A Machine Learning Algorithm to Detect Fog from Space

MPALG, MPCAS

NILS JOHNSSON
KEVIN SVENSSON

DEPARTMENT OF SPACE, EARTH AND ENVIRONMENT
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Degree project report 2024

A Machine Learning Algorithm to Detect Fog from Space

NILS JOHNSSON
KEVIN SVENSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Earth, Space and Environment
Chalmers University of Technology
Gothenburg, Sweden 2024

A Machine Learning Algorithm to Detect Fog from Space
NILS JOHNSSON
KEVIN SVENSSON

Supervisor: Chiara Ceccobello, AI Sweden
Supervisor: Ronald Scheirer, SMHI
Examiner: Patrick Eriksson, Department of Space, Earth and Environment

Degree project report 2024
Department of Space, Earth and Environment
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Car in traffic on a foggy road. Photo by chris-mueller / iStock.

Typeset in LaTeX
Gothenburg, Sweden 2024

A Machine Learning Algorithm to Detect Fog from Space
NILS JOHNSSON, KEVIN SVENSSON
Department of Space, Earth and Environment
Chalmers University of Technology

Abstract

Fog detection is important for traffic safety. Detecting fog using machine learning on satellite data has been researched before, but not on a global scale using synthetic data. The aim of the thesis is to use a synthetic dataset of simulated MODIS satellite data to determine the viability of machine learning algorithms for detecting fog in satellite images. The synthetic dataset we use is simulated using a fast radiative transfer model called RTTOV by inputting various atmospheric information for different conditions. The dataset is tabular and no spatial or temporal relationship exists between the data points meaning each pixel is treated independently. We use the synthetic data to train and evaluate numerous machine learning models including various implementations of XGBoost and feed forward deep neural networks. We also apply a model trained on synthetic data to a real MODIS image. We demonstrate that classification models can achieve good recall values on synthetic data when oversampling fog in the training data, the best being 0.87 recall with a deep neural network. However, we find that this comes at the cost of a large amount of false positives evident by the low precision value of 0.27. It is concluded that no model performed satisfactory results for replacing existing methods of fog detection. We identify the acquisition of supplemental labeled real satellite images as a possibility for future improvement, allowing for spatial analysis which is impossible with the independent pixels of the synthetic dataset alone. However, this is a non-trivial task due to the challenges in obtaining and labeling a sufficiently large and diverse dataset of real satellite images.

Keywords: MODIS, fog, machine learning, nowcasting.

Acknowledgements

We would like to thank our supervisors, Chiara Ceccobello and Ronald (Rolle) Scheirer, for their guidance, support and encouragement throughout this project.

We also thank our examiner, Patrick Eriksson, for his helpful feedback and assistance.

Thank you all for your contributions.

Nils Johnsson, Kevin Svensson, Gothenburg, May 2024

List of Acronyms

Below is the list of acronyms that are used throughout this thesis listed in alphabetical order:

AI	Artificial Intelligence
COT	Cloud Optical Thickness
DNN	Deep Neural Network
GBDTs	Gradient Boosting Decision Trees
IPA	Independent Pixel Approximation
MODIS	Moderate Resolution Imaging Spectroradiometer
RTTOV	Radiative Transfer for TOVS
SMHI	Swedish Meteorological and Hydrological Institute
TOA	Top-of-atmosphere
XGBoost	Extreme Gradient Boosting

Contents

1
Introduction

Ground-level atmospheric impairment of visibility, commonly known as fog, presents a considerable challenge to the safety and efficiency of transportation across land, water, and air. Vehicles on highways, ships at sea, and aircrafts arriving and departing all face significant risks when fog reduces visibility. The unpredictable and often sudden onset of foggy conditions makes it a critical aspect of weather forecasting, particularly in the realm of nowcasting, which focuses on short-term weather predictions. The accurate observation and detection of fog fields are crucial for issuing timely warnings to mitigate these risks.

1.1 Background

Traditional in-situ measurements, such as those from weather stations, provide precise data but are limited in coverage. They offer high accuracy in specific locations but cannot offer a comprehensive view of large geographical areas. This limitation obstructs the ability of meteorologists to detect fog over regions that lack sufficient monitoring equipment. Satellite-based observation, on the other hand, offers extensive coverage and the ability to monitor large areas continuously. This is where the integration of machine learning offers promising potential. Machine learning algorithms can analyze vast amounts of data and have the capability to identify patterns and anomalies within it. Using this technology on data collected by instruments onboard satellites it should be possible to identify the presence of fog if the necessary information exists within the data.

The use of satellite imagery combined with machine learning to identify fog and low altitude clouds has been actively researched in the last years, with several new algorithms developed in 2023 utilizing different model architectures [1]–[3]. In 2022, Li et al. [4] proposed a multimodal self-supervised convolutional neural network for daytime sea fog detection with experimental results showing an F1 score of 96.06%. In line with the model by Li et al., other algorithms developed for fog detection are tailored to specific geographical regions characterized by largely homogeneous ground surfaces. Several algorithms have been developed to detect sea fog, including [1]–[4], which limits their applicability across varied terrains. Additionally, these models often operate within constrained time frames, such as during daylight hours or around dusk and dawn, further narrowing their utility.

In addition to the models designed for sea fog detection, there have been notable efforts to develop algorithms targeting fog over other specific surface types, such as mountainous areas [5] or desert terrain [6]. In 2018, Andersen et al. developed an algorithm for fog and low cloud satellite detection over the Namib desert, showing an overall correctness of classification of 97% [6]. While proving that high prediction accuracies are achievable, these models each face firm limitations akin to those developed for maritime environments. Limitations include geographic specificity, operational time constraints and not being able to distinguish between fog and low clouds. Although progress has been made in developing satellite-based fog detection algorithms, achieving a truly general model capable of handling a variety of terrain types remains an ongoing challenge.

A primary reason why no algorithm like this exists for fog detection is due to the challenges associated with acquiring a dataset of satellite images with accurate ground truth data for ground-level visibility. For starters, such data collection demands extensive ground station infrastructure capable of providing accurate visibility measurements across various locations. Furthermore, labeling satellite image pixels with ground truth data requires precise alignment to ensure that the satellite observations correspond to the correct ground measurements [7]. Additionally, due to the rapid changes that occur in the atmosphere, the process demands exact time synchronization between satellite and ground stations. These factors make the collection and validation of this data difficult, as well as costly and time-consuming.

In late 2023 Pirinen et al. [8] used a synthetic dataset generated using the Radiative Transfer for Television Infrared Observation Satellite Operational Vertical Sounder (RTTOV) [9] to train machine learning models to estimate cloud optical thickness (COT). This dataset was processed using Independent Pixel Approximation (IPA), which treats each pixel independently without reference to its neighbors. When tested on real data, in the form of two publicly available datasets; KappaZeta [10] and a novel one from the Swedish Forest Agency, the models were able to perform satisfactory estimations of the COT of clouds in the satellite images. Considering the challenges associated with obtaining labeled real-world data of this nature, the success of [8] demonstrates the value and applicability of synthetic, simulated datasets in the field of weather nowcasting.

In this thesis, we aim to explore the idea of developing a more general algorithm for satellite fog detection. We train and test a set of different machine learning models on a synthetic dataset and test their performances on real-world data from satellite images. The models will be trained to estimate the ground-level visibility to detect fog located above several different types of ground surfaces. We will discuss our results in comparison with current state-of-the-art methods and identify the trade-off in prediction accuracy when utilizing generalized machine learning models trained on synthetic data, compared to their more specialized counterparts.

The capability to remotely detect fog through the use of satellites represents a meaningful advancement in weather forecasting, particularly for nowcasting. The integra-

tion of machine learning in fog detection could lead to the development of dynamic, real-time monitoring systems. These systems could provide continuous updates on fog conditions, allowing for quicker responses and more effective dissemination of fog warnings to the public and relevant authorities. While the remote detection of fog located above different terrain types remain challenging, the incorporation of AI presents a promising avenue for improvement. By leveraging AI's ability to process and analyze large datasets, meteorologists can enhance fog prediction models, leading to safer navigation for land, sea, and air traffic, and contributing to the overall advancement of nowcasting techniques.

1.2 Dataset

For this thesis, a dataset inspired by that used by Pirinen et al. in [8] was provided by SMHI (Swedish Meteorological and Hydrological Institute) to train the AI models. This dataset includes simulated reflection and brightness temperatures as captured by the Moderate Resolution Imaging Spectroradiometer (MODIS), an instrument onboard the AQUA satellite launched by NASA [11]. The dataset also includes additional information of the surface such as temperature and pressure, along with a description of the specific terrain. The target value for machine learning models is the ground-level visibility. Ground-level visibility is the maximum distance at which an object can be seen clearly along the ground under given atmospheric conditions. It is calculated according to Koschmieder's visibility equation [12].

The dataset was generated using RTTOV [9]. The data is one-dimensional, meaning that each pixel is horizontally homogeneous and varies only in the vertical direction. Each data point represents a single pixel in an image captured by MODIS, completely independent from other pixels. The dataset is therefore processed using IPA and utilized for training machine learning models. When employing the model to an actual MODIS image it will estimate the visibility of each pixel individually. Training a machine learning-based fog detection algorithm on independent pixels represents a novel approach, distinct from more conventional methods that utilize full satellite images. This approach has demonstrated success in applications such as predicting cloud optical thickness [8]. Motivated by these achievements, we are exploring this method for global-scale fog detection. It is important to note that this task presents the major challenge of differentiating between fog and low clouds, which are notoriously difficult to distinguish due to their similar spectral signatures [13].

1.3 Purpose

The primary aim of this thesis is to explore how the synthetic dataset, provided by SMHI, can contribute to real-world fog detection on a global scale. To do this, several different machine learning model architectures for tabular data are tested with a benchmark on their ability to estimate ground-level visibility. The estimations are done in IPA to enable training and testing with the synthetic dataset. Given that

the data is simulated and includes accurate measurements of ground-level visibility for each data point, our objectives are:

1. to detect foggy pixels.
2. to estimate the visibility in those pixels.

Achieving good accuracy in objective 1 can enable quick alerts of fog to ground, sea, and air traffic. Objective 2 is expected to be a non-trivial task. However, this can provide more informative warnings and different classifications of fog, which can further contribute to traffic safety. Ideally, our goal is to develop a general model that can operate for variations in terrain type and time of day. Should we encounter subpar results, it might become necessary to restrict the model's applicability to specific terrains or times.

Furthermore, the thesis work will aim to investigate which features of the data, for example which spectral bands of the MODIS imager, are the most important when it comes to detecting fog, as well as identifying any irrelevant features. Additional features not directly captured by MODIS will also be explored. These include surface temperature and pressure, which can be obtained from ECMWF (European Centre for Medium-range Weather Forecasts) numerical weather prediction models. This investigation will help determine how data from other sources may strengthen model predictions. If time allows it, a post-processing algorithm using spatial information in satellite images should be developed.

1.4 Goals

The goal of this thesis project is to answer the following questions:

- Can the synthetic dataset be used to train machine learning models to detect fog and estimate ground-level visibility using IPA? With competitive accuracy?
- Do the algorithm's results on the synthetic data transfer to real-world data?
- Which features, other than the spectral bands, like surface temperature and pressure, are important for detecting fog?

1.5 Limitations

One limitation of this project is time. What is possible to achieve within the scope of the project is largely ruled by the limited time set out for the project. Because of this limitation, priorities in what avenues to explore have to be made, as not all ideas can be tested within the time frame.

While the premise of the thesis is to use synthetic data as a substitute for real-world data, which is difficult to obtain with a ground truth on a global scale, the scarcity of such annotated real-world data poses a limitation to our ability to validate the real-world applicability of our findings.

One limitation of using satellite data is that all information is gathered from above.

From the perspective of a satellite, fog appears very similar to other types of clouds that are not at ground-level. This poses a possible limitation, that is not faced by in-situ methods such as ground stations, in what is possible to achieve with the satellite data.

2
Theory

In this chapter, the theory of the used models and frameworks is described. This will also include an explanation of the tool used to generate the synthetic data.

2.1 The MODIS instrument

MODIS [14] is a radiometer mounted on NASA's Terra [15] and Aqua [11] satellites. Between the two satellites, MODIS captures the entirety of Earth's surface every one to two days [14]. This capability facilitates studies of global phenomena affecting the land, oceans, and lower atmosphere, for instance, fog. MODIS plays an important role in the development of predictive models for global changes, aiding in the understanding of the Earth's environment.

MODIS records data across 36 spectral bands, ranging from 0.405 to 14.385 µm, which includes visible, near-infrared, and infrared light [14], see Figure 2.1. The instrument captures detailed data with a spatial resolution of 250 meters for bands 1-2, 500 meters for bands 3-7, and 1000 meters for bands 8-36. This, along with the high frequency at which MODIS can capture all of earth's surface, makes it suitable for studying various atmospheric phenomena such as fog, clouds, water vapor, and aerosols [16].

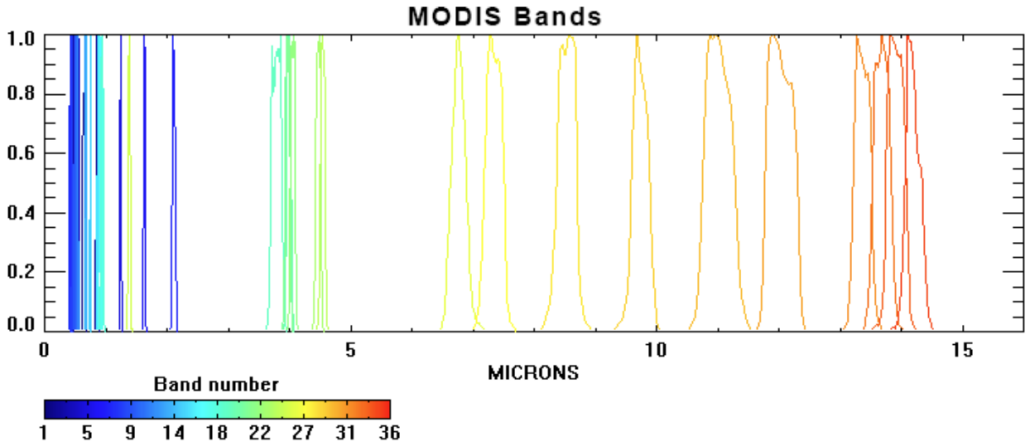

Figure 2.1: The 36 spectral bands recorded by MODIS. Each peak is a band and the y-axis shows wavelength in microns. Illustration by Praveen Kumar.

