODIS's DB/DT AOD to its MAIAC AOD retrieval, which according to Shaylor et al. (2022) outperforms DB when evaluated against AERONET [6]. Lastly, we have seen from our error analysis that spatial relationships have the potential to improve AOD predictions. Therefore, future works could also explore more advanced techniques to build models that are spatially aware, with one option being to leverage image-based techniques from computer vision that could capture spatial relationships using convolutional neural networks (CNNs).

Appendix

Git repository

Our code, experiment logs, and plots are stored in our private GitHub repository. Invites have been sent to Prof Michael Kirley, Dr Joyce Zhang, and Dr Jeremy Silver, and it should be accessible through this link.
Alternatively, paste this URL https://github.com/vilberto/DataScienceProject-Group3

Meeting minutes

Minutes for our weekly meetings can be found in this link.
Alternatively, paste this URL https://geode-saturnalia-584.notion.site/73d184f1185f47a7b226d7017a846337?v=a965898ce8a0494db6fda6e3cdfc0cd6.

References

[1] L. Wang, C. Yu, K. Cai, F. Zheng, and S. Li, "Retrieval of aerosol optical depth from the himawari-8 advanced himawari imager data: Application over beijing in the summer of 2016," *Atmospheric Environment*, vol. 241, p. 117788, 2020.

[2] U. Daisaku, "Aerosol optical depth product derived from himawari-8 data for asian dust monitoring," *Meteorol. Satell. Cent. Tech. Note*, vol. 16, pp. 56–63, 2016.

[3] Y. Mano, "Maximum information composite channels of a high-resolution satellite sounder," *Papers in Meteorology and Geophysics*, vol. 60, pp. 1–6, 2009.

[4] L. Ding, Q. Kai, L. Wu, J. Xu, H. Letu, B. Zou, Q. He, and Y. Li, "Evaluation of jaxa himawari-8-ahi level-3 aerosol products over eastern china," *Atmosphere*, vol. 10, p. 215, 04 2019.

[5] L. She, H. K. Zhang, Z. Li, G. de Leeuw, and B. Huang, "Himawari-8 aerosol optical depth (aod) retrieval using a deep neural network trained using aeronet observations," *Remote Sensing*, vol. 12, no. 24, p. 4125, 2020.

[6] M. Shaylor, H. Brindley, and A. Sellar, "An evaluation of two decades of aerosol optical depth retrievals from modis over australia," *Remote Sensing*, vol. 14, no. 11, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/11/2664

[7] Japan Meteorological Agency, "Geostationary meteorological satellites — Himawari-8/9," 2017. [Online]. Available: https://www.jma.go.jp/jma/kishou/books/himawari/201703_leaflet89.pdf

[8] NASA, "MODIS Aerosol product." [Online]. Available: https://atmosphere-imager.gsfc.nasa.gov/products/aerosol

[9] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] V. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[11] "Generalized additive model." [Online]. Available: https://en.wikipedia.org/wiki/Generalized_additive_model

[12] N. Pya, "Additive models with shape constraints," Ph.D. dissertation, University of Bath, 2010. [Online]. Available: https://purehost.bath.ac.uk/ws/portalfiles/portal/187948884/UnivBath_PhD_2010_N_Pya.pdf

[13] "What is a generalised additive model?" [Online]. Available: https://towardsdatascience.com/generalised-additive-models-6dfbedf1350a

[14] J. Maindonald, "Smoothing terms in gam models," *Smoothing Terms in GAM Models*, 2010. [Online]. Available: https://maths-people.anu.edu.au/~johnm/r-book/xtras/autosmooth.pdf

[15] K. Larsen, "Gam: The predictive modeling silver bullet," 2015. [Online]. Available: https://multithreaded.stitchfix.com/blog/2015/07/30/gam/

[16] G. Cybenko, "Mathematics of control, signals, and systems-aproximation by superpositions of a sigmoidal functions," 1989.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[18] D. Sussillo and L. Abbott, "Random walk initialization for training very deep feedforward networks," *arXiv preprint arXiv:1412.6558*, 2014.

[19] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018. [Online]. Available: http://jmlr.org/papers/v18/16-558.html

[20] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[21] Kaggle, "State of Data Science and Machine Learning 2021," 2022. [Online]. Available: https://www.kaggle.com/kaggle-survey-2021

[22] P. Li, "Robust logitboost and adaptive base class (abc) logitboost," ser. UAI'10. Arlington, Virginia, USA: AUAI Press, 2010, p. 302–311.

[23] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[24] H. Alshari, Y. Saleh, and A. Odabas, "Comparison of gradient boosting decision tree algorithms for cpu performance," *Erciyes Tip Dergisi*, pp. 157–168, 04 2021.

[25] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1937–1967, 2021.

[26] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016. [Online]. Available: http://arxiv.org/abs/1603.02754

[27] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *CoRR*, vol. abs/1810.11363, 2018. [Online]. Available: http://arxiv.org/abs/1810.11363

[28] NASA, "MCD12C1 - MODIS/Terra+Aqua Land Cover Type Yearly L3 Global 0.05Deg CMG - Overview." [Online]. Available: https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/MCD12C1#overview

[29] "LightGBM - Parameters Tuning." [Online]. Available: https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html

[30] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, ser. Springer Texts in Statistics. Springer US, 2021. [Online]. Available: https://books.google.com.au/books?id=5dQ6EAAAQBAJ

[31] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," *CoRR*, vol. abs/1705.07874, 2017. [Online]. Available: http://arxiv.org/abs/1705.07874

[32] S. M. Lundberg, G. G. Erion, and S. Lee, "Consistent individualized feature attribution for tree ensembles," *CoRR*, vol. abs/1802.03888, 2018. [Online]. Available: http://arxiv.org/abs/1802.03888