The top-of-atmosphere (TOA) radiances measured by MODIS are influenced by a variety of atmospheric and surface interactions shown in Figure 2.2. Solar radiation, encompassing ultraviolet, visible light, and some near-infrared wavelengths—collectively known as shortwave radiation—is partially reflected into space by clouds, aerosols, and the Earth's surface. A portion of this incoming solar radiation is absorbed by the Earth's surface, thereby heating it. The Earth re-emits energy in the form of longer-wavelength infrared radiation [17]. Some of the terrestrial radiation, along with a portion of the solar radiation, is reflected back towards the Earth by clouds and aerosols. The atmosphere also absorbs some radiation from both solar and terrestrial sources and re-emits it in both upward and downward directions. Radiation that is not reflected or absorbed directly escapes to the TOA. Shorter wavelengths are scattered more extensively by the atmosphere, whereas longer wavelengths are predominantly absorbed. Given these differences in source and properties of short- and long-wave radiation, MODIS measures infrared radiation through brightness temperatures and shorter wavelengths through reflectivity. This enables thermal properties as well as surface and atmospheric details to be studied.

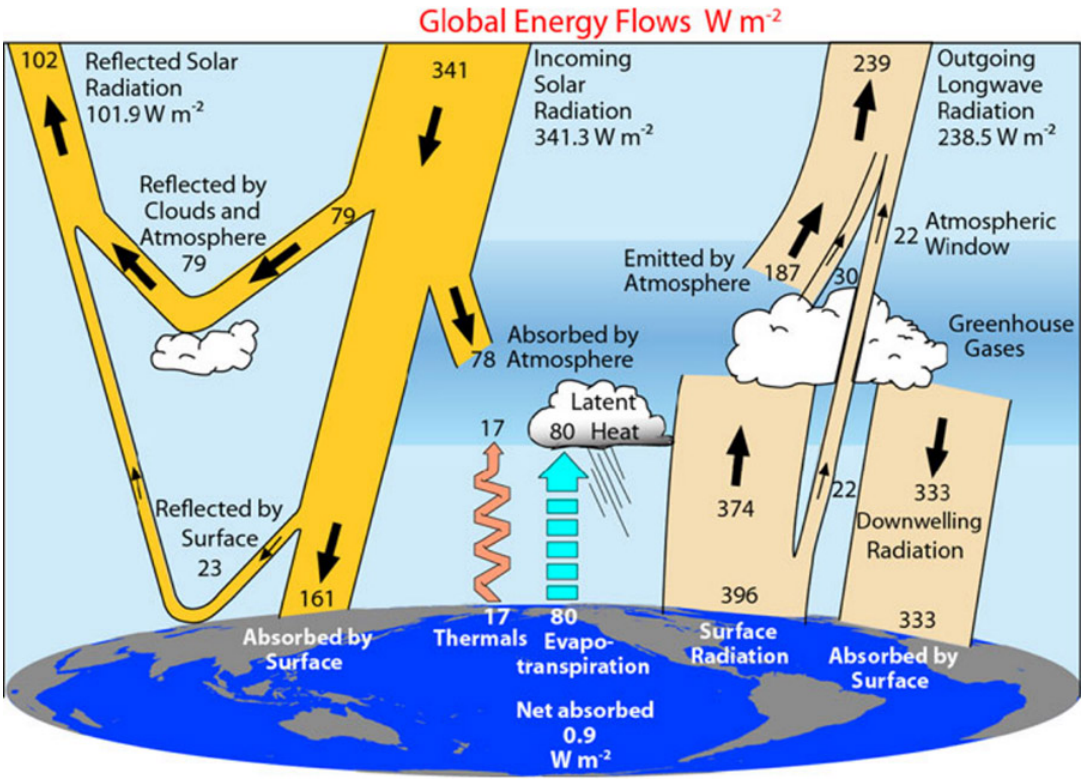

Figure 2.2: Radiation Budget Diagram of Earth's Atmosphere, showing the distribution of solar and terrestrial radiation. This diagram illustrates the balance between incoming energy from the sun and outgoing energy from the Earth. Trenberth et al., 2011 [18].

Geometric factors, such as the satellite zenith angle, sun zenith angle, and azimuth difference angle influence the data captured by MODIS. The satellite zenith angle and sun zenith angle measure the deviation from the vertical above the Earth to the

satellite's line of sight and to the sun's rays respectively, see Figure 2.3. The azimuth difference angle determines the sun's directional position relative to the satellite's viewing direction. Atmospheric scattering and surface reflectance are dependent on these angles. For instance, high zenith angles cause bigger atmospheric interference, which affects the detection accuracy of atmospheric phenomena. MODIS algorithms account for these angles to correct atmospheric effects as best as possible to ensure data accuracy [14].

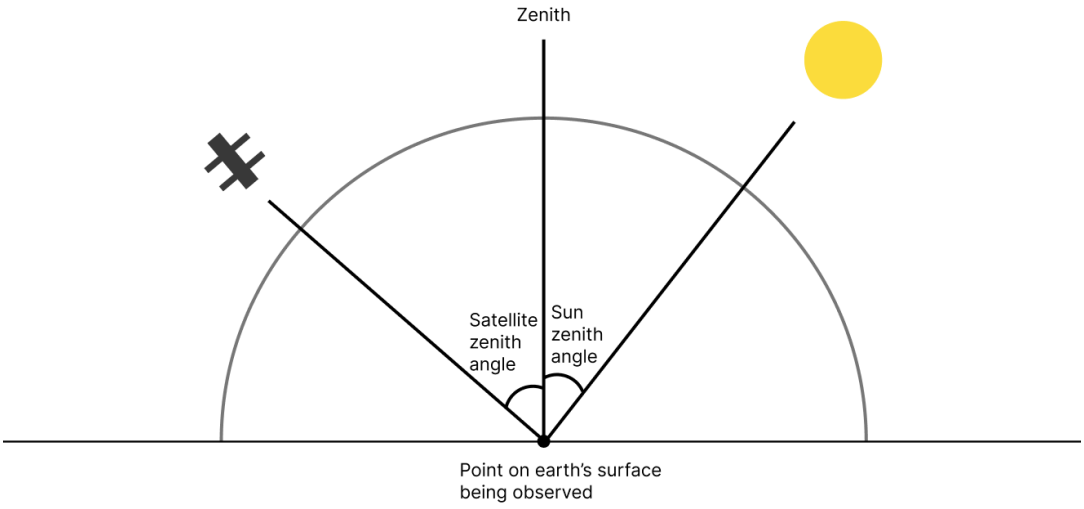

Figure 2.3: A sketch of the satellite zenith angle and sun zenith angle.

2.2 Synthetic data with RTTOV

RTTOV [9] is a fast radiative transfer model used for simulating satellite radiances. Its primary function is the simulation of TOA radiances, as would be observed by a space-borne radiometer. The model's simulations are widely used in numerical weather prediction and satellite data assimilation. To generate a synthetic dataset, RTTOV requires input parameters describing the environmental conditions of the atmosphere and surface. For the fog dataset these include atmospheric profiles, cloud properties, fog characterization and surface properties. By varying the values of the input parameters, RTTOV generates a diverse dataset of simulated TOA radiances. This way a synthetic dataset can contain a wide range of scenarios, including cloud cover of various types, clear sky, and fog or haze, as well as various ground surfaces.

The synthetic data, which mimic the spectral bands of the MODIS instrument and include ground-level visibility as a regression target, provide a suitable dataset for supervised machine learning. The ground-level visibility is calculated using Koschmieder's law [12]:

$$V \approx \frac{3.9}{e} \tag{2.1}$$

which relates visibility (V) to the atmospheric extinction coefficient (e). The constant 3.9 is derived from the logarithmic threshold of human visual contrast detection. Incorporating this equation into our dataset ensures that the synthetic data realistically represent the conditions under which ground-level visibility is measured.

Additionally, the dataset also includes the satellite zenith, sun zenith and azimuth difference angles, which are part of the input given to RTTOV. Although MODIS itself does not detect surface pressure, this dataset includes it along with surface temperature, which can be derived from the spectral bands. This additional information is obtainable from other sources and is hypothesized to be useful for detecting fog, since fog formation is influenced by the surface conditions [19]. A complete list of the features in the dataset is shown in Table 2.1.

Table 2.1: Features in the synthetic dataset.

Feature	**Unit**	**Description**
Brightness Temperature Band 20	K	3.750 microns. Not used.
Brightness Temperature Band 21	K	3.959 microns. Not used.
Brightness Temperature Band 22	K	3.959 microns. Not used.
Brightness Temperature Band 23	K	4.050 microns
Brightness Temperature Band 24	K	4.466 microns
Brightness Temperature Band 25	K	4.516 microns
Brightness Temperature Band 27	K	6.715 microns
Brightness Temperature Band 28	K	7.325 microns
Brightness Temperature Band 29	K	8.550 microns
Brightness Temperature Band 30	K	9.730 microns
Brightness Temperature Band 31	K	11.030 microns

Continued on next page

Table 2.1 continued from previous page

Feature	Unit	Description
Brightness Temperature Band 32	K	12.020 microns
Brightness Temperature Band 33	K	13.335 microns
Brightness Temperature Band 34	K	13.635 microns
Brightness Temperature Band 35	K	13.935 microns
Brightness Temperature Band 36	K	14.235 microns
Reflectivity Band 08	Ratio	0.412 microns. Not used.
Reflectivity Band 09	Ratio	0.443 microns. Not used.
Reflectivity Band 10	Ratio	0.488 microns. Not used.
Reflectivity Band 11	Ratio	0.531 microns
Reflectivity Band 12	Ratio	0.551 microns
Reflectivity Band 13	Ratio	0.667 microns
Reflectivity Band 14	Ratio	0.678 microns
Reflectivity Band 15	Ratio	0.748 microns
Reflectivity Band 16	Ratio	0.870 microns
Reflectivity Band 17	Ratio	0.905 microns
Reflectivity Band 18	Ratio	0.936 microns
Reflectivity Band 19	Ratio	0.940 microns
Reflectivity Band 26	Ratio	1.375 microns
Visibility	m	The ground-level visibility.
Satellite Zenith Angle	°	Ranges from 0 to 15.
Sun Zenith Angle	°	Ranges from 0 to 90.
Azimuth Difference Angle	°	Ranges from 0 to 180.
Surface Temperature	K	Temperature of the surface.
Surface Pressure	hPa	Atmospheric pressure at ground-level.
Cloud Optical Thickness	Dimensionless	Indicates the density of a cloud, measured by its ability to block sunlight.

Continued on next page

Table 2.1 continued from previous page

Feature	Unit	Description
Cloud Type	Dimensionless	Represented with integers as following: 1 Stratus Continental, 2 Stratus Maritime, 3 Cumulus Continental Clean, 4 Cumulus Continental Polluted, 5 Cumulus Maritime, 6 Cirrus, >6 mixed clouds (Cirrus and Water cloud Type). Mixed clouds are not used for any evaluations in this thesis.
Profile ID	Dimensionless	Integer identifier for different condition profiles.
Gas Optical Thickness	Dimensionless	Indicates the amount of light blocked by atmospheric gases.
Vertically integrated water vapor	g/cm^2	Indicates the total amount of water vapor contained in a column of atmosphere from the surface up to the TOA.
Surface description	Dimensionless	Three word description of the surface

Simulations with RTTOV [9] are grounded in the fundamental principles of radiative transfer, providing a physically realistic basis for the dataset. Using the input parameters RTTOV calculates the top of atmosphere radiance that satellite instruments measure. It starts by determining the extinction coefficient and the source function with the input parameters. The extinction coefficient is a measure of the degree to which absorption and scattering weakens the radiance, and oppositely the source function is a measure of how emission and scattering contribute to a stronger radiance [20]. In the next step RTTOV [9] incorporates the extinction coefficient and the source function, along with other input parameters, to solve the radiative transfer equation. This equation is used to calculate the radiances at TOA, taking into account the different aspects shown in the radiation budget (Figure 2.2). Finally, it convolves the radiance at each wavelength according to a response function specific to the satellite which is being simulated. The dataset used in this thesis simulates the MODIS instrument on the AQUA satellite [11].

Due to RTTOV not simulating aerosols—small particles suspended in the atmosphere—MODIS bands capturing wavelengths below 500 nm, which are sensitive to these particles, were not used to train any of the machine learning models. Infrared bands capturing wavelengths below 4 microns were also excluded, as they don't capture reflected solar light in the simulation, opposed to in reality. Even though RTTOV is able to simulate mixed clouds or different cloud types stacked on top of each other, these complex scenarios were also disregarded to simplify the task of detecting fog.

As mentioned, each data point in the synthetic dataset represents a single pixel in an image captured by MODIS. Table 2.2 shows the number of data points for each cloud type, including the number of data points corresponding to fog. Here

fog is defined as clouds touching the ground causing ground-level visibility to be less than 1000 meters [21]. The table also shows the percentage of foggy data points for each cloud type. Figures 2.4 and 2.5 compare data points related to fog and fog-free conditions for the different cloud types present in the dataset. These figures display the range of values and the mean for each of the MODIS spectral bands. By examining the mean values, a slight difference between fog and fog-free data points is evident in the majority of the bands. However, the range of values shows significant overlap between the two conditions. This overlap suggests that machine learning models trained to detect fog using spectral information would need to rely on more complex relationships between the MODIS bands to discern whether fog is present or not.

Table 2.2: Number of data points for different cloud types within the synthetic dataset.

Cloud types	**Nr. of datapoints**	**Nr. of fog datapoints (ground-level visibility < 1000 m)**	**Fog percentage (%)**
Clear	150,000	0	0.00
Stratus Continental	30,073	4,706	15.65
Stratus Maritime	29,898	4,222	14.12
Cumulus Continental Clean	29,766	4,916	16.52
Cumulus Continental Polluted	30,088	5,234	17.40
Cumulus Maritime	30,175	4,012	13.30
Cirrus	149,970	18,396	12.27
All cloud types	**449,970**	**41,486**	**9.22**

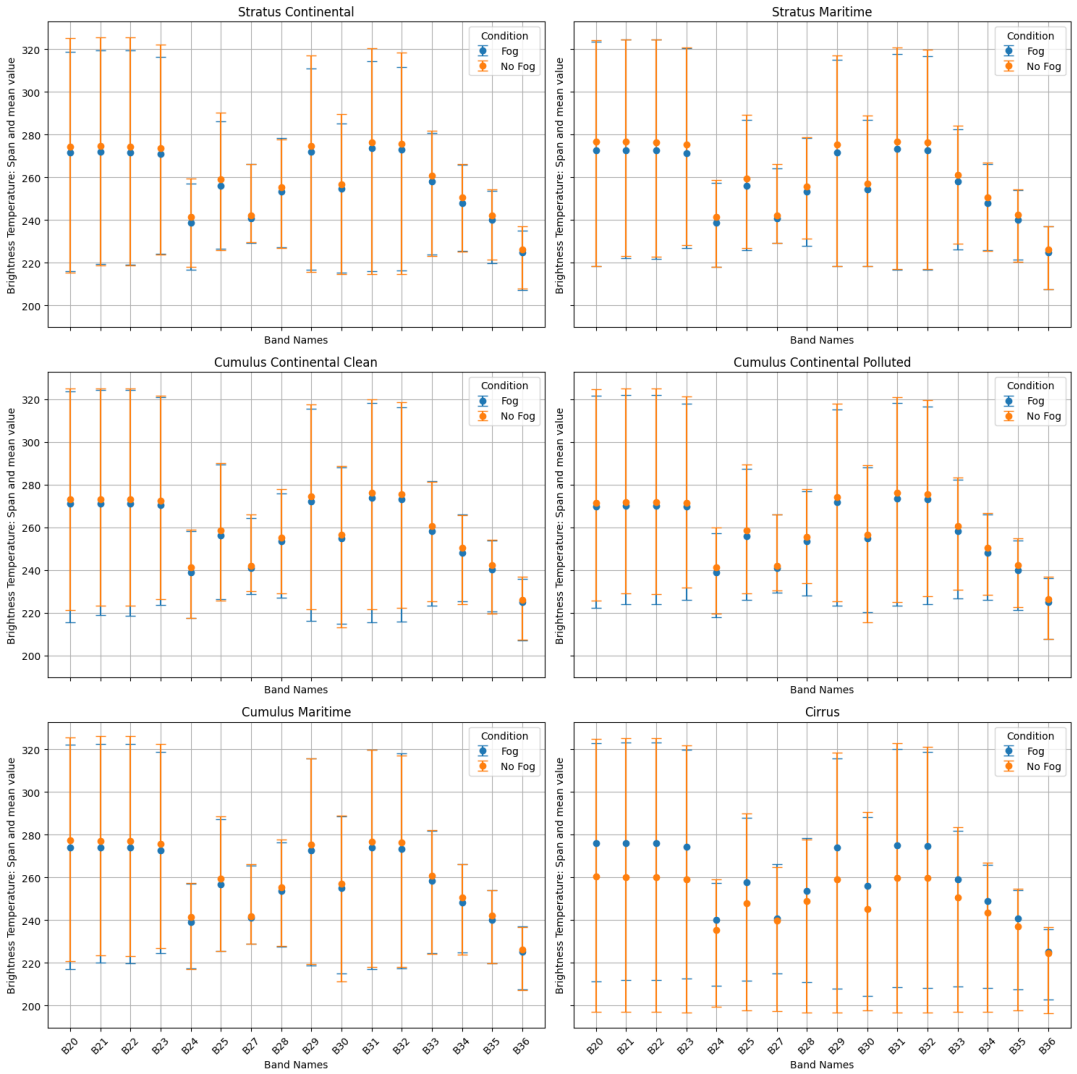

Figure 2.4: Six plots illustrating the brightness temperatures recorded in the dataset for each cloud type. For each band (20-36, excluding 26), the orange dot represents the mean brightness temperature without fog, with error bars indicating the range between the highest and lowest values. Similarly, the blue dot and error bars represent the data points with fog.

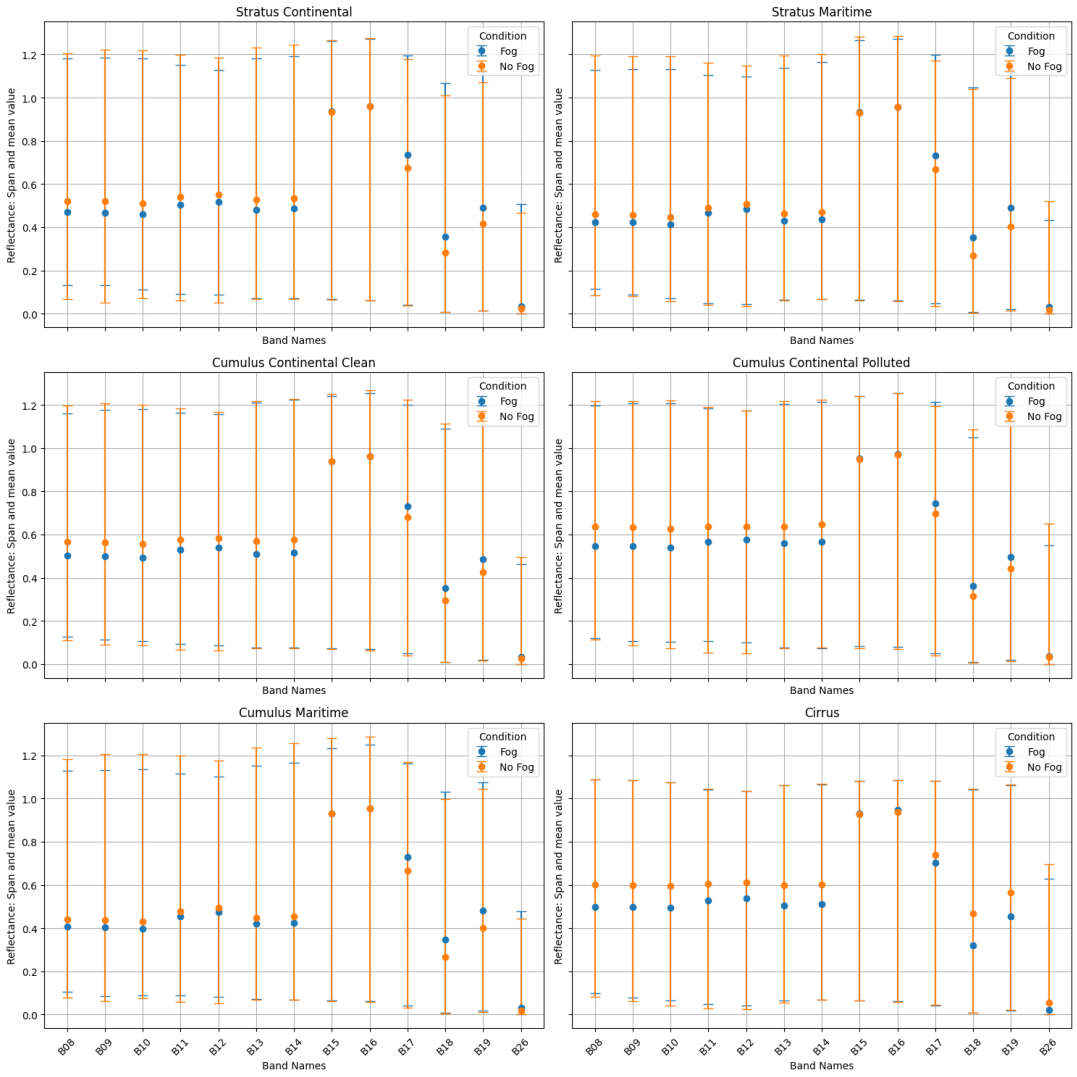

Figure 2.5: Six plots illustrating the reflectance recorded in the dataset for each cloud type. For each band (8-19 and 26), the orange dot represents the mean brightness temperature without fog, with error bars indicating the range between the highest and lowest values. Similarly, the blue dot and error bars represent the data points with fog.

2.3 Machine learning models

Given that the synthetic dataset includes a target column, being the ground-level visibility, the machine learning models can be trained using supervised learning. Supervised learning is a type of machine learning where the model is trained on a labeled dataset. This means that for each input data point, the correct output (target) is provided, allowing the model to learn the mapping from inputs to outputs. In this case, the input data includes various features, and the target is the ground-level visibility, as described in Section 2.2.

This is a leading advantage of using a synthetic dataset, as acquiring a large la-

beled dataset with real data is very difficult for fog detection. Supervised learning is generally superior to unsupervised learning for tasks where there is a clear target. Unsupervised learning deals with unlabeled data and tries to find hidden patterns in the input data without any target variable to predict [22]. Supervised learning is well-suited for classification and regression tasks, where the goal is to predict discrete categories or continuous values, respectively.

In this section, we explain the machine learning models that are most important for this thesis. Their relevance will become clear in the results chapter.

2.3.1 Decision trees and gradient boosting decision trees

Modeled on a structure of sequential questions, decision trees offer a straightforward machine learning approach. The tree is organized hierarchically, starting from a root node, which represents the entire dataset. Each internal node represents a decision based on the value of a specific feature, leading to two or more branches. These branches split the data into progressively smaller groups. This process continues until it reaches the leaf nodes, which represent the final predictions or outcomes [23].

The operations used for making decisions at each node involve selecting a feature and a threshold that best splits the data according to a certain criterion, typically aiming to maximize the separation of the target variable. Common criteria include Gini impurity and information gain for classification tasks [24], or mean squared error for regression tasks. The choice of these criteria and thresholds determines how well the tree performs in making predictions.

Gradient Boosting Decision Trees (GBDTs) [25] are powerful ensemble models built upon the foundation of decision trees. GBDTs start with a single weak decision tree, often referred to as a base learner. Iteratively, new trees are added to the ensemble to correct the errors made by the previous trees. Each new tree is trained to predict the residual errors of the combined ensemble of trees that came before it. This iterative process continues until a set number of trees are added or until a performance criterion is met. Figure 2.6 illustrates the structure of GBDTs.

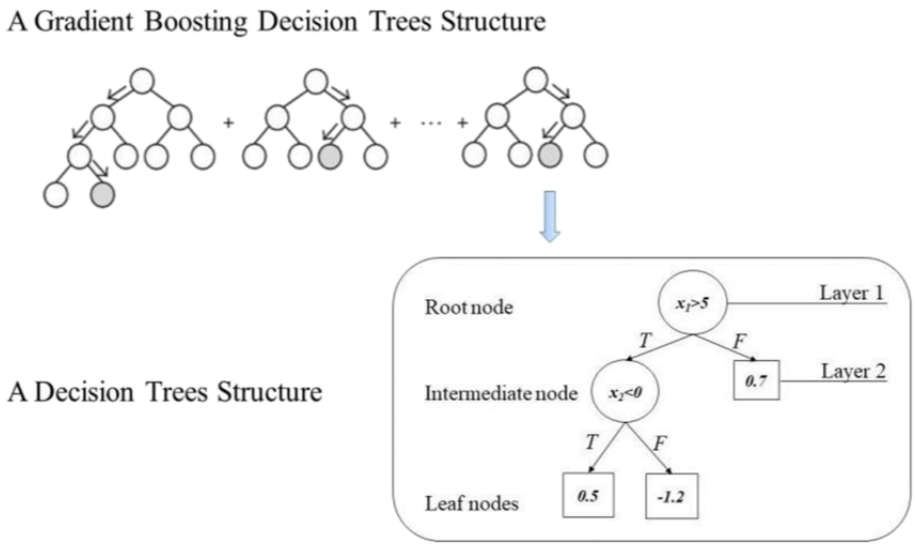

Figure 2.6: A diagram showing the structure of Gradient Boosting Decision Trees [26].

The training of GBDTs uses gradient descent, a method for minimizing the loss function of the model. In this context, the loss function measures how well the model's predictions match the actual outcomes. Gradient descent optimizes the model's performance by adjusting the parameters, in this case, the decision rules in the trees, to minimize this loss. Each new tree in the ensemble is trained on the negative gradient of the loss function with respect to the model's predictions, hence the name "gradient boosting."

GBDTs are commonly used and can achieve high accuracy on diverse datasets. One of the most efficient and popular implementations of gradient boosting is a model called Extreme Gradient Boosting (XGBoost) [27]. XGBoost enhances the performance of traditional GBDTs with additional techniques such as regularization and parallel processing, which help prevent overfitting and to speed up computation respectively.

2.3.2 Deep neural networks

Neural networks are a powerful type of machine learning model inspired by the structure of the human brain. Neural networks consist of interconnected layers of artificial neurons, which process data and learn from it [28]. A deep neural network (DNN) is a neural network with two or more hidden layers of neurons. Figure 2.7 illustrates the structure of a DNN.

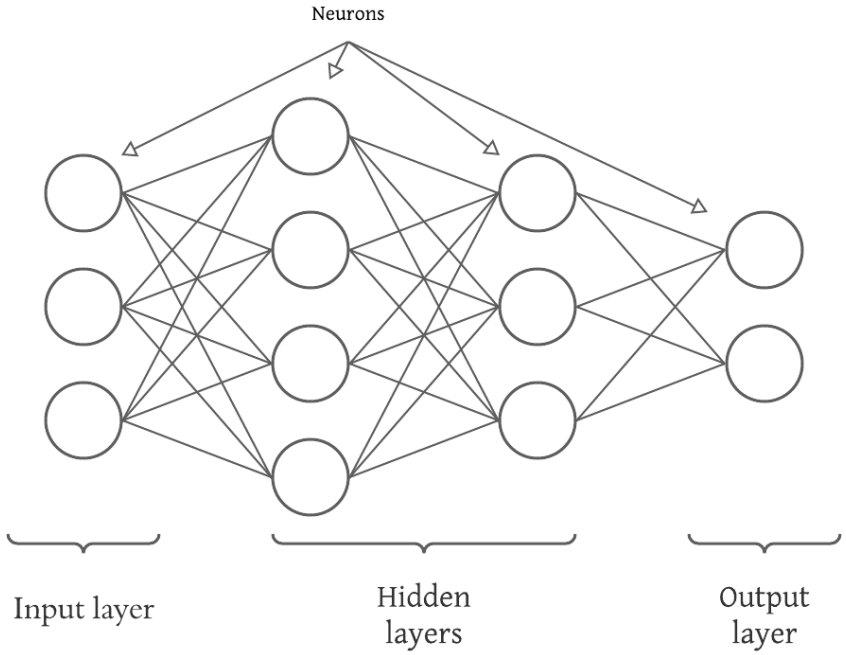

Figure 2.7: A diagram showing the structure of a Deep Neural Network. Each neuron is represented as a circle while the weighted connections are depicted as lines between the neurons.

Each neuron in a DNN receives weighted inputs from the neurons in the previous layer. The neuron then applies an activation function to this weighted sum to introduce non-linearity, enabling the network to learn complex patterns. The output of this activation function is then passed as input to the neurons in the next layer. This process allows DNNs to model intricate, non-linear relationships within the data.

One of the key components of a DNN is its hidden layers. These layers are situated between the input and output layers and consist of several neurons that transform the input data through a series of computations. The term "deep" in deep neural networks refers to the presence of multiple hidden layers, which enable the network to learn multiple levels of abstraction from the data. DNNs are highly effective at capturing complex data relationships and can handle high-dimensional data, making them usable in many applications.

2.3.3 Quantile regression neural network

Quantile Regression Neural Network (QRNN) is an extension of neural networks that predicts specific quantiles of the target distribution, rather than the mean or a single value [29]. This approach is particularly useful for understanding the uncertainty and distribution of the target variable, rather than just predicting a point estimate.

Typically in regression tasks, models are trained to minimize the error, often the mean squared error (MSE). MSE is a good overall measure of performance and it primarily focuses on reducing the average prediction error across all cases. However, MSE does not provide information about the variability in prediction errors for individual cases or the spread of the target values. In other words, it does not address case-specific errors effectively, where some predictions might have higher uncertainty or variability.

QRNNs address this limitation by predicting quantiles, such as the median (50th percentile) and upper (e.g., 90th percentile) quantiles [30]. By doing so, QRNNs handle case-specific errors more comprehensively. The quantile regression loss function is different from the MSE loss. For a given quantile q, the loss function L_q is defined as:

$$L_q = \sum_{i=1}^{n} \max(q(y_i - \hat{y}_i), (q-1)(\hat{y}_i - y_i)) \tag{2.2}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted quantile, and n is the number of data points. Depending on the quantile, this loss function punishes overestimates and underestimates differently, allowing the model to prioritize different parts of the target distribution.

3
Methods

In this chapter, the methods used for developing and evaluating machine learning models are explained. It is detailed how experiments with various models and pre-processing techniques are conducted using synthetic data. Finally, it is explained how a model was tested on a real satellite image captured by MODIS.

3.1 Python testing environments in Jupyter Notebook

To facilitate the training, testing, and comparison of various machine learning models, several Python testing environments are established using Jupyter Notebook. These notebooks offer the flexibility to execute code across discrete cells, with the potential for each one to handle specific tasks such as data loading, model training, or result visualization through plots. Libraries such as Pandas and Numpy are used for data manipulation and Matplotlib for graphical outputs. Additionally, Jupyter's support for Markdown enables the inclusion of descriptive text cells, simplifying the organization and documentation of the research process. This structured approach not only aids in maintaining clarity but also with reproducibility and ease of collaboration. Since the notebooks are stored in a Virtual Machine (VM) rather than locally, using Git for collaboration is unnecessary. However, Git was still used for backup purposes to safeguard against unexpected VM shutdowns.

3.2 Data loading and pre-processing

The provided dataset is segmented into several files based on the content of features and types of cloud coverage, including clear skies, ice clouds, water clouds, and mixed clouds. Further subdivision of these files is based on the wavelength of the MODIS channels, specifically into groups of infrared bands and solar bands. The initial step in the data processing involves aggregating the information scattered across different files. Each data point is represented in separate files based on their spectral characteristics—one for infrared bands and another for solar bands. Through the use of NumPy, these are consolidated to ensure that all features of a single data point are collocated. This process creates a complete NumPy array where each data point has all its features, making it easier for analysis and modeling.

By implementing variables and Boolean flags, we were able to change these set-

tings in various combinations, effectively filtering the data points and features for use in different machine learning models. This modular approach made it easy to explore various preprocessing strategies efficiently, enabling precise selection of data points and features tailored to specific needs. Consequently, this allowed for the creation of an extensive set of models, each trained on carefully chosen subsets of data, such as specific cloud types or particular times of the day.

After selecting the appropriate data points and their features, we proceed to prepare the data for model training by scaling the features using the StandardScaler from scikit-learn. This tool adjusts the features so that each has a mean of 0 and a standard deviation of 1 as:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma} \tag{3.1}$$

where:

- X is the original feature value,
- μ is the mean of the feature values,
- σ is the standard deviation of the feature values.

Before scaling, we calculate the noise standard deviation for each feature, defined as the ratio of the range (maximum value minus minimum value) to the standard deviation of the feature. The noise standard deviation for a feature X is calculated as:

$$\sigma_{\text{noise},X} = \lambda \cdot \frac{\max(X) - \min(X)}{\sigma} \tag{3.2}$$

where:

- λ is the noise level variable,
- $\max(X)$ is the maximum value of the feature values,
- $\min(X)$ is the minimum value of the feature values,
- σ is the standard deviation of the feature values.

Subsequently, the noise standard deviation is used to add Gaussian noise to the scaled data points. The noise level variable λ can be adjusted to change the amount of noise added to the data. This process enhance model robustness, ensuring that the models are less sensitive to nuances specific to the training data while retaining information potentially lost during scaling. Due to initial results indicating that the models are very sensitive to noise, a low noise level of $\lambda = 0.01$ is used.

3.3 Comparing and tuning different machine learning models

Once the dataset is configured, it serves as the foundation for training a diverse array of machine learning models, enabling a fair comparison of their performances.

This process begins by dividing the dataset into three subsets: a training set comprising 90% of the data, a validation set comprising 5%, and a test set comprising the remaining 5%. Different models are trained using the same training set and evaluated using the same test set, ensuring consistency in performance comparison. The validation set is specifically used for training DNN models to prevent overfitting. It is important to note that while the training set can be modified to explore different outcomes, the test set remains a consistent representation of the original dataset.

Given the extensive variety of models tested—each with numerous parameter settings—the efficiency of tracking every performance metric is compromised. Typically, only the results that surpass previous benchmarks are retained. While it might be practical to document all outcomes when comparing a limited number of models, the broad scope of this study, aimed at investigating the potential applications of synthetic data, benefits from a selective approach to performance tracking. Additionally, the Python testing environment enhances the reproducibility of our experiments, allowing us to easily recreate results without the immediate need to document every outcome.

Similar to the comparison of different machine learning models, different parameter settings of the same model were also compared. Generally, we began by training and testing a model with parameters indicative of high complexity. For neural networks, this meant using many layers and a high number of neurons per layer. For tree-based models like XGBoost, this involved setting the maximal tree depth, corresponding to the tree depth of new decision trees added during training. These parameters were then incrementally increased or decreased until a noticeable drop in performance was observed. By retraining and comparing various architectures within the same model type, we could manually tune the models to identify suitable parameters.

3.4 Evaluation metrics

To evaluate the performance of regression models, root mean squared error (RMSE) is used. This is a metric that measures the average magnitude of the errors between predicted and observed values. RMSE is calculated by taking the square root of the average squared differences between the predicted and actual values (3.3). Lower RMSE values indicate better model performance, as they represent smaller errors between predicted and observed data.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.3}$$

To evaluate the performance of the classification models, several common metrics are used: F1-score, accuracy, recall, and precision. Each of these metrics provides insights into different aspects of the models' performances. All four metrics have a maximum possible score of 1 and a minimum of 0 with higher values being better. Accuracy measures the proportion of correct predictions, true positives, and true

negatives, among all examined cases (3.4). Precision refers to the proportion of true positive results in the set of positive predictions (3.5), effectively measuring the accuracy of positive predictions. Recall assesses the model's ability to identify all positive instances and is calculated as the ratio of true positives to the sum of true positives and false negatives (3.6). F1-score is calculated as the harmonic mean of precision and recall (3.7), to provide a balance between the two. Collectively, these metrics give a comprehensive view of classifier effectiveness.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Population (all cases)}} \tag{3.4}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.5}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.6}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.7}$$

3.5 Exploring regression models on synthetic data

Since our dataset lacked spatial or temporal information, our strategy focused on exploring a range of machine learning models tailored for tabular data analysis. These include tree-based models such as random forest and GBDTs, which are readily accessible via Sklearn, as well as feed-forward neural networks leveraging Keras and Tensorflow. Additionally, we considered publicly available models like NODE and TabTransformer. Early on, it became evident that feed-forward neural networks and GBDTs were up to par with, or often outperforming, the other model alternatives. Consequently, these models were predominantly utilized in subsequent tests. Moreover, due to the relatively short training time required for XGBoost, it emerged as the primary choice for many experiments.

The aforementioned regression models were trained to estimate the ground-level visibility for each synthetic data point using spectral features, along with pressure and temperature. Figure 3.1 shows a histogram of the ground-level visibility frequencies for the synthetic data points, which served as the target values for the regression models. These target values were scaled in the same manner as the features, as described in Equation 3.1.

As can be seen in Figure 3.1 the distribution of target values exhibits an atypical pattern with two distinct clusters identifiable. These clusters, termed the low visibility and high visibility clusters, are separated by a large span where very few data points reside. By excluding all data points in the high visibility cluster, we were able scale the target values differently using log-transforming, resulting in a more typical distribution. Figure 3.2 illustrates the distribution of target values before and after applying log-transforming. This enables the training of regression models solely on data points in the low visibility cluster, with the distribution of target values being less left-skewed.

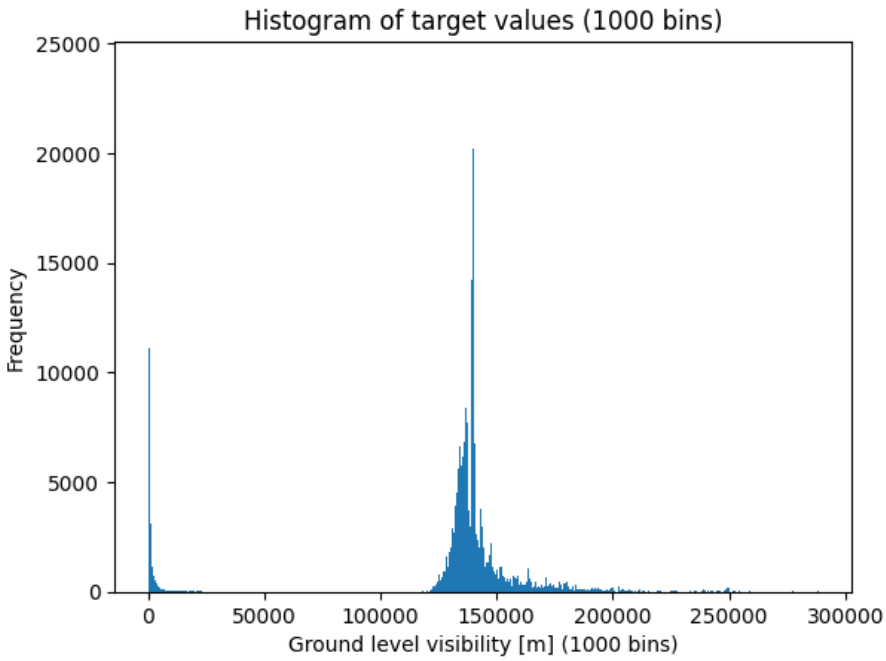

Figure 3.1: Histogram of the ground-level visibility of the data.

3.6 Exploring classification models on synthetic data

In order to classify data points as either low or high ground-level visibility, we needed to establish a threshold value that would delineate between the two clusters shown in Figure 3.1. This threshold was termed the low visibility threshold and could be modified to alter how data points were labeled for training the classification models. By setting the threshold to 1000 m the models can be trained to perform precise fog/no fog classification. Once the data points were assigned class labels, where the positive label 1 corresponds to fog and the negative label 0 corresponds to no fog, the training and testing of different classification models could begin. Notably, several of the models previously tested for regression could be readily adapted for this classification task by change of loss function.

Due to the high prevalence of high visibility data, constituting approximately 90% of the dataset, machine learning models could attain exceptionally high accuracy by simply classifying all data points as belonging to the high visibility class. To counteract this, we introduced an additional variable representing the fraction of low visibility data points to be included in the training set. This allowed us to undersample the high visibility data points, effectively oversampling the low visibility data points, since foggy data constituted the minority class. Undersampling is performed by randomly selecting the desired amount of data points from the high visibility class. By equalizing the class sizes, classification models essentially had to discern the distinctions between the two clusters to achieve high accuracy.

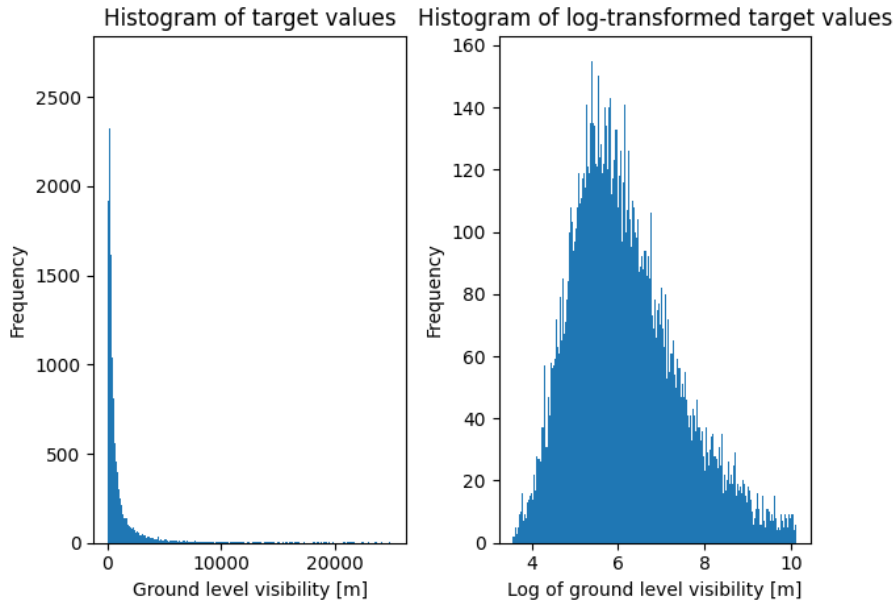

Figure 3.2: Histogram of the ground-level visibility of the low visibility data, along with the histogram of log-transformed ground-level visibility for the same data.

3.7 Classifying pixels in real MODIS images

Due to the lack of publicly available labeled data suitable for our models, evaluating them on real MODIS images proved to be a significant challenge. However, from a talk regarding NWCSAF cloud products by SMHI [31], we learned of a specific date and time that had been used to evaluate other models for similar tasks. Specifically, Denmark and southern Sweden were foggy throughout the day on December 18, 2021. Finding images captured by MODIS on board the AQUA satellite over that region could indicate whether the models trained on synthetic data would be able to detect fog in real satellite images.

To obtain this specific satellite data, we searched through the MODIS archive on NASA's Earthdata website and identified the images taken on the specified date. By filtering the images based on longitude and latitude, we were able to find a MODIS dataset, corresponding to an image that had captured all of Denmark on December 18, 2021. The raw satellite data then had to be converted into the same format as the synthetic data, which the models had been trained on, in order to utilize the classification models on each pixel in the image independently.

The MODIS dataset is in HDF-EOS format and can be accessed using the Python package pyhdf. The spectral values for the various MODIS bands are stored as integers and need to be scaled and offset using provided values. The solar bands are scaled directly to reflectance. However, for bands 20 to 36 (excluding band 26), the radiance must be converted to brightness temperature using an inverse formulation of the Planck Function [32]. The specific code and MODIS coefficients used were provided by SMHI. The equation can be described as:

$$\mathrm{BT}_i = \frac{\frac{h \cdot c}{k} / \left(\frac{1}{\mathrm{CWN}_i} \cdot \ln \left(\frac{2hc^2}{\mathrm{RAD}_i \cdot \mathrm{CWN}_i^5 \cdot 10^6} + 1 \right) \right) - \mathrm{TCI}_i}{\mathrm{TCS}_i} \tag{3.8}$$

where:

- BT_i is the brightness temperature for band i.
- h is the Planck constant ($6.6260755 \times 10^{-34}$ Joule $\cdot$ second).
- c is the speed of light in vacuum (2.9979246×10^8 meters per second).
- k is the Boltzmann constant (1.380658×10^{-23} Joules per Kelvin).
- CWN_i is the effective central wavenumber for band i (in cm^{-1}).
- RAD_i is the radiance for band i.
- TCI_i is the temperature correction intercept for band i.
- TCS_i is the temperature correction slope for band i.

The solar and satellite zenith and azimuth angles did not have the same resolution as the pixels captured by the MODIS bands. Instead, each angle corresponded to a 5x5 area of pixels. Therefore, the same angles were applied to the entire 5x5 area of corresponding pixels. Since the synthetic data uses the azimuth difference angle, it was calculated for each pixel as:

$$\Delta\phi = |\phi_{\mathrm{solar}} - \phi_{\mathrm{satellite}}| \tag{3.9}$$

where:

- $\Delta\phi$ is the azimuth difference angle.
- ϕ_{solar} is the solar azimuth angle.
- $\phi_{\mathrm{satellite}}$ is the satellite azimuth angle.

The sun and satellite azimuth angles are the differences between the north direction and the projections of the sun's and the satellite's respective positions onto the horizontal plane, measured clockwise.

As it had emerged as the primary model for the synthetic data, XGBoost was also used for classifying the real MODIS image. Since the MODIS dataset does not include any information regarding pressure and temperature, these features were excluded during the training of this model.

4
Results

This chapter presents the findings from the various experiments conducted to explore fog detection within synthetic MODIS data. Each section examines how different elements influence model accuracy in fog detection. We also delve into the importance of specific data features, identifying which are most and least meaningful for model performance. The evaluation not only highlights the effectiveness of each approach but also sets the stage for discussing decisions, priorities, and possible improvements for future research in fog monitoring using satellite imagery.

Most results in this chapter are based on an average over five runs. This means that a model is trained and tested five separate times with different folds of the dataset and new noise each time, after which the mean of all runs is calculated for each metric.

4.1 Model architectures and parameters

In this section, we present the general architectures and parameters of the DNNs implemented using Keras, as well as the XGBoost models, highlighting the key differences required for performing either regression or classification tasks. These models were fine-tuned by experimenting with various parameter values and comparing the resulting performance. Through this process, we observed that the models converge towards similar solutions within a specific range of parameter values, leading to largely identical results. Minor adjustments to individual parameters resulted in negligible differences on separate test sets. Therefore, we adopted similar model architectures for both classification and regression tasks. For the classification models, a threshold of 0.5 is used for the predictions, with values above 0.5 classified as 1 and values below 0.5 classified as 0.

DNN model parameters

- **Input layer (regression and classification):**
  - **Dense Layer:** The input layer is a dense (fully connected) layer with the number of neurons equal to the length of the feature set. The activation function used is ReLU.
- **Hidden layers (regression and classification):**

  - **Dense layer:** Each hidden layer is a dense layer with 64 neurons and ReLU activation, using L2 regularization with a coefficient of 0.001.
  - **Dropout layer:** In between the hidden layers there is a dropout layer with a dropout rate of 0.2, which helps in regularization.
  - This pattern is repeated for a total of four dense layers and three dropout layers.
- **Output Layer (regression):**
  - **Dense layer:** The output layer is a dense layer with a single neuron, which provides a continuous value as the output, suitable for regression tasks.
- **Output layer (classification):**
  - **Dense layer:** The output layer is a dense layer with a single neuron. The activation function used is sigmoid, suitable for binary classification tasks.

The model was compiled with the following settings:

- **Optimizer (regression and classification):** Adam optimizer, which is a popular choice due to its adaptive learning rate capabilities.
- **Loss function (regression):** Mean Squared Error (MSE), which is commonly used for regression tasks as it measures the average of the squares of the errors between predicted and actual values.
- **Loss function (classification):** Binary Cross-entropy, which measures the performance of a classification model whose output is a probability value between 0 and 1. Binary Focal Cross-entropy, or focal loss, which applies a modulating factor to the cross-entropy loss, was also used in specific cases to achieve different results.

XGBoost model parameters

- **Max depth (regression and classification):** `20`
  - This parameter defines the maximum depth of the individual trees. A deeper tree can model more complex patterns in the data but may also lead to overfitting. Here, the maximum depth is set to 20.
- **Learning rate (regression and classification):** `0.05`
  - The learning rate (also known as `eta`) controls the step size at each iteration while moving toward a minimum of the loss function. A lower learning rate makes the model more robust by shrinking the weights, preventing overfitting. In this model, the learning rate is set to 0.05.
- **Number of rounds (regression and classification):** `1000`
  - This parameter specifies the number of boosting rounds, which is the number of trees that will be built. In this model, 1000 boosting rounds are used to iteratively improve the model performance.
- **Subsample (regression and classification):** `0.8`
  - This parameter specifies the fraction of the training data to be randomly sampled for each tree. Subsampling helps to prevent overfitting and im-

prove generalization. In this case, 80% of the training data is used for each tree.

- **Objective (regression):** `reg:squarederror`
  - This parameter specifies the learning task and the corresponding learning objective. In this case, `reg:squarederror` is used for regression, where the model aims to minimize the squared difference between the predicted and actual values.
- **Objective (classification):** `binary:logistic`
  - `binary:logistic` is used for binary classification, where the output is a probability that a given input belongs to the positive class.

The following parameters for XGBoost were also used in specific cases for classification to achieve different results:

- **Objective:** `binary:logitraw`
  - `binary:logitraw` is used for binary classification, where the output is the raw untransformed score before applying the logistic function.
- **Scale pos weight:** `n_negative / n_positive`
  - This parameter is used to control the balance of positive and negative weights, particularly useful for imbalanced datasets. The ratio `n_negative / n_positive` is used to adjust the balance between the positive (fog) and negative (no fog) classes.

4.2 Regression models

In Table 4.1 the RMSE on the full test dataset is shown for each model. The table also includes the RMSEs on only the low visibility and only the high visibility datapoints from the same training. This is to distinguish between the models' performances on low and high visibility cases respectively. This split is made at a visibility of 25,000 meters because, as can be seen in Figure 3.1, there are no data points in this range and it is at a good point between the two clusters.

The results are detailed in Table 4.1, and they show that the DNN model slightly outperforms the XGBoost model across two scenarios. However, both models exhibit poor performance overall, indicated by high RMSE values. When the dataset is oversampled to a 0.45 fog ratio, the RMSE is higher for both models compared to using the dataset without oversampling. DNN achieves an RMSE of 51,509 for all test data, with 52,998 for low visibility, and 51,292 for high visibility. In comparison, XGBoost performs slightly worse on all metrics. Interestingly, when the data is not oversampled, meaning a fog ratio of 0.0922, both models show lower RMSE for all test data, and much lower in high visibility cases specifically. For DNN this means an RMSE value of 39,321 for all test data, 96,164 for low visibility, and 21,554 for high visibility. The XGBoost model shows similar trends and is again slightly worse than DNN on all metrics.

Table 4.1: The performance of different regression algorithms on all clouds. The results of each model are averaged over five runs.

Model	RMSE (all) [m]	RMSE (low visibility) [m]	RMSE (high visibility) [m]
DNN no oversampling (0.0922 fog ratio)	39,321	96,164	21,554
XGBoost no oversampling (0.0922 fog ratio)	40,239	96,546	23,574
DNN oversampled (0.45 fog ratio)	51,509	52,998	51,292
XGBoost oversampled (0.45 fog ratio)	53,596	57,227	53,074

Figure 4.1 illustrates the predictions of XGBoost on the y-axis against the true values on the x-axis, with no oversampling. The model demonstrates some capability in estimating the target values. Notably, an upper limit for the target values is discernible for data points with ground-level visibility exceeding 100,000. The predicted values for low visibility data points, the cluster on the left in Figure 4.1, are on average much further from the actual values. Figure 4.2 shows the predictions of XGBoost on a dataset containing only low visibility data, below 25,000 meters. The target values for this run are log-transformed to better fit normal distribution. In this plot, it is clear that the model trained and tested only on low visibility data is unable to fit its target.

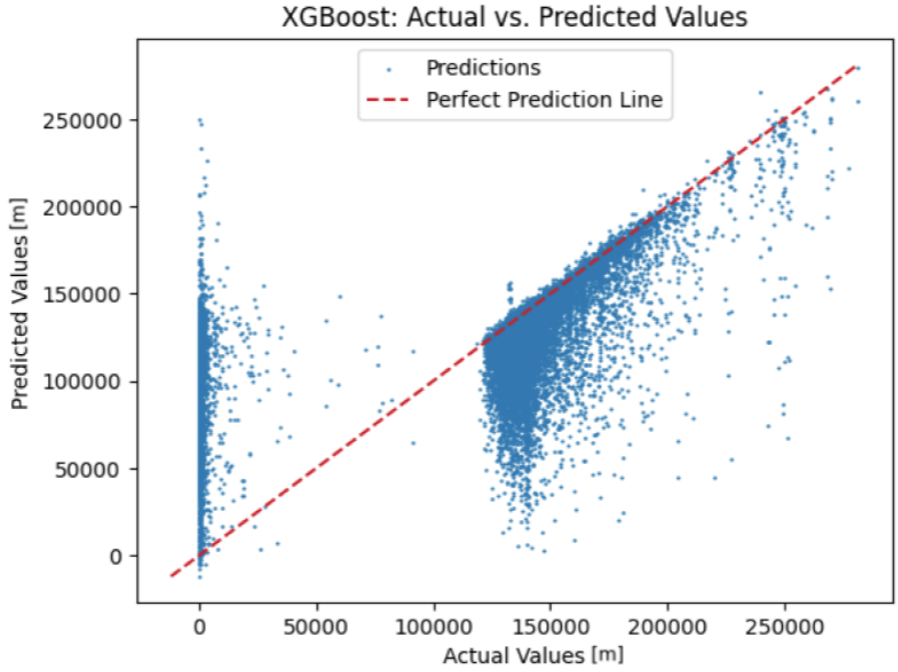

Figure 4.1: Scatter plot of XGBoost predictions trained without oversampling fog.

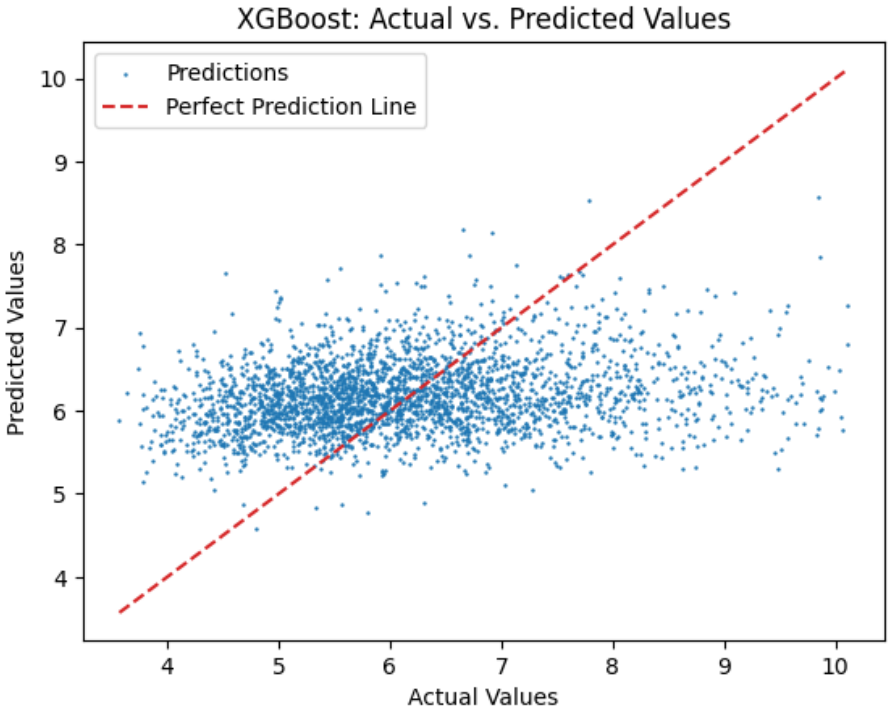

Figure 4.2: XGBoost predictions of low visibility data, below 25,000 meters. The model is trained on data with target values that have been log-transformed, where 25,000 m corresponds to approximately 10.1.

During the exploration phase of this thesis, a Quantile Regression Neural Network (QRNN) was also tested. However, since QRNN performed worse and was less efficient than the presented models, its results were not saved, which accounts for why it is not present in Table 4.1.

An important finding is that, without oversampling, the RMSE values for low visibility conditions are almost twice as high compared to the oversampled data. As these are the points of concern, where fog can be present, this is very problematic.

In summary, while DNN performs slightly better than XGBoost, both models' performances are highly suboptimal. Oversampling fog in the dataset helps improve estimations in low visibility cases, but not overall. A different approach is needed.

4.3 Classification models

Table 4.2 summarizes the performance of some classification algorithms applied to the synthetic dataset. The models evaluated in this section include DNN and XGBoost in different configurations, with the results averaged over five runs for each model. During exploration, several other classification models known to perform with tabular data were tested. These include TabNet, NODE, FTTransformer, and Random forest, as well as anomaly detection models, such as KNN, One class SVM, Isolation forest, and Autoencoder. What was found at this point was that DNN and

XGBoost outperformed all of these alternatives, which is why these are the ones included and further experimented with.

The DNN model, trained with a fog ratio of 0.45, achieved an F1-score of 0.41. It exhibited a high recall of 0.87, but a poor precision of 0.27, suggesting a considerable number of false positives. The accuracy of the DNN model was recorded at 0.78. XGBoost with the standard logistic objective and a fog ratio of 0.45 performed similarly to the DNN with the same F1-score of 0.41. Its accuracy was marginally better at 0.79 as well as the accuracy, scoring 0.28. Contrarily the recall for this configuration was lower than that of the DNN at 0.81. Interchanging the loss function of the DNN model to focal loss, to address class imbalance present in the task, increases F1-score and accuracy at the cost of recall.

With XGBoost configured to use the logitraw objective, logits are not converted into probabilities as they are with the logistic objective, affecting how the data points are classified. Using the same fog ratio of 0.45, this configuration resulted in an increased F1-score of 0.43. However, the recall dropped to 0.75, with precision and accuracy seeing modest increases to 0.30 and 0.82 respectively. The most notable performance was observed with XGBoost configured with logitraw and scaled weights adjusted to account for class imbalance, with a lower fog ratio of 0.25 in the dataset. This setup yielded the highest F1-score of 0.48 and an improved accuracy of 0.88 and precision of 0.41, indicating far fewer false positives. However, this came at the cost of a thoroughly reduced recall of 0.56. The lower recall suggests that the model missed a considerable number of foggy instances, which is an important metric for fog detection.

Despite variations in performance metrics across these configurations, and the fact that DNN and XGBoost were the best-performing models identified, none achieved satisfactory overall performance. Investigation into the reasons behind this suboptimal performance is necessary.

Table 4.2: The performance of different classification algorithms on all clouds. The results of each model are averaged over five runs.

Model	**F1-score**	**Accuracy**	**Recall**	**Precision**
DNN (0.45 fog ratio)	0.41	0.78	0.87	0.27
XGBoost (0.45 fog ratio)	0.41	0.79	0.81	0.28
DNN focal loss (0.30 fog ratio)	0.47	0.86	0.69	0.36
XGBoost logitraw (0.45 fog ratio)	0.43	0.82	0.75	0.30
XGBoost logitraw, scaled weights (0.25 fog ratio)	0.48	0.88	0.56	0.41

4.4 Influence of surface temperature and pressure on classifier performance

In this section, we examine the impact of incorporating surface temperature and pressure into the dataset on the performance of the classifier. These features are included in the training in all other cases, so it is leaving them out that is being tested here. The features' effect on model performance is determined from the changes in the performance metrics.

Table 4.3 demonstrates a minimal, but distinct decrease in performance not using surface temperature and pressure in the dataset. While the difference is small, it is clear to see that surface temperature and pressure aid in fog detection as all performance metrics are higher in this case. Because of this finding, surface temperature and pressure are included features throughout all exploration and presented results, barring the real data in Section 4.9.

Table 4.3: The performance of XGBoost classifier on a dataset with all clouds and a 0.45 proportion of fog, with and without surface temperature and pressure. The results are averaged over five runs.

Dataset includes	F1-score	Accuracy	Recall	Precision
Surface temperature and pressure	0.41	0.79	0.81	0.28
No surface temperature and pressure	0.40	0.78	0.79	0.27

4.5 Impact of fog proportion on classifier performance

This section examines the effects of varying the proportion of fog in the training dataset on the performance of the XGBoost classifier. The aim is to identify trends in the performance metrics related to the composition of the dataset.

The data presented in Table 4.4 illustrates how the classifier's performance metrics are influenced by the frequency at which fog occurs in the training dataset. Results are based on an average over five runs for each composition of fog. These are categorized within the table by the portion of fog used for training, ranging from the full dataset's composition of 9.22% to a balanced out 50%.

Table 4.4: XGBoost classification on all clouds with different portions of fog in data. The values of each model are averaged over five runs.

Proportion of fog in data	F1-score	Accuracy	Recall	Precision
Same as dataset (0.0922)	0.31	0.92	0.20	0.66
1/6	0.45	0.91	0.41	0.49
1/4	0.46	0.88	0.55	0.40
1/3	0.45	0.85	0.69	0.34
2/5	0.43	0.82	0.76	0.30
9/20	0.41	0.79	0.81	0.28
1/2	0.40	0.76	0.85	0.26

The best F1-score was achieved using a dataset with one-fourth foggy data. The results in Table 4.4 show a trend of increasing recall with higher proportions of fog. At the lowest, with a fog rate of 0.0922, the recall is at 0.20. This increases considerably up to 0.85 at the high end with fog in half the dataset. This suggests that the model improves its ability to identify fog when it's exposed to a higher frequency of these cases during training.

In contrast to recall, a downward trend as fog proportions increase can be noted for both precision and accuracy. Precision drops from 0.66 to 0.26, and accuracy from 0.92 to 0.76 as the dataset's composition transitions from the lowest to the highest split of fog. This pattern indicates that the model over-predicts fog when this becomes more frequent in the training, leading to more false positives.

4.6 Performance across different cloud types

Investigations were made into the performance of the XGBoost classifier on different cloud types in an attempt to identify the cause of misclassifications. In doing so the prevalence of different cloud types in the dataset was also uncovered, leading to further experiments with balancing this composition in the dataset.

Table 4.5 displays the classifier's performance for each cloud type, detailing the rates of false positives and false negatives when trained using the distribution of cloud types in the full dataset. The table also shows the prevalence of each type within the set of data. What is found is that clear conditions are present in about a third of the data points and that the model does very well at classifying these occurrences with only a 0.16% false positive rate. Another thing that is apparent in Table 4.5 is that most cloud types occur in the dataset at an approximately equal rate with the exception of cirrus clouds. Cirrus clouds, similarly to clear skies, are present in about a third of the data in this set. Furthermore, while the model's performances on the other cloud types are mostly similar, the performance on cirrus clouds is considerably better, especially regarding false positives.

Table 4.5: The performance of XGBoost classification, trained using a dataset with the full dataset's distribution of cloud types and a fog ratio of 0.45. Results are split to show the model's performance on each cloud type and the rate at which these occur.

Cloud type	**Prevalence in dataset** [%]	**False positives** [%]	**False negatives** [%]
None (Clear)	33.16	0.16	-
Cirrus	33.39	20.33	10.57
Stratus Continental	6.87	44.88	12.18
Stratus Maritime	6.64	46.33	19.63
Cumulus Continental Clean	6.30	43.26	11.30
Cumulus Continental Polluted	6.86	43.25	10.55
Cumulus Maritime	6.77	47.55	12.96

Subsequent analysis, as shown in Table 4.6, confirms these observations under conditions of evenly distributed cloud types within the dataset. Note that even though the training data has been altered, the models are tested on a test set with proportions representative of the original data shown in Table 2.2. It is found that the model continues to show strong performance on clear skies, despite the balanced dataset, and still excels on cirrus clouds compared to other types. This consistency suggests that the classifier's superior results for cirrus clouds are not only due to their higher prevalence in the initial dataset. Instead, it may indicate that the dataset contains features particularly effective for identifying cirrus clouds.

Table 4.6: The performance of XGBoost classification, trained using a dataset with an equal distribution of cloud types and a fog ratio of 0.45. Results split to show the model's performance on each cloud type when these all occur with equal frequency during training.

Cloud type	**Prevalence in dataset** [%]	**False positives** [%]	**False negatives** [%]
None (Clear)	14.29	0.11	-
Cirrus	14.29	19.18	17.74
Stratus Continental	14.29	29.16	23.94
Stratus Maritime	14.29	31.96	26.13
Cumulus Continental Clean	14.29	29.75	24.74
Cumulus Continental Polluted	14.29	26.65	30.46
Cumulus Maritime	14.29	33.26	23.88

Comparing tables 4.5 and 4.6 we see that equalizing the proportions of cloud types results in an increase in false negatives across all cloud types and a decrease in false positives. This shift highlights the trade-off between sensitivity and precision once again, showing that a more balanced dataset reduces the risk of falsely predicting fog, but increases the likelihood of missing this event.

4.7 Isolating cloud types

We explored the idea of training models on a specific cloud type to see if narrowing down the differences in the training would help with performance. This approach involved first having a classifier capable of identifying different cloud types and then several models trained on isolated cloud types. Table 4.7 shows the performance of XGBoost classifying cloud types. Most notably the model performs almost perfectly in identifying clear conditions and cirrus clouds. Identification of the rest of the cloud types show much worse results with the best being cumulus continental polluted with an f1-score of only 0.52. Because of the good performance in identifying cirrus clouds, the possibility of a method for detecting cirrus fog was tested. However, training and testing a classification model on only cirrus clouds and fog resulted in negligible improvement compared to what is shown in Section 4.6.

4.8 Other explored areas

The synthetic dataset contained several features that were not used for training models. This is because they are not directly recorded by MODIS, nor are they

Table 4.7: The performance of XGBoost classifying cloud types in the synthetic dataset. Overall accuracy was 0.78.

Cloud type	F1-score	Recall	Precision
None (Clear)	1.00	1.00	1.00
Cirrus	0.99	0.99	0.99
Stratus Continental	0.29	0.32	0.27
Stratus Maritime	0.29	0.27	0.30
Cumulus Continental Clean	0.27	0.24	0.31
Cumulus Continental Polluted	0.52	0.51	0.53
Cumulus Maritime	0.39	0.43	0.36

Table 4.8: The performance of XGBoost classifier on a dataset with all clouds and a 0.45 proportion of fog, with and without cloud optical thickness, gas optical thickness, vertically integrated water vapor and cloud type.

Dataset	F1-score	Accuracy	Recall	Precision
Without extra features	0.41	0.79	0.81	0.28
Including extra features	0.48	0.82	0.92	0.32

as simple to acquire for real data as surface temperature and surface pressure. To figure out if this additional information would have an impact on performance, a test was conducted including the extra features when training the XGBoost classifier. The extra features that were included were cloud optical thickness, gas optical thickness, vertically integrated water vapor and cloud type. In Table 4.8 it is shown the additional features do effectively help improve the models, with the F1-score increasing to 0.48 as accuracy, recall and precision are all elevated.

It was suspected that it was easiest to detect fog when the sun was high in the sky during the day. To test if classification during this part of daytime would perform better, the XGBoost model was trained and tested on a dataset containing only data points with a sun zenith angle between 0 and 15 degrees, and a fog portion of 0.45. However, this test showed that the performance in these conditions had a negligible difference compared to the full dataset with sun zenith angles ranging up to 90 degrees. As such, limiting the algorithm to specific times of day was ruled out and solutions for an all day model was the continued focus.

4.9 Classifying pixels in real MODIS images

Using XGBoost, trained with a fog ratio of 0.45, to classify pixels in real MODIS images reveals that the machine learning model is identifying certain areas as fog, as shown in Figure 4.3. These predictions are binary, with values corresponding to either 0 or 1, with 1 corresponding to fog. Regions with intermediate colors on the color bar indicate areas with varying prediction densities. Figure 4.4 provides a zoomed-in view of the predictions over Denmark and southern Sweden, centered around Copenhagen. In this area, where fog is anticipated during 2021-12-18, the model predicts minimal fog presence. Figure 4.5 displays the model's initial predictions before they are classified into binary values, where initial predictions exceeding 0.5 are classified as fog.

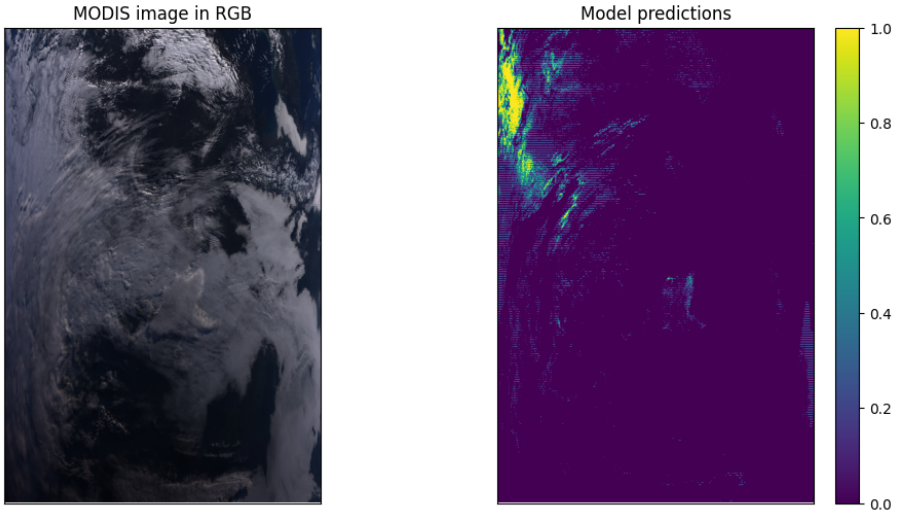

Figure 4.3: XGBoost predictions of MODIS image from 2021-12-18 13:20 (UTC). MODIS image shown in RGB color spectrum (left) and models predictions (right)

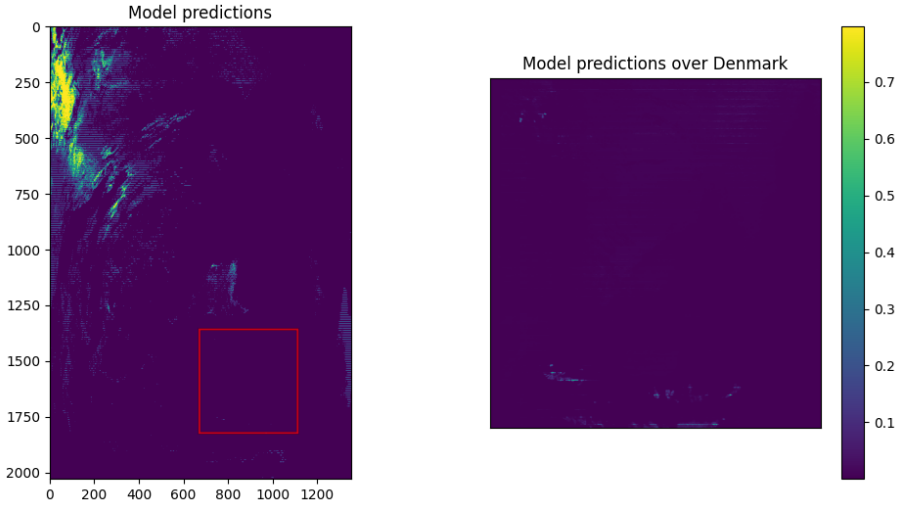

Figure 4.4: XGBoost predictions of MODIS image from 2021-12-18 13:20 (UTC) zoomed in over Denmark and south of Sweden, centered around Copenhagen.

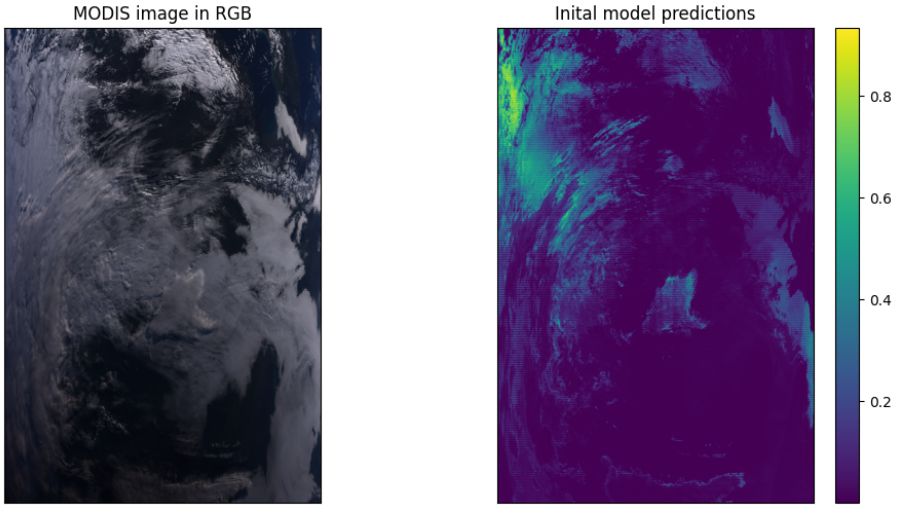

Figure 4.5: Initial XGBoost predictions of MODIS image from 2021-12-18 13:20 (UTC). MODIS image shown in RGB color spectrum (left) and initial models predictions (right). Initial predictions exceeding 0.5 are classified as fog.

5
Discussion

In this chapter, the results and the decisions made throughout the project are discussed. An analysis is also made of the underlying causes of the results. Finally, the synthetic dataset is discussed, as well as the limited real MODIS data that was acquired and used.

5.1 Comparing regression and classification models

During the initial phase of the study, we decided to explore various alternative machine learning models. Our primary objective was to train these models not simply to classify whether a pixel was foggy or not but also to estimate the ground-level visibility. This decision was motivated by the fact that the data used in our study was simulated, thus providing precise measurements of ground-level visibility. By incorporating this variable, we aimed to determine if the spectral information captured by MODIS, in conjunction with additional features such as surface pressure and temperature from the simulated data, could be used to accurately estimate ground-level visibility. We believed that estimating visibility would offer more valuable information compared to a binary classification of foggy or not foggy. This would have the benefits of enabling different levels of warnings could be used depending on fog density, and distinguishing between hazy and clear conditions.

Given that the regression models primarily fitted the data points with high visibility, which isn't particularly informative for fog detection purposes, the decision was made to employ a classification model first. This model would classify each data point as either low visibility or high visibility. Subsequently, a regression model trained specifically on low visibility data points could be utilized to offer more accurate predictions of ground-level visibility. By linking together these models as a chain, it could potentially enhance predictions compared to a standalone regression model.

As depicted in Figure 4.2, the regression model trained exclusively on low visibility data fails to accurately predict ground-level visibility, even after log-scaling the target values. Nevertheless, the initial classification between high and low visibility data points demonstrated some promise in differentiating between the two. Seeing these results it was decided to further investigate classification models for fog detection.

5.2 Trade-offs between recall, precision, and accuracy in fog detection

An inverse relationship is observed throughout the tables in the Results chapter, where an increase in recall comes at the cost of reduced precision and accuracy. This is a typical trade-off that is common in classification tasks, especially those with uneven datasets. Recall is a measure of the classifier's ability to identify positive instances within the dataset. For fog detection, a high recall means that most of the foggy cases are classified as foggy, true positives. This is of course a crucial aspect of a fog detection algorithm as its purpose is to provide warnings for safety. Additionally all unnoticed instances of fog, false negatives, can mean danger for traffic, clearly recall is very important. However, this is where the trade-off comes in. As the recall increases, precision reduces, which in practice means that the number of falsely detected instances of fog increases. While this might not appear as immediately dangerous, there are several possible effects to consider. With lower precision, warnings of fog might not be taken as seriously. Consider a case where fog warnings are more commonly incorrect than correct, people would likely ignore them, which would make all warnings insignificant. These different aspects make the trade-off a very central question and what to prioritize is far from simple.

Several factors were identified as contributing to the trade-off and these can be altered to tune the model towards the prioritized sweet spot in the trade-off. One of these factors is the composition of the dataset. A larger proportion of fog in the training data makes the model more attuned to recognize fog which results in higher recall, but lower precision. Another factor is the model sensitivity. By adjusting the requirements for what is predicted as fog conditions for the trade-off are changed. Lowering the threshold for classifying fog means more fog predictions, higher recall, and lower precision.

5.3 Performance on different cloud types

When classifying a test set representative of the original data, with a minority of fog, models trained on more equal class distributions yielded a considerable number of false positives. In response, we explored various strategies to lower the amount of false positives. These strategies include altering the classification threshold, adjusting class weights to account for uneven distributions of the classes, and employing different loss functions. However, each attempt to address this issue led to an elevated number of false negatives.

In order to gain deeper insights into the root causes of the high false positive rate observed in the classification models, we extracted the original data points from the test dataset post-classification. This enabled us to investigate the properties of the misclassified points that were not present in the training data. Through visualization of specific features in the misclassified data, our objective was to discern patterns that would explain why certain points were misclassified. Although this

analysis did not provide any clear indications of patterns related to features such as pressure, temperature, or vertically integrated water vapor, it revealed that the false positives were predominantly attributed to clouds, particularly water clouds.

In Section 4.6 we showed that the classifier performs differently across the various cloud types. Most notably the model performed substantially better at detecting fog in cirrus cloud conditions compared to the other cloud types. This cloud type also occurs much more frequently in the dataset than the rest do. To examine if the poor performance across the other cloud types was due to their low prevalence, a new test was run with a dataset split equally across all cloud types. The results, seen in Table 4.6, showed that this was not the case, once again displaying a superior performance in cirrus conditions compared to the other cloud types. While distribution could be ruled out, figuring out the reason for the uneven performance may help improving the model overall.

Cirrus clouds, often called ice clouds, are composed primarily of ice crystals. In contrast, stratus and cumulus clouds, also known as water clouds, are mainly composed of water droplets. A key difference between cirrus clouds and water clouds is that they have different spectral properties because ice crystals scatter and absorb light differently compared to water droplets. This difference is particularly prominent in the infrared spectrum. The unique spectral signature of ice clouds provides a clearer differentiation that models utilizing spectral data can exploit. Cirrus clouds form in very cold conditions most often at high-altitudes above 6000 meters [33]. We believe that the difference in performance can be attributed to the larger difference in altitude between cirrus clouds and fog, compared to water clouds and fog. Cirrus fog only forms when the air near the ground is very cold so surface temperature may be more informative for ice fog detection. These differences likely also account for the much more accurate identification of cirrus clouds compared to water clouds, shown in Table 4.7.

Another interesting result presented in Section 4.6 is the high accuracy on clear conditions demonstrated by the model. This is to be expected given the distinct contrast between clear skies and foggy conditions. The absence of clouds to emulate fog by scattering and absorbing radiation leads to distinctive spectral signatures that are easily identifiable by the model.

One approach aimed at enhancing model performance involved training a classification model to distinguish between different cloud types. Following this classification, the data points could then be fed into another model trained specifically for that cloud type to determine whether fog was present or not. Our findings revealed that the first model could effectively classify clear and cirrus clouds with high accuracy. However, misclassifications were common among the water clouds. Additionally, training a model to classify fog in only one particular cloud type yielded negligible improvement compared to models trained across multiple cloud types. The same trade-off between precision and recall was the only thing seen here too. Given the limited potential observed in combining these models, further exploration into this

approach was discontinued.

5.4 Influence of surface temperature and surface pressure on fog detection

In Section 4.4 the effects of including surface temperature and surface pressure in the dataset was presented. Incorporating this information into the synthetic dataset for fog detection was anticipated to substantially enhance the model's performance. These expectations were due to the features' influence on atmospheric conditions that affect water vapor and gasses at ground-level. These factors, in turn influence the extinction coefficient, which is central to Koschmieder's law for calculating visibility. The results, however, revealed only a modest improvement in the performance metrics, warranting a deeper discussion.

The inclusion of surface temperature and pressure in the dataset led to a small but noticeable improvement in the F1-score for fog detection. This upgrade was achieved without any trade-off between precision and recall, rather, both metrics saw slight increases. This means that the additional features helped the model better capture actual cases of fog, while also reducing false fog predictions.

While the effects were exclusively positive, the small magnitude of them, F1-score increased by only 0.01, was surprising. As stated, the surface temperature and pressure indirectly affect the extinction coefficient. Because of this, the expectation was for these features to have a strong correlation with ground-level visibility, thereby substantially improving fog detection. Why this was not the case may be related to the synthetic dataset. Atmospheric conditions are dynamic and complex, with a lot of variability and interactions. It is possible that the intricacies involved with the influence of surface temperature and pressure might not be fully captured when simulating the synthetic dataset. Another possible explanation is that other features, such as the spectral bands, are more dominant factors. These may already contain the majority of the information that surface temperature and pressure provide for fog detection. This would explain the marginal gain as most of what these features can contribute to the task is already present in the rest of the data. Notable about this explanation is that it implies the dataset does not contain sufficient information for a high-performing, global fog detection algorithm. Consequently, additional information would be required to reach satisfactory performance if this is the case.

5.5 Synthetic dataset

The use of synthetic data in the development and evaluation of fog detection algorithms presents both opportunities and challenges. While the results do not demonstrate outstanding performance in fog detection, several observations and considerations highlight the potential and limitations of this synthetic dataset.

Although the performance of the fog detection models on the synthetic dataset was not fully satisfactory, it is evident that the classifier predictions are not random. The models have shown the ability to learn some things and to achieve high recall, meaning they can identify a substantial proportion of actual fog. In Table 4.8 the classifier results when including the additional features of cloud optical thickness, gas optical thickness, cloud type and vertically integrated water vapor are displayed. While the classifier still did not perform exceptionally well despite being trained with the additional features, it did show a notable improvement. This signals that there may be potential for a satisfactory fog detection model by including additional information from other sources, more on this in Section 7.2.

One notable concern with using only synthetic data is the transfer of biases, that are inherent in the data, to the models. The composition of the synthetic dataset, including the distribution of cloud types, and the frequency of clear and foggy conditions, may not accurately reflect real-world conditions. These discrepancies can vary by location, season, and time of day, potentially leading to biased model performance when applied to real-world scenarios. It is important to consider these potential biases, especially when transferring a model trained on synthetic data to be used with real data.

Another concern is that the synthetic dataset used in this study does not perfectly simulate real-world conditions. Most notably, it does not account for the effects of aerosols, which can impact visibility and atmospheric properties. As a result, reflectivity bands 8, 9, and 10, which cover parts of the solar range and are heavily influenced by aerosols in real data, were not used in this thesis. While it is only these three bands that are disregarded, aerosols can also have some impact on other features in real data. Not replicating this in the synthetic data could potentially affect the accuracy and robustness of the fog detection models.

As stated in Section 1.2, one of the biggest challenges in fog detection is the spectral similarity between low altitude clouds and fog. These similarities likely make it difficult for the models to distinguish between the two. Unfortunately, the synthetic dataset does not include information on cloud height, preventing a direct test of this hypothesis. Including cloud height data in future datasets would enable a more thorough investigation and potentially confirm whether low altitude clouds are the primary source of detection difficulties. Understanding this aspect better could be highly beneficial for improving model performance.

5.6 Classifying pixels in real MODIS images

Evaluating the models trained on synthetic data using actual MODIS images proved to be a significant challenge. As illustrated in Figure 4.3, the models do not produce random predictions. Instead, they identify some of the cloud formations visible in the RGB image as containing fog. It is especially evident from Figure 4.5 that the predictions correspond to many of the observed cloud structures. However, the results outside of the marked region in Figure 4.4 do not provide conclusive infor-

mation, as we cannot determine if fog is expected in those areas. Additionally, due to the high false positive rates for several cloud types when testing the model on synthetic data, there is a possibility that the fog predictions are misclassifications of clouds.

Inside the marked area centered around Copenhagen, some fog is indeed detected. However, without more specific geographical information regarding the actual fog, we cannot confirm if the detected fog corresponds to real fog. Additionally, we expected a higher density of fog classifications within the marked area. The results suggest that the models trained solely on synthetic data are not very effective when directly applied to real images. However, further testing over other regions where fog has been observed is needed to obtain a definitive answer.

During the processing of the real data, we noticed a significant difference between the synthetic and real data points. In the synthetic data, the satellite zenith angles for the data points ranged between 0 and 15 degrees. However, the real data contained satellite zenith angles ranging up to 65 degrees. This discrepancy means that during the classification of the real images, the model trained on synthetic data is exposed to feature values not present during training, potentially affecting the model's predictions. Generating and using a new synthetic dataset that includes larger satellite zenith angles would be a necessary step to produce more robust predictions in real satellite images.

6
Conclusion

In this study, we developed and evaluated machine learning models to detect fog using independent pixel approximation on synthetic data. Our findings indicate that regression models trained to estimate ground-level visibility are ineffective for foggy data points. Conversely, classification models demonstrated the ability to discern fog from fog-free data points, achieving high recall scores. However, due to the heavy class imbalance, these models produced many false positives, often mistaking clouds for fog. This suggests that using only spectral information, along with surface pressure and temperature, it is challenging to distinguish fog from certain clouds with the methods employed in this project. We suspect that the information contained within our synthetic dataset and by extension what is captured by MODIS, is not sufficient for distinguishing between fog and low altitude clouds. Applying a classification model to real satellite images showed that the predictions align with cloud structures visible in the RGB color spectrum, but detected minimal fog where we expected it. Given the limitations and the broad scope of exploration within a limited time frame, these results do not conclusively negate the potential of utilizing synthetic data for fog detection. There are still many avenues for improvement, as discussed in Chapter 7.

7
Future work

This thesis has extensively explored the efficacy of a synthetic dataset in training algorithms for fog detection. Despite the comprehensive range of experiments conducted, some areas remain unexplored. As highlighted in the conclusion chapter, it is suspected that the MODIS instrument data lacks sufficient detail to effectively differentiate between fog and certain types of clouds, especially those at low altitudes. Nonetheless, there are several untested strategies that could potentially address these unresolved challenges. The following sections propose strategies to enhance future research in this area.

7.1 Real data

One limitation of utilizing the RTTOV-generated synthetic dataset is its lack of spatial information. Each data point corresponds to an individual pixel in a satellite image, without any contextual data from adjacent pixels. This isolation poses a challenge for developing algorithms that could benefit from spatial analysis, where the estimate for one pixel could be influenced by its surrounding pixels. To advance this aspect of the research, future work could focus on developing a post-processing algorithm that incorporates spatial information. Implementing such an algorithm requires access to datasets where spatial relationships between pixels are preserved. Real-world satellite imagery, where pixels are manually labeled using ground truth data from ground stations, could provide the necessary context. We believe that this is the most important area for the future trajectory of this research

7.2 Advanced pre-processing

There is a lot that can be done with satellite data in order to pre-process it. This can include combining it in different equations, adding supplementary data from other sources, and cutting out data that proves irrelevant. Throughout the project a lot of basic pre-processing was experimented with, however, the more advanced methods were left outside of the scope due to time and resource constraints. This makes for a promising direction for future research.

A key strategy for future advancements could involve taking advantage of other instruments by combining MODIS data with relevant and trustworthy data predicted by NWC-SAF [34]. Data such as cloud top temperature, pressure and height, cloud

type, and cloud masks could all be retrieved from other nowcasting. We hypothesize that this data is highly relevant for distinguishing fog and could substantially enhance model performances. Additionally, information about the surface type might be helpful when combined with the spectral data.

Implementing this advanced pre-processing would probably involve developing helper algorithms that can efficiently integrate diverse data types. This approach would likely require collaborative efforts with satellite data providers and meteorological agencies to access real-time data while ensuring that the fog detection model is reliable.

7.3 Ensemble learning

An, in theory, quite simple avenue for future exploration would be to try utilizing ensemble learning approaches. This could improve prediction robustness by aggregating the strengths of multiple learning algorithms, thereby enhancing overall performance [35] [36]. One possibility would be to create an ensemble of models on either side of the precision/recall trade-off. However, while there is a chance that this could improve the performance, it would also be more computationally expensive and subsequently a less resource efficient alternative.

7.4 Residual learning

Residual learning could be a way of achieving deeper neural network architectures without overfitting. With residual learning, layers are grouped into blocks and it works by forwarding the input of each block to the end of the block and including the input as part of the output. In this way the vanishing gradient problem is avoided, meaning that much deeper models can work without overfitting. This could potentially lead to the discovery of patterns within the data that the models tested in this thesis were not able to identify.

7.5 QRNN

In this thesis, further exploration into QRNN was identified as a promising avenue for future research. One of the standout features of QRNN is its ability to provide not only predictions but also associated confidence intervals for these predictions. This dual capability can be particularly valuable in cases where a distinct subset of the data is challenging for the model to hit the target of. This is what we believe low altitude clouds are for the fog detection dataset.

The confidence intervals generated by QRNN could serve as a critical diagnostic tool. The model might indicate low confidence in predictions where the model struggles to differentiate between fog and clouds, which are often problematic areas in current models. Conversely, when QRNN shows high confidence, it suggests that

the visibility predictions are likely to be accurate. This mechanism could enable meteorologists to treat high-confidence predictions as reliable for nowcasting, while predictions marked by low confidence could be categorized as ‘unclear’. This ability to flag ’unclear’ predictions could make the model useful to nowcasting even with a less than desirable estimation accuracy.

Bibliography

[1] Y. Wang, Z. Qiu, D. Zhao, *et al.*, "Automatic detection of daytime sea fog based on supervised classification techniques for fy-3d satellite", *Remote Sensing*, vol. 15, no. 9, 2023, ISSN: 2072-4292. [Online]. Available: https://www.mdpi.com/2072-4292/15/9/2283.

[2] "Detection of dawn sea fog/low stratus using geostationary satellite imagery", *Remote Sensing of Environment*, vol. 294, p. 113 622, 2023, ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2023.113622. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425723001736.

[3] H. Yan, S. Su, M. Wu, *et al.*, "Seamae: Masked pre-training with meteorological satellite imagery for sea fog detection", *Remote Sensing*, vol. 15, no. 16, 2023, ISSN: 2072-4292. [Online]. Available: https://www.mdpi.com/2072-4292/15/16/4102.

[4] T. Li, W. Jin, R. Fu, and C. He, "Daytime sea fog monitoring using multimodal self-supervised learning with band attention mechanism", *Neural Computing and Applications*, vol. 34, no. 23, pp. 21 205–21 222, Dec. 2022, ISSN: 1433-3058. DOI: 10.1007/s00521-022-07602-w. [Online]. Available: https://doi.org/10.1007/s00521-022-07602-w.

[5] M. Schulz, B. Thies, S.-C. Chang, and J. Bendix, "Detection of ground fog in mountainous areas from modis day-time data using a statistical approach", *Atmospheric Measurement Techniques Discussions*, vol. 8, pp. 12 155–12 201, Nov. 2015. DOI: 10.5194/amtd-8-12155-2015.

[6] H. Andersen and J. Cermak, "First fully diurnal fog and low cloud satellite detection reveals life cycle in the namib", *Atmospheric Measurement Techniques*, vol. 11, no. 10, pp. 5461–5470, 2018. DOI: 10.5194/amt-11-5461-2018. [Online]. Available: https://amt.copernicus.org/articles/11/5461/2018/.

[7] K. C. Cavanaugh, T. Bell, M. Costa, *et al.*, "A review of the opportunities and challenges for using remote sensing for management of surface-canopy forming kelps", *Frontiers in Marine Science*, vol. 8, p. 753 531, 2021.

[8] A. Pirinen, N. Abid, N. A. Paszkowsky, *et al.*, "Creating and leveraging a synthetic dataset of cloud optical thickness measures for cloud detection in msi", *Remote Sensing*, vol. 16, no. 4, p. 694, 2024.

[9] R. Saunders, J. Hocking, E. Turner, *et al.*, "An update on the rttov fast radiative transfer model (currently at version 12)", *Geoscientific Model Development*, vol. 11, no. 7, pp. 2717–2737, 2018.

[10] M. Domnich, I. Sünter, H. Trofimov, *et al.*, "Kappamask: Ai-based cloudmask processor for sentinel-2", *Remote Sensing*, vol. 13, no. 20, p. 4100, 2021.

[11] *Modis*, https://aqua.nasa.gov/modis, [Accessed: 2024-02-15].

[12] Z. Lee and S. Shang, "Visibility: How applicable is the century-old koschmieder model?", *Journal of the Atmospheric Sciences*, vol. 73, no. 11, pp. 4573–4581, 2016. DOI: 10.1175/JAS-D-16-0102.1. [Online]. Available: https://journals.ametsoc.org/view/journals/atsc/73/11/jas-d-16-0102.1.xml.

[13] S. Zhang and L. Yi, "A comprehensive dynamic threshold algorithm for daytime sea fog retrieval over the chinese adjacent seas", *Pure and Applied Geophysics*, vol. 170, pp. 1931–1944, 2013.

[14] NASA, *MODIS: Moderate Resolution Imaging Spectroradiometer*, https://modis.gsfc.nasa.gov/, [Accessed: 2024-03-19], 2021.

[15] *Terra: The eos flagship*, ://terra.nasa.gov/, [Accessed: 2024-03-19].

[16] C. O. Justice, E. Vermote, J. R. Townshend, *et al.*, "The moderate resolution imaging spectroradiometer (modis): Land remote sensing for global change research", *IEEE transactions on geoscience and remote sensing*, vol. 36, no. 4, pp. 1228–1249, 1998.

[17] N. E. Science. "Earth's top-of-atmosphere radiation budget". (2024), [Online]. Available: https://ceres.larc.nasa.gov/documents/science_team/publications/Loeb_et_al_ScienceDirect_2016.pdf.

[18] K. Trenberth, J. Fasullo, and J. Kiehl, *Radiation budget diagram of earth's atmosphere*, https://scied.ucar.edu/image/radiation-budget-diagram-earth-atmosphere, Accessed: May 7, 2024, 2011.

[19] K. Lakra and K. Avishek, "A review on factors influencing fog formation, classification, forecasting, detection and impacts", *Rendiconti Lincei. Scienze Fisiche e Naturali*, vol. 33, no. 2, pp. 319–353, 2022.

[20] G. W. Petty, *A first course in atmospheric radiation*. Sundog Publishing LLC, 2023, ch. 7.

[21] World Meteorological Organization, *Aviation | hazards | low visibility and low cloud*. [Online]. Available: https://community.wmo.int/en/activity-areas/aviation/hazards/visibility.

[22] J. Delua. "Supervised vs. unsupervised learning: What's the difference?", IBM. (Mar. 2021), [Online]. Available: https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning.

[23] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012, ch. 16.

[24] V. Lendave. "Gini impurity vs information gain vs chi-square - methods for decision tree split". Accessed: 2024-05-22. (2021), [Online]. Available: https://analyticsindiamag.com/gini-impurity-vs-information-gain-vs-chi-square-methods-for-decision-tree-split.

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

[26] Y. Zhang, X. Beudaert, J. Argandoña, S. Ratchev, and J. Munoa, "A cpps based on gbdt for predicting failure events in milling", *The International Journal of Advanced Manufacturing Technology*, vol. 111, pp. 341–357, 2020.

[27] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016, ch. 6.

[29] A. J. Cannon, "Quantile regression neural networks: Implementation in r and application to precipitation downscaling", *Computers & geosciences*, vol. 37, no. 9, pp. 1277–1284, 2011.

[30] Q. Xu, S. Liu, C. Jiang, and X. Zhuo, "Qrnn-midas: A novel quantile regression neural network for mixed sampling frequency data", *Neurocomputing*, vol. 457, pp. 84–105, 2021.

[31] N. Håkansson, "Nwcsaf cloud products - low clouds and fog", Presented on April 27, 2023, 2023.

[32] R. B. Smith, *Computing the planck function*, `https://yceo.yale.edu/sites/default/files/files/ComputingThePlanckFunction.pdf`, Revised September 21, 2005, Sep. 2005.

[33] National Weather Service, *Cloud classification*, `https://www.weather.gov/lmk/cloud_classification`, Accessed: 2024-05-20, 2024.

[34] NWC SAF, *EUMETSAT NWC SAF - Support to Nowcasting and Very Short Range Forecasting*, `https://www.nwcsaf.org/`.

[35] C. Makhijani, "Advanced ensemble learning techniques", *Towards Data Science*, 2020. [Online]. Available: `https://towardsdatascience.com/advanced-ensemble-learning-techniques-bf755e38cbfb`.

[36] A. Bonnet, "What is ensemble learning?", *Encord Blog*, 2023. [Online]. Available: `https://encord.com/blog/what-is-ensemble-learning/`.

DEPARTMENT OF SPACE, EARTH AND ENVIRONMENT
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY