CLARITY THROUGH A CLOUDED MIND

Developing machine learning algorithms to forecast visibility categories at Amsterdam Airport Schiphol

By Vera Buis

Master Thesis

Meteorology & Air Quality

Wageningen University & Research

In collaboration with

LVNL / Air Traffic Control the Netherlands

Supervisors

Dr. ir. Gert-Jan Steeneveld (WUR)

Ferdinand Dijkstra (LVNL)

Abstract

Accurate weather forecasts are crucial for minimizing delays at airports. Specifically low-visibility conditions can greatly impair airport operations, reducing capacities up to 67%. However, current Numerical Weather Models often struggle to accurately forecast low-visibility conditions. In this study, we develop and evaluate two types of classifying machine learning algorithms for forecasting visibility conditions. We introduced an independent, deterministic Random Forest Classifier for classifying pre-defined classes. The model performed well at short timescales, but its performance quickly declined over the forecasting horizon, particularly for intermediate classes. Additionally, we developed a probabilistic Temporal Fusion Transformer, which we fitted with a custom Focal Loss function for the first time in visibility forecasting. We demonstrate the success of this approach in forecasting visibility, specifically for low horizontal visibility. The results highlight the potential of using Temporal Fusion Transformers for short-scale, categorical visibility forecasts.

Table of Contents

List of Abbreviations

Table 1: List of abbreviations used throughout this thesis.

Abbreviation	Description
AI	Artificial Intelligence
ATC	Air Traffic Control
AWS	Automated Weather Sensor
BS	Brier Score
BSS	Brier Skill Score
BZO	Beperkt Zicht Omstandigheden
CLB	Cloud Base Height (ceiling)
CSI	Critical Success Index
FAR	False Alarm Ratio
ILS	Instrument Landing System
KNMI	Koninklijk Nederlands Meteorologisch Instituut (Royal Netherlands Meteorological Institute)
LVNL	Luchtverkeersleiding Nederland (Air Traffic Control the Netherlands)
LVP	Low Visibility Procedures
ML	Machine Learning
NWM	Numerical Weather Model
NWP	Numerical Weather Prediction
POD	Probability of Detection
PWS	Present Weather Sensor
RF	Random Forest
RFC	Random Forest Classifier
RVR	Runway Visual Range
TFT	Temporal Fusion Transformer

1 Introduction

The weather substantially impacts daily life: people check the weather to plan what they will wear, when to be inside or outside, or when it is safe to travel. Extreme weather events can greatly influence safety, especially in transportation sectors such as road (Goodwin & Pisano, 2003; Westcott, 2007) and air traffic (De Villiers & Van Heerden, 2007; Keith & Leyton, 2007; Stolaki et al., 2009). An example of an extreme weather event is low visibility. Low visibility prevents people from noticing other vehicles or objects, often resulting in accidents in road, marine and air traffic (Wu et al., 2018; Bergot & Koracin, 2021). In aviation, extra measures are taken to ensure safety during low visibility conditions. These measures decrease an airport's capacity, often leading to delays (Pejovic et al., 2009). Delays result in significant economic consequences as well as unnecessary CO_2 emissions.

In 2019, Amsterdam Airport Schiphol, the world's number 11 largest airport, experienced one million minutes of delay. Each minute results in a cost of about €100 (LVNL, 2023) for either the airport or airlines. Many of these delays were caused by fog or heavy rain, which both greatly reduce visibility (Mullers, 2023). To reduce these delays, Air Traffic Control at Amsterdam Airport Schiphol, LVNL, developed a Decision Support Tool (DST) that combines meteorological forecasts with flight plans to forecast the airport's capacity every 10 minutes for the coming four hours. LVNL divides low-visibility conditions into 6 categories, each with a maximum airport capacity. Therefore, LVNL requires accurate forecasting of these visibility categories to predict the airport's capacity. Additionally, LVNL bases the airport's capacity on visibility conditions at two key locations, requiring a separate visibility forecast for each. Currently, LVNL uses a visibility forecast based on a Numerical Weather Model, or NWM.

Problem statement

The main factor that determines visibility is fog, but NWM often perform poorly in forecasting fog (Bergot et al., 2007; van der Velde et al., 2010; Román-Cascón et al., 2012; Zhou et al., 2012; Steeneveld et al., 2015; Herman & Schumacher, 2016; Izett et al., 2018; Martinet et al., 2020). This is partly caused by the fact that fog depends on many different variables and can develop through various processes. Moreover, fog often occurs on a small spatial and temporal scale (Tapiador et al., 2019). NWM often produce forecasts over a gridded domain with a spatial scale of several kilometers and temporal scales of hours, making it very difficult for NWM to accurately predict the location, time of onset and duration of fog with high detail (Steeneveld et al., 2015). This level of detail is essential to airport operations, as local or short-lived fog can still cause major disruptions and delays (Huang & Chen, 2016).

As a result of the reasons mentioned, this research focuses on the following problem statement:

> *Current visibility forecasts produced by NWM are of insufficient quality for aviation, impeding timely prevention of delays, leading to unnecessary costs and emissions*

Scientific context

Recent research has identified machine learning as a possible solution to this problem. Studies have shown that machine learning (ML) algorithms show promising results in forecasting visibility (Bartoková et al., 2015; Colabone et al., 2015; Guijo-Rubio et al., 2018; Miao et al., 2020; Bartok et al., 2022; Castillo-Botón et al., 2022; Salcedo-Sanz et al., 2022). The major difference between ML algorithms and NWM is

that these models are trained to recognize historical patterns in atmospheric conditions rather than rely on physics and parameterizations. An ML algorithm could recognize the conditions related to fog formation without having to understand the physics of fog formation. Furthermore, NWM often produce forecasts on temporal and spatial scales too large to capture short-lived or local fog. Since ML algorithms rely on historical patterns, they are capable of producing forecasts at the same scale at which observations were recorded. ML algorithms thus exclude the need for high mathematical complexity, making them quicker, easier and cheaper to operate than NWM (Schultz et al., 2021) while still producing small-scale forecasts.

Boneh et al. (2015), Durán-Rosal et al. (2018) and Miao et al. (2012) all successfully implemented ML methods on fog forecasts or nowcasts at different airports, the first and latter even being operational now. In these studies, ML algorithms were applied as a nowcasting algorithm or post-processing technique for NWM predictions. In the latter example, ML algorithms are used to improve and fine-tune NWM output.

Knowledge gap

This post-processing technique still utilizes complex and computationally expensive NWM, producing forecasts at the same spatial and temporal scale as NWM. This approach does not maximize the potential benefits of using ML: forecasting visibility on smaller spatial and temporal scales, without increasing costs and computational power.

Fewer studies were conducted on machine learning algorithms that independently forecasted visibility conditions (Fabbian et al., 2007; Cornejo-Bueno et al., 2017; Castillo-Botón et al., 2022). Castillo-Botón et al. (2022) also focused on the classification of visibility. The categories used in this study were based on the dataset's statistics and not pre-defined like the categories used by LVNL. To the author's knowledge, there is no research on independent ML algorithms that forecast visibility conditions into pre-defined categories. Additionally, no research has been done on ML algorithms that forecast visibility conditions at multiple locations. Finally, the research mentioned also focused on relatively long timescales, from 1 up to 48 hours. There is very limited research on predictions on very short lead times, such as the 10-minute-timestep that is required for DST.

Research objective

In this research, we aim to fill the knowledge gap on the performance of independent ML algorithms on pre-defined visibility categories at multiple locations on relatively short lead times. We investigate the performance of ML algorithms in forecasting pre-defined visibility categories at two locations at Amsterdam Airport Schiphol. We introduce a simple, easy-to-implement ML algorithm, the Random Forest Classifier, and a more complex and time-sensitive algorithm, the Temporal Fusion Transformer. By developing these algorithms, we aim to improve the current forecast of visibility conditions and, with that, improve the accuracy of capacity predictions. Therefore, our main research question is:

How do machine learning algorithms perform in forecasting pre-defined visibility categories for two locations at Amsterdam Airport Schiphol, on lead times up to 4 hours, with timesteps of 10 minutes?

We will compare the performance of these models to the quality of the current forecast. We will also train both models as a NWM post-processing technique to assess whether the models perform better or worse as an independent model rather than a NWM post-processing technique.

To reach the objective, the following questions will be answered:

1. How does a Random Forest Classifier perform in forecasting pre-defined visibility categories for two locations at Amsterdam Airport Schiphol, on lead times up to 4 hours, with timesteps of 10 minutes?
2. How does a Temporal Fusion Transformer perform in forecasting pre-defined visibility categories for two locations at Amsterdam Airport Schiphol, on lead times up to 4 hours, with timesteps of 10 minutes?
3. How does the performance of each model change when including NWM data as input, e.g. using the model as a post-processing technique?

Hypothesis

Since studies have shown that machine learning algorithms based on observations can produce accurate visibility forecasts (Boneh et al., 2015; Bartok et al., 2022), we hypothesize that ML algorithms can predict fog events more accurately than the current NWM-based forecast. We expect the Random Forest Classifier to show the best results, since Castillo-Botón et al. (2022) found the best forecasting results using ensemble-model-based algorithms, especially Random Forest models.

However, we also hypothesize that it is likely that using ML as a post-processing technique could show better results than using an ML algorithm independently, since most research has demonstrated the potential of ML as a post-processing technique.

Since fog is generally a short-lived phenomenon, we hypothesize that the ML models are well capable of predictions visibility conditions on short timescales such as 10 minutes. We suspect that the models perform better on shorter timescales than longer timescales.

Report structure

The remainder of the report is structured as follows: In Chapter 2, we will discuss background information on the topic. The background information will cover the introduction of the study area, fog formation, how fog influences aviation operations and delays, Schiphol's current fog forecast and machine learning principles. In the last section we will also elaborate further on earlier research on machine learning and visibility forecasting. We describe our data in Chapter 3 and the methods in Chapter 4, followed by the results and their discussion in Chapter 5. We discuss limitations and recommendations for future research in Chapter 6 and finish the paper with a conclusion in Chapter 7.

2 Background information

This chapter forms a detailed explanation of the concepts referred to in this study. We first introduce our study area, Amsterdam Airport Schiphol. Then, we discuss atmospheric visibility, fog formation processes, and climatological patterns of fog at Amsterdam Airport Schiphol. Thereafter we elaborate on the impacts of low visibility circumstances on aviation in general and Amsterdam Airport Schiphol in particular. Finally, we discuss the principles of machine learning and how machine learning has been applied to visibility forecasting in the past.

2.1 Visibility

This research focuses on the forecasting of atmospheric visibility. Atmospheric visibility is defined as 'the ability to see and identify prominent unlighted objects by day and prominent lighted objects by night' (Federal Aviation Administration). We can divide visibility into horizontal visibility and vertical visibility. Horizontal visibility is commonly measured in meteorology as the meteorological optical range (MOR). MOR is defined as the distance over which a beam of light travels before its luminous flux is reduced to 5% of its original value (MetOffice). In contrast, vertical visibility measures the distance a person can see vertically from the earth.

2.1.1 Horizontal Visibility

The most crucial phenomenon influencing horizontal visibility is fog. In meteorology, we define fog as a situation where the MOR is below 1000 meters (KNMI). However, in aviation, a common threshold for fog is 1500 meters (Schiphol, 2016). To forecast visibility accurately, we must be able to forecast fog accurately. Understanding what processes contribute to its formation is essential to understanding why forecasting fog is challenging.

Fog formation

Fog develops when air saturates with water vapor, and the water vapor condenses to form tiny droplets. These droplets block the optical path via scattering and absorption of light, reducing visibility (e.g. Izett et al., 2018). There are several mechanisms that can cause the air to reach saturation, making some locations and seasons more sensitive to fog. The following sub-chapter will give a short overview of the different types of fog and their formation processes (Hang et al., 2016; Román-Cascón et al., 2019; Lakra & Avishek, 2022). These processes can be broadly categorized based on how saturation is achieved: by cooling, adiabatic expansion, adding moisture, or by movement of saturated air masses.

1. When cooling occurs by removal of heat
    i. *Radiation fog*. This type of fog forms when air near the ground cools rapidly, which typically occurs during clear nights with little wind. This situation allows the ground to radiate heat back into space, cooling rapidly (radiative cooling). The cooled ground cools the air above, lowering the air's temperature to reach dew point.
    ii. *Valley fog* is closely related to radiation fog. Colder air usually settles in valleys overnight, so the dew point is reached first at the lowest elevations. This can make certain areas more sensitive to fog if located at a lower elevation. Especially in combination with clear nights, these areas often experience fog in the early mornings.
    iii. *Advection fog*. When a warm, moist air mass moves over a colder surface, the warm air loses heat to the cold air. The warm, humid air reaches the dew point to form fog. This type of fog is common in coastal regions. *Sea fog* is a type of advection fog. It occurs above locations where warm and cold ocean currents meet.
2. When cooling occurs by adiabatic expansion

   i. *Upslope fog*. This fog forms when air with high relative humidity is moved upslope. This causes the air parcel to reach the dew point due to adiabatic cooling.
3. When water vapor is added to an air parcel
   i. *Steam fog*. In contrast to advection fog, this type forms when a cold air mass moves over a warmer water surface. The temperature difference causes water evaporation from the surface, increasing the water content in the air parcel above. The evaporated water vapor eventually leads to air saturation, causing fog.
   ii. *Frontal fog*. If rain, originating from warm air, falls through a colder air mass at a frontal zone, the precipitation can evaporate into this colder layer. Saturation is reached as evaporation increases dew point temperature and decreases absolute temperature.
   iii. *Evaporation* or *mixing fog*. When two parcels of different temperatures mix, it can lead to the mixed parcel becoming saturated. The resulting temperature after mixing can be low enough to reach the dew point, even though one or both parcels were unsaturated. An example is when one parcel is relatively warm and humid while the other is cool and dry.
4. When saturated air changes location
   i. *Cloud-base lowering fog*. Essentially, cloud-base lowering fog is a type of fog that does not develop on-site but rather gets classified as fog as it changes vertical position. A base of low-stratus clouds is not much different from fog other than that it is located at an altitude and not at the surface. The base of this cloud can descend, and when it reaches the ground, it will be classified as fog (Román-Cascón et al., 2019).

Fog evolution

Most fog types persist or grow through the same processes that caused their initial formation. However, if fog layers are exposed to clear skies, additional radiative cooling can enhance fog layer growth (Yang et al., 2023), known as fog-top radiative cooling. This enhanced radiative cooling results in thickening of the fog layer, in the same way that radiative fog is formed. This process dominates in thicker fog layers. In shallow layers, it is primarily the radiative cooling from the ground that promotes growth (Duynkerke, 1999).

Opposite to radiative cooling, clear skies have the opposite effect when fog occurs during the daytime. Direct sunlight then causes radiative heating of both the fog layer as well as the surface underneath, resulting in dissipation of the fog (Dione et al., 2023). If middle- or high-level clouds are present, they block this direct sunlight, preventing dissipation. Long-lasting, heavy day-time fog events therefore often occur with clouds at middle or high levels (Guo et al., 2021).

Fog dissipation

Fog dissipation is often caused by turbulent mixing. In principle, a stable inversion layer promotes the persistence of fog. Moderate wind speed can promote fog formation for some fog types, like upslope or evaporation fog, because moisture is transported towards the fog layer. However, if turbulence increases, it causes the inversion layer to mix, redistributing heat and moisture. This often causes breakup and dissipation of the fog layer (Dione et al., 2023). Similarly, high wind speeds also have this effect.

Fog-top radiative cooling, described above, can also have a dissipating effect on fog. As the fog layer continues to grow, the decrease in temperature at the top destabilizes the inversion layer, increasing turbulent mixing with the layer above. The weakening of the inversion layer and the increased turbulence often cause breakup of the fog layer. However, in some situations, this turbulence can also cause saturated air to be entrained into the fog layer, promoting growth (Li & Pu, 2024).

2.1.2 Vertical Visibility

Opposite to horizontal visibility, vertical visibility is most commonly determined by clouds. The vertical visibility is, therefore, often equal to the height of the lowest cloud layer. In aviation, this is often referred to as the cloud 'ceiling', or 'cloud base height (CLB)'. In unique situations, vertical visibility can also be determined by other phenomena, like smoke. Clouds generally form on larger spatial and temporal scales than fog, and therefore vertical visibility is less influenced by short-term fluctuations (Bony et al., 2015).

2.1.3 Challenges in Forecasting Visibility

Numerous studies have identified that Numerical Weather Models (NWM) produce fog forecasts that often perform significantly worse than the prediction of other variables like precipitation (Zhou et al., 2012; Izett et al., 2018; Bergot & Koracin, 2021). The cause of this low performance is related to the nature of fog in several ways.

As appears from the processes dominating fog formation, evolution and dissipation, fog is highly dependent on many different meteorological variables, like temperature, wind speed and air pressure. The dependency on a large number of variables makes the parameterization of fog formation in weather models difficult. More specifically, Román-Cascón et al. (2012) argue that the role of turbulence in the formation and dissipation of fog is poorly understood.

Not only meteorological variables, but also environmental factors play an essential role in fog occurrence. Local differences in land use or vegetation influence ground temperatures, directly influencing fog formation (Duynkerke, 1999). Additionally, due to the correlation with air temperature, fog often quickly dissipates in urban areas as a result of the urban heating effect (Gautam & Singh, 2018).

The dependencies on many variables also cause fog to occur on very small spatial and temporal scales. This makes modelling fog more challenging, as NWM generally operate on larger scales, making them incapable of capturing the short-lived, local nature of fog. Specifically, the models struggle to accurately model the time of onset and dissipation of fog (Steeneveld et al., 2015). Research shows that a high resolution is necessary to accurately forecast fog in NWM, even on vertical scales (van der Velde et al., 2010).

Many studies confirm that fog forecasts by NWM require improvement. Oppositely, forecasts for vertical visibility are often quite accurate (Inoue et al., 2015), but there is much less research on their quality.

2.2 Impact of Visibility on Aviation

To understand the relevance of accurate visibility forecasting, it is important to highlight the effects of low-visibility conditions. Today's modern society greatly relies on highly efficient transport systems. As the world's population keeps growing, the dependence and demand for transport systems increases (Shafer & Victor, 1997). One of the most critical sectors providing mobility is the aviation industry. The following section will give an overview of the leading causes and consequences of delays in aviation and how this affects operations at Schiphol Airport. We will also discuss measures to mitigate delays at Schiphol and the details and quality of the current fog forecasts at Schiphol.

2.2.1 Delays and Costs

The aim for efficiency mainly drives the aviation industry. Since airports often operate close to their maximum capacity, the sector is sensitive to delays. Between 2015 and 2024, over 20% of the scheduled US commercial flights experienced delays, of which 25-30% caused by weather (Bureau of Transportation Statistics, 2024). The FAA estimates the resulting economic losses to be $33 billion for the year 2019 (Federal Aviation Administration, 2022). Besides economic effects, delays also greatly influence passenger satisfaction. Flight delays are often the primary source of passenger complaints. In January 2023, over 37% of the complaints issued in the US concerned delays, cancellations or deviations (Office of Aviation Consumer Protection, 2023). Finally, delays also result in considerably higher CO_2 emissions, thus negatively impacting the environment (Dissanayaka et al., 2019) . For all the reasons mentioned, airlines and airports continuously aim to minimize delays as much as possible.

Delays are, for the most prominent part, caused by weather (Coy, 2006). In Europe, over half of all flight delays result from adverse weather conditions (Rodríguez-Sanz et al., 2022). Specifically, extreme weather events such as thunderstorms, snow and fog are the main factors contributing to delays. A study showed that these events increased the probability of delays by over 25% (Pejovic et al., 2009). Reduced visibility, caused by either fog, precipitation, or low clouds, greatly influences delays since reduced visibility directly influences an airport's capacity. As an example, half of all weather-related accidents were caused by reduced visibility at a Canadian airport (Leung et al., 2020).

2.2.2 Airport Capacity

The main factor determining an airport's capacity is the *separation* between aircraft (see info-box). This determines how many aircraft can land or take off in a specific timeframe. Many factors, like aircraft size, runway length, and the weather, determine the separation. One of the meteorological factors greatly influencing aircraft separation is visibility.

Why do we need separation between aircraft?
Safety is the number one priority in the aviation industry. Enough spacing between aircraft means safety can be guaranteed. It allows one aircraft to exit a runway, before another plane uses it to take off or land. In case incidents do occur, air traffic control and following aircraft will have sufficient time to respond. Furthermore, the power produced by an engine of an aircraft taking off or landing causes turbulence ("wake"). Wake turbulence can be dangerous for aircraft following. Certain weather events call for extra separation between aircraft.

Many airports have a navigation system called the Instrument Landing System (ILS), allowing pilots to land in all visibility conditions. The worse visibility gets, the more pilots need to rely on ILS rather than their view from the cockpit. Therefore, we require higher precision from the ILS system in conditions with worse visibility. An aircraft on the ground can block or disturb the signal of the ILS system for the subsequent aircraft following. Therefore, the preceding aircraft needs sufficient time to move sufficiently far away from the ILS system so that it does not interfere with the signal for the following aircraft. This ensures the highest precision possible from ILS, necessary for low-visibility operations (see info-box) (Dijkstra, 2024).

ILS: principle, categories and protection areas

ILS, or "Instrument Landing System" is a system designed to allow for aircraft to land during low-visibility conditions, for example at night or during fog. The system transmits radio signals to the aircraft to provide guidance in its path towards the runway (also known as the glideslope).

In its most basic form, the system provides guidance until the aircraft is 200 feet (61 m) above the ground. If the pilot is not able to see the runway at that point, it has to cancel the landing and perform a 'missed approach'. This height is also known as a 'decision height': the height where the pilot decides to continue the landing or not. However, ILS systems can be upgraded in precision. This decreases decision height. ILS systems are divided in three categories, CAT I (basic), CAT II and CAT III. For CAT II the decision height is 30 m, for CAT III it can even be decreased to 0 m, allowing for landings with no visibility at all (ICAO, 2018b) .

Objects can block or interfere with the radio signal emitted by the ILS system and therefore decrease the accuracy of the aircrafts position in the glideslope. For this, so-called "protection areas" are set up around the ILS antennas. These areas need to be free of obstacles to allow for an accurate ILS performance during CAT II or CAT III operations. It is likely that these protection areas include taxi-ways. Therefore, aircraft landing need to be sufficiently separated to allow for the preceding aircraft to exit the protection area. An example of protection areas at Schiphol is given in Appendix A.

Besides ILS, low visibility also decreases runway capacities since non-parallel runways can no longer be used. Runways that cross are a risk of collision when pilots do not have a clear view of other traffic during go-arounds. Since *go-arounds* are likely to occur during low visibility conditions, crossing runways are never used. The thresholds for using crossing runways are 1) horizontal visibility of at least 5 kilometers and 2) a cloud base no lower than 2000 feet.

2.3 Amsterdam Airport Schiphol

For our research, we focus on the airfield of Amsterdam Airport Schiphol, located in the western part of the Netherlands, in western Europe (Figure 2.1a, 2.1b). The center of the airport is located at 52.308 N, 4.764 E. The airfield has a total of 6 runways (Figure 2.1c).

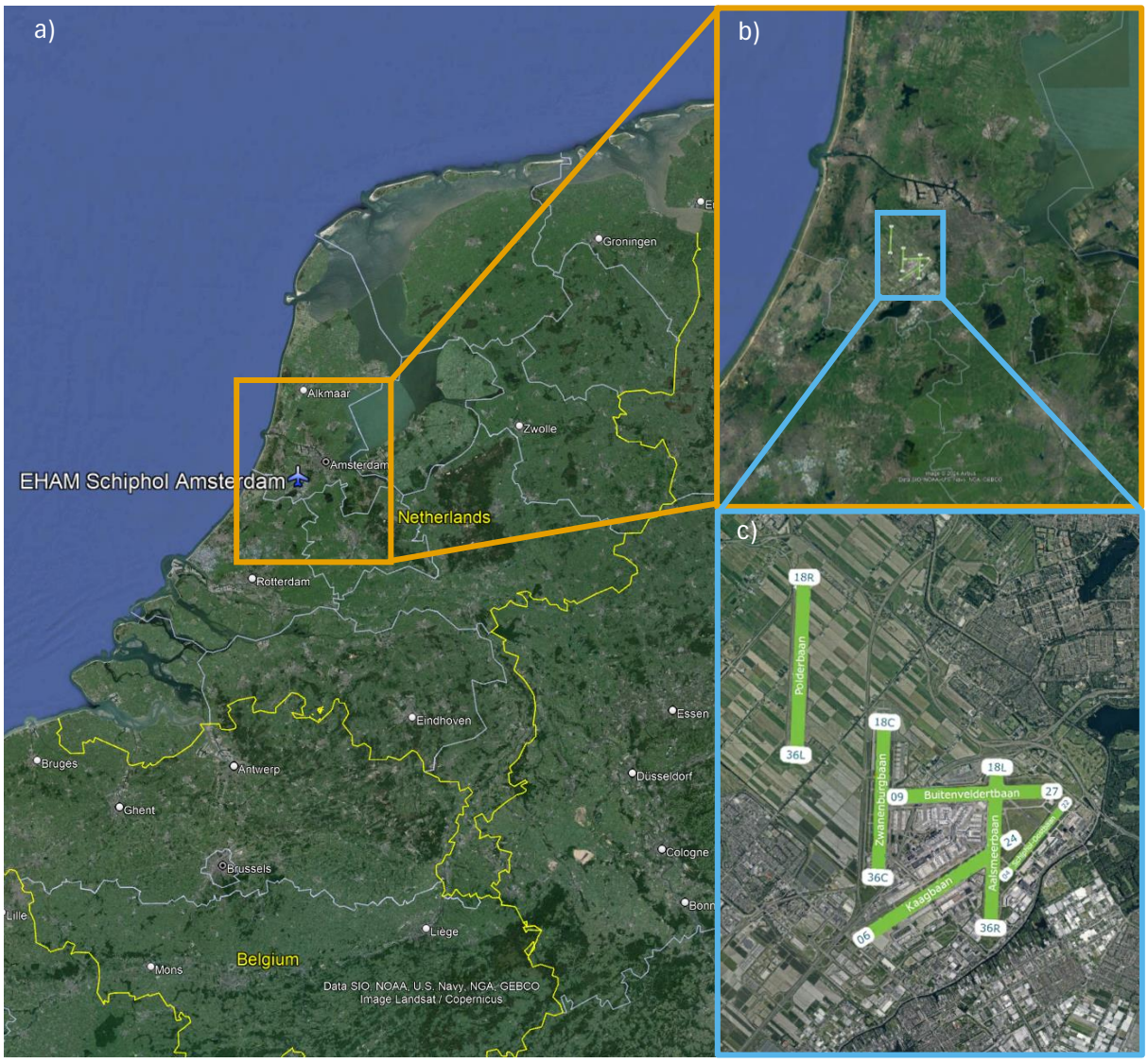

Figure 2.1: Illustration of the location of the study area on a map of the Netherlands (a), an illustration of its location in relation to the western part of the Netherlands (b) and an illustration of the entire airfield and its runways in green (c). Source satellite images: Google Earth. Source runway image: GMAP

Between 2003 and 2020, fog occurred 42 days of the year on average at Schiphol (KNMI). The most common fog types are radiation fog and advection fog (Cannemeijer & Stalenhoef, 1977; Izett et al., 2019). Fog occurs two to three times more often in winter and fall than in summer and spring (Figure 2.2). In colder seasons, conditions are usually more favorable for fog formation. This can be related to the formation of radiation fog. Longer nights allow for more radiative cooling and favor the formation of radiation fog in the early mornings.

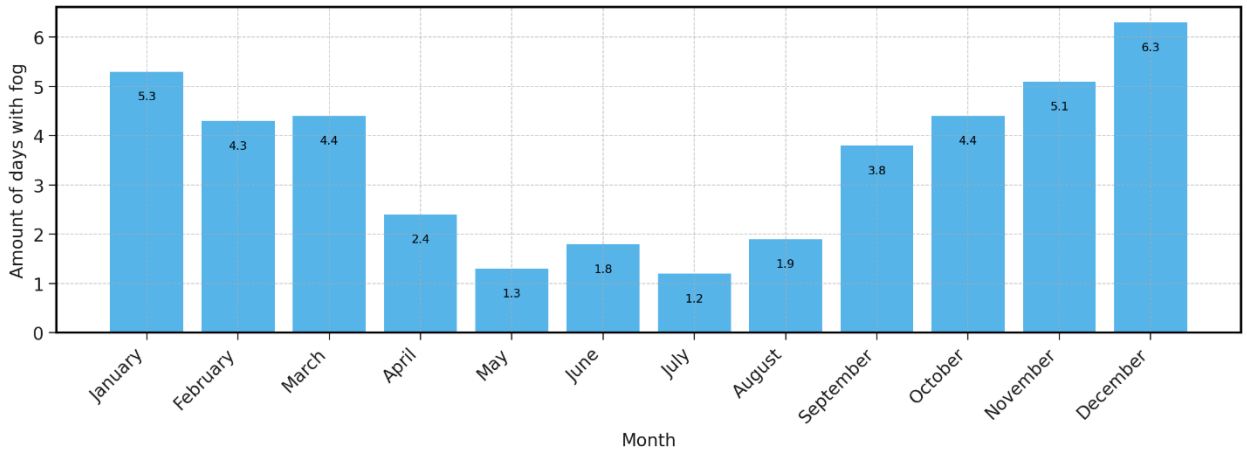

Figure 2.2: The seasonal cycle of fog occurrence at Schiphol for the period of 2003 through 2020. Each bar shows the amount of days on which fog occurred during that month, averaged over all years. A day is classified as 'foggy' when the horizontal visibility dropped below 1000 m at any point in time on that day. Data source: KNMI

In recent years, fog has occurred less often (Figure 2.3). Between 1971 and 2000, Schiphol experienced 74 foggy days per year (KNMI). This implies fog frequency at Schiphol decreased by almost 57% in the most recent climatological period (1991-2020) compared to two periods prior (1971-2000).

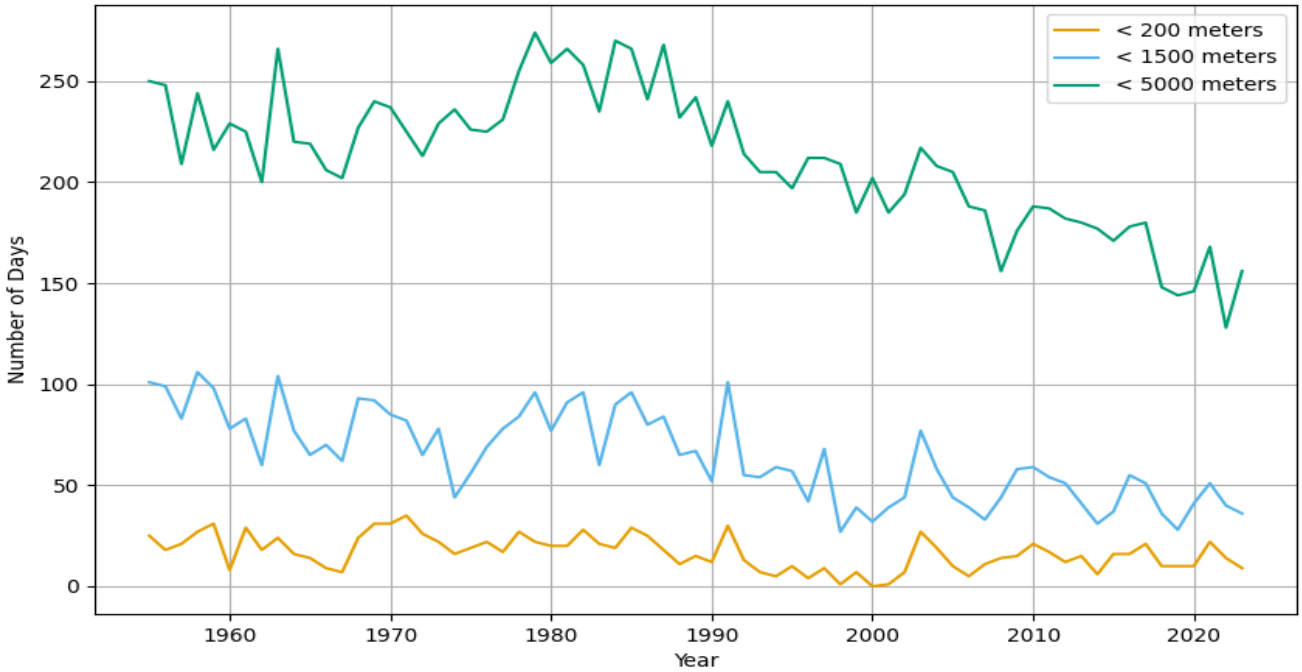

Figure 2.3: The trend of the amount of days per year with reduced horizontal visibility at Schiphol Airport between 1955 and 2023. Amount of days are shown for horizontal visibility below 5000 meters (green), below 1500 meters (blue) and below 200 meters (orange). Measured between runway 22 and runway 27. Data source: KNMI.

This downward trend has not been a development in recent years. Studies dating from as early as the 1960s discussed decreasing fog frequencies, including at Schiphol Airport. Early studies suggest improved soil drainage or changes in large-scale weather patterns as causes for this decline (Tonkelaar, n.d.). The latter hypothesis was supported by van Oldenborgh et al. (2010), who also noted the downward trend was surprisingly similar to that of the atmospheric sulfur dioxide concentration. Some studies argue the downward trend is caused by climate change (Hingmire et al., 2022), but they often lack significant evidence.

2.4 Low Visibility Procedures at Amsterdam Airport Schiphol

Even though foggy days are becoming less frequent at Schiphol, their effects can still be substantial, as explained in subchapter 2.2. To operate smoothly during low visibility, LVNL (Air Traffic Control the Netherlands) has set up BZO phases (Low-visibility procedure phases, Dutch: BZO (Beperkt Zicht Omstandigheden (Schiphol, 2016)) which are based on horizontal and vertical visibility (Table 2.1 / Appendix A). The runway capacity decreases for each phase because operating in a higher ILS category is necessary. The runway with the lowest visibility determines the category.'Reduced visibility classes' are defined as BZO phases 'Marginal' and 'A', 'B', 'C', and 'D', while 'low-visibility classes' are BZO phases 'A', 'B', 'C' and 'D' only.

Thresholds in horizontal visibility and vertical visibility determine ILS categories and BZO phases. However, as seen in Table 2.1, below 1500 meters, the horizontal visibility threshold parameter changes to 'Runway Visual Range', also known as 'RVR.' RVR is a meteorological parameter explicitly designed for aviation. It considers visibility but combines this with the brightness of the runway lights, as these can be adjusted. RVR therefore denotes the distance at which a pilot can identify the runway lights (ICAO, 2018a). Since lights make the runway more noticeable, the RVR is often higher than the horizontal visibility.

Table 2.1: Reduced-visibility-procedure categories and their thresholds for visibility and cloud base height, the maximum amount of aircraft movements during the phase, and the climatological occurrence of the phase. Table adapted from Wolters et al. (n.d.).

Phase	Horizontal visibility threshold	Vertical visibility threshold	Max. aircraft movements per hour	Occurrence (%)
Good	Visibility > 5000 km	Visibility > 1000 ft	108	87
Marginal	Visibility < 5000 m	Visibility < 1000 ft	108	10
BZO A	RVR < 1500 m	Visibility < 300 ft	80	2
BZO B	RVR < 550 m	Visibility < 200 ft	74	2
BZO C	RVR < 350 m		47	0-1
BZO D	RVR < 200 m		36	0-1

A separate case is the one of the "Polderbaan", Schiphol's longest runway located almost 7 kilometers taxiing distance from the main terminal (Figure 2.3: runway 18R/36L). This runway is intensively used: in 2016, it handled 24% of all take-offs and 38% of all landings at Schiphol Airport (Schiphol, 2017). Due to its relatively low noise pollution, the runway is also one of the two primary runways to use during the night (23h – 06h) (bewoners aanspreekpunt schiphol, 2022). On about 10 of the 42 foggy days at Schiphol, the Polderbaan experiences low-visibility conditions while the rest of the airfield has good visibility. This is because the location of the Polderbaan is more susceptible to fog formation than other runways (Izett et al., 2019).

First, the Polderbaan is situated at a lower elevation causing cold air to settle in this location (see Background Information - Visibility). Also, the distance between the runway and the main field is so large, that the urban heating effect by warm buildings and asphalt is expected to be marginal. Instead, the runway is surrounded by grasslands that retain more water and cool quickly. Finally, turbulence of jet engines also causes fog to dissipate (Izett et al., 2019). Even though the Polderbaan is extensively used, there is only one taxiway beside the runway. Other runways are surrounded by many more taxiways leading to different runways and areas of the airfield. Therefore, the turbulent mixing effect of taxiing aircraft is much smaller at the Polderbaan.

Since BZO phases are determined by the runway with the lowest visibility, this would imply that a strict BZO phase would have to be applied to the entire airfield if visibility conditions are bad at the Polderbaan. In 2022, LVNL changed its procedures so that the Polderbaan could operate in a different BZO phase from the rest of the airport, creating two 'BZO areas': West' and 'Center' (Figure 2.4). Separating the Polderbaan from the rest of the airport increased the airport's capacity by 20-35% (LVNL, 2022). Therefore, a separate visibility forecast for each BZO area is required.

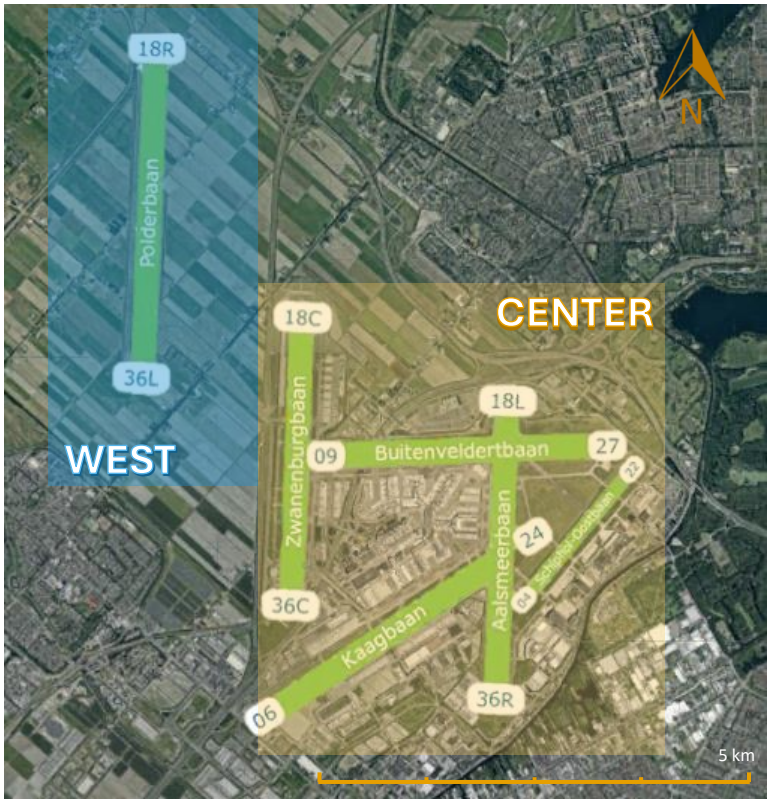

Figure 2.4: The runways of Schiphol with their direction at each end of the runway. The orange area is the BZO phase area 'Center' (orange) and the Polderbaan is located solely in BZO phase area 'West' (blue). Source: Luchthaven Schiphol/Google Earth/GMAP

Decision Support Tool

During low-visibility operations, timely adaptation and mitigation for delays is crucial. LVNL uses a model called the Decision Support Tool, or DST, to forecast runway capacities at Schiphol for every 10 minutes in the coming 4 hours. It includes meteorological forecasts and real-time data on aircraft en route to Schiphol. If the model predicts airport capacities lower than the amount of aircraft to land during a particular hour, an air traffic controller can interfere and 1) choose to use a runway combination with higher capacity, if available, 2) instruct traffic to hold over a location somewhere along its route or 3) instruct traffic to delay its departure so that it will land in a timeslot with higher capacity (Dijkstra, 2024). The model also accounts for the number of go-arounds during a specific period. More go-arounds also decrease a runway's capacity, as the same plane will have to come in for landing twice.

To maximize efficiency, the airport aims to operate close to its maximum capacity. A demand that is higher than the capacity will lead to delays. This occurs when aircraft are not regulated enough – the controller applies procedures that are not strict enough for the actual conditions. The opposite scenario is also undesirable. When the controller applied too restrictive procedures because the current conditions were not as limiting as expected, the airport could have handled more aircraft. These aircraft were delayed or diverted to reach the restricted capacity, which wasn't necessary. This results in unnecessary costs and delays. Accurate meteorological forecasts are crucial for the airport to operate close to its maximum capacity continuously.

Current Fog Forecasts for Schiphol Airport

The current visibility forecast for LVNL is produced by the Royal Netherlands Meteorological Institute (KNMI) specifically for the airfield (Figure 2.5). In the forecast, an ML algorithm named Quantile Regression Forest (QRF) produces probabilities of the different BZO phases at Schiphol and their determining factors CLB, RVR and visibility. The model uses features from the NWP model 'HARMONIE' and observations. We will refer to this forecast as the HARMONIE-QRF model.

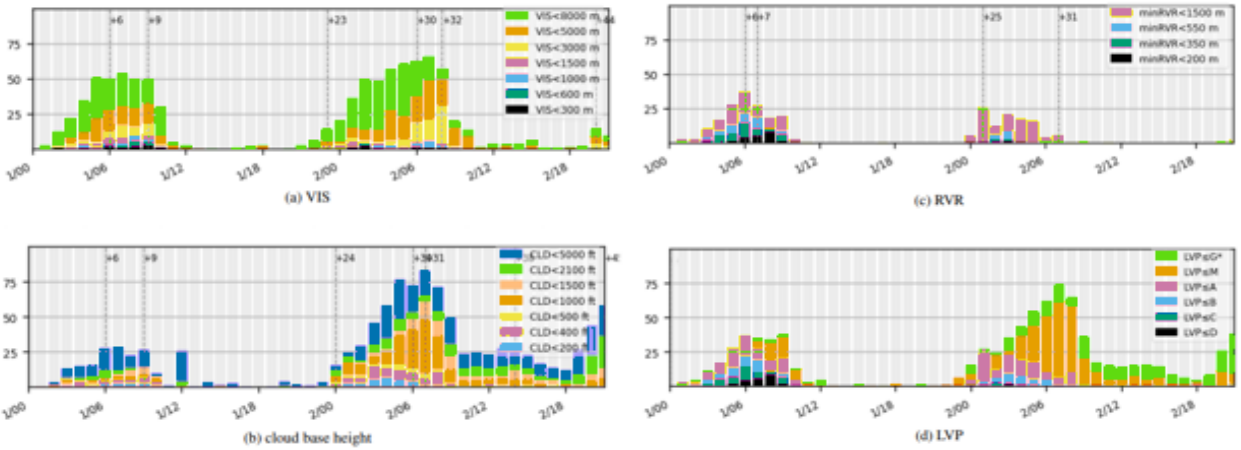

Figure 2.5: An example of a probability forecast from the KNMI, showing target features visibility (a), cloud base height (b), RVR (c) and LVP (Low Visibility Procedures) (d). The LVP phases correspond to the BZO phases. From Wolters et al. (n.d.)

The HARMONIE-QRF algorithm produces higher accuracy compared to the previous forecast (Wolters et al., n.d.). However, there is still room for improvement since it underestimates most BZO phases (Figure 2.6). In contrast, BZO phase D (LVP D) is mostly overestimated. This phase is especially poorly forecasted, partly caused by the meager number of observations.

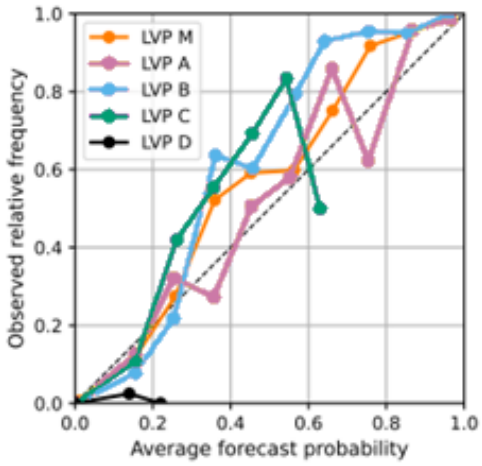

Figure 2.6: Reliability diagram of the forecast with lead time +1 hr. Data points above the diagonal suggest an underestimation of a low-visibility event. From Wolters et al. (n.d.)

When comparing the model's skill to a climatological forecast (Figure 2.7), we can observe that initial scores are high, but the forecast quality decreases with increasing lead time. BZO phase 'D' shows a negative BSS, indicating it is worse than the baseline. The baseline depicts the score of a climatological forecast. In a climatological forecast, the probability of future fog occurrence is the same as its historical occurrence.

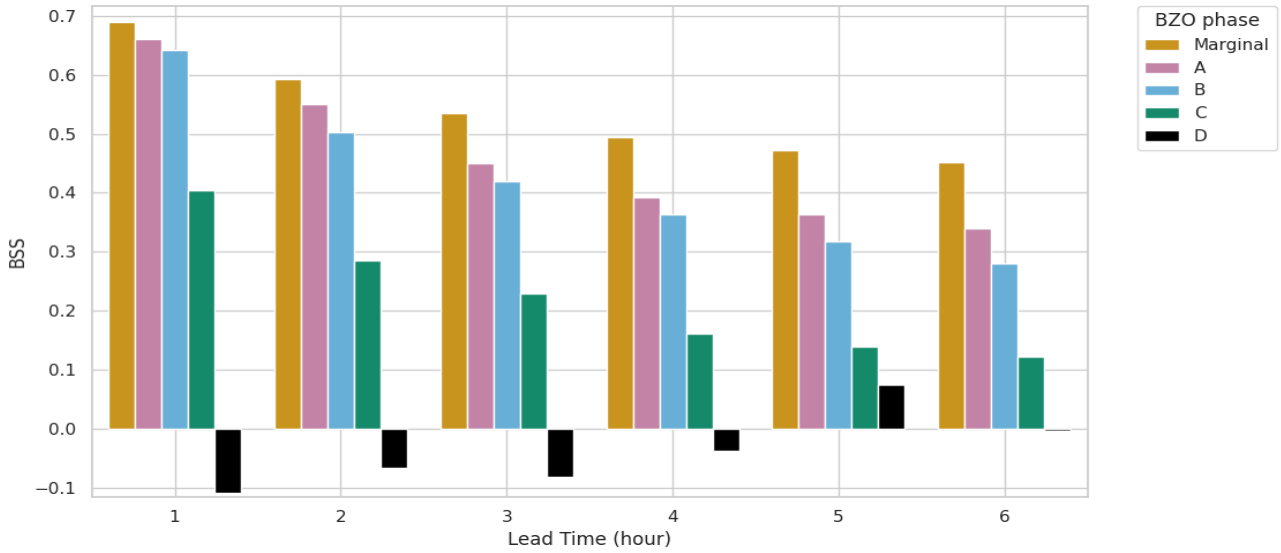

Figure 2.7: The Brier Skill Score of the HARMONIE-QRF forecast for lead times up to 6 hours. BSS > 0 indicates better performance than climatology, BSS < 0 indicates worse performance than climatology. From Wolters et al. (n.d.)

Prior to HARMONIE-QRF, forecasts were made solely by the HARMONIE model (Figure 2.8) and updated where necessary by a meteorologist. When starting this research, the latter forecasts were still in use. Therefore, we will also benchmark our results to the quality of the HARMONIE model in forecasting visibility. We calculated these scores using the HARMONIE-AROME Cy43 re-forecast for the period of October 1st, 2020, through September 31st, 2023 (see Data – NWM data). Additional metrics of this forecast can be found in Appendix B. The plots indicate the models can capture class 'Good' well, but substantially underperform in other classes, specifically class 'A' and 'B'.

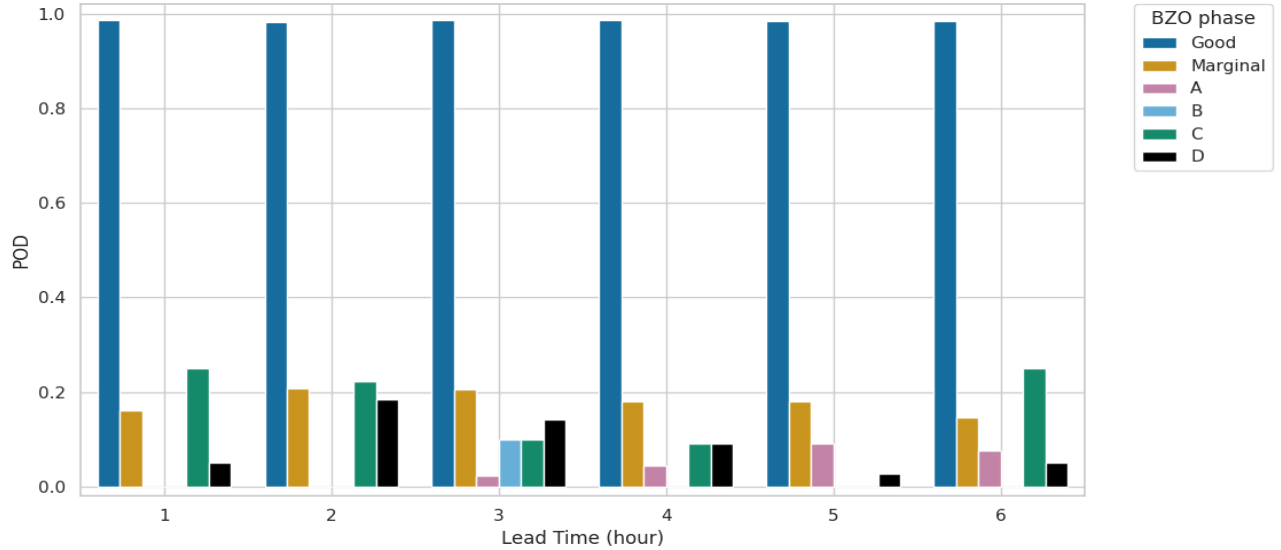

Figure 2.8: Probability of Detection of the horizontal visibility classes of the HARMONIE cy-43 model. 0 indicated worst possible score, 1 indicates best possible scores. Some scores missing since there were no instances of the class at that specific lead time.

2.5 Machine Learning

Earlier studies showed that machine learning (ML) algorithms produce visibility forecasts of similar or higher quality than NWM (e.g. Bartoková et al., 2015; Colabone et al., 2015; Guijo-Rubio et al., 2018; Miao et al., 2020; Bartok et al., 2022; Salcedo-Sanz et al., 2022). Hence, using ML could likely improve the quality of Schiphol's current forecast. In this section, we will explain the basic principles of ML, different types of ML algorithms and the challenges that arise when using ML for imbalanced datasets and time series. Finally we discuss previous research on ML and visibility forecasts.

2.5.1 Principles of Machine Learning Algorithms

ML is an Artificial Intelligence (AI) type. At its core, ML involves algorithms that learn from historical data, by identifying patterns or relationships within this data (El Naqa & Murphy, 2015). In turn, the algorithm uses these identified patterns to make predictions for new, unseen data. The learning process is usually divided into two types: regression, where the model predicts continuous values, and classification, where the model predicts outputs into categories.

A limitation of ML algorithms is that they often need large amounts of data to capture all relations between the variables in the dataset accurately (Mastorakis, 2018). These amounts of data are not always available, especially when considering rarely occurring situations like low visibility, making the application of ML on visibility forecasts challenging (Bari et al., 2023). The exact amount of necessary data points highly depends on the type of algorithm and data.

Target and predictor variables

Within ML algorithms, we make a distinction between targets and predictors (Balali et al., 2020). The output variable, or the variable the algorithm tries to predict, is called the *target*. In light of this study, the target would be visibility. The model tries to find relationships of the target variable to other variables, which we call *predictors*. For example, predictors for visibility could be temperature, wind speed and air pressure, as visibility is dependent on these variables.

Input data: training and testing

The dataset that is presented to an ML algorithm should be split into different subsets: one for training the model and one for evaluating its performance (Goodfellow et al., 2016). The *training set* is the data used to train the model, while the *test set* is used to evaluate the model's performance on unseen data. It is important to split these datasets, so that the model will not be evaluated on data that it has already seen. Therefore, the two sets must be independent of each other, e.g., do not have overlapping data points (Thomasson, 2023).

Hyperparameters

Finally, *hyperparameters* are the settings and configurations of the model that allow for output optimization. These settings are set before the model training begins and, thus, do not change throughout the training process. Hyperparameters differ significantly between different kinds of ML algorithms. Optimizing hyperparameters is known as *hyperparameter tuning*, where the model's performance is evaluated for each hyperparameter setting to find the most optimal combination.

2.5.2 Training and Fitting Machine Learning Algorithms

Training an ML algorithm is done by presenting a dataset to the model, and telling it what variable is the target and which variables are predictors. The model then tries to identify meaningful relationships between each predictor and the target. However, a situation can occur where the model relates small

changes in a predictor to the target, that are not meaningful in real life. For example, the data may contain a small spike in relative humidity which was followed by a fog event. The model might mistake this spike as the cause of the fog event, and assume such a spike always leads to fog. This is called 'overfitting'. When a dataset is noisy, like large real-world datasets (e.g., meteorological observations), the risks of overfitting are relatively large (Bentéjac et al., 2021). Similarly, underfitting can occur, where the algorithm fails to capture the complexity of relationships between variables.

One way to prevent under- and overfitting is cross-validation, which includes splitting the entire dataset into subsets. The model is trained for multiple iterations, using a different subset each time. This method is especially suitable for relatively small datasets, as the complete dataset is used to train the algorithm.

2.5.3 Decision Trees

The Random Forest algorithm is a type of Decision Tree (Figure 2.9). Decision trees are one of the most used ML algorithms (Breiman, 1984). A DT model works just as a real-life decision tree. At each level of the tree, the dataset is split into subsets based on a threshold of a particular variable. Eventually, the splitting reaches a final subset, where the dataset is classified into pure 'leaf nodes'. The final forecast is based on the output of all leaf nodes.

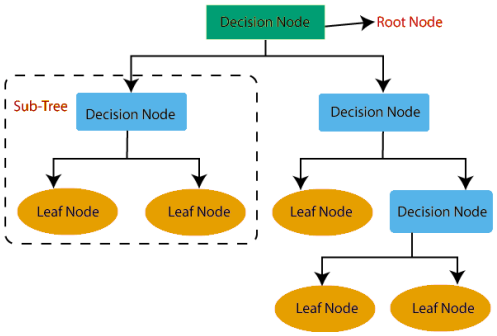

Figure 2.9: the principle of a decision tree. The decision nodes split into new nodes until a final stage is reached at the leaf nodes. Source:

2.5.4 Neural Networks

The Temporal Fusion Transformer is a model based on a neural network. A neural network is a type of ML algorithm designed to function like the human brain. The models are composed of layers, often called 'neurons'. In each layer, a transformation is performed on the input data, enabling the model to recognize patterns. If the pattern is significant for the outcome, a weight is assigned to this layer. The final output is a joint prediction based on all layers. If there are two or more layers in a neural network, they are classified as a 'deep learning' network. The first trainable neural network was presented by Rosenblatt (1957).

2.5.5 Imbalanced Datasets

The quality of input data is of great importance to the performance of any type of ML algorithm (Mastorakis, 2018). When some cases occur significantly more often than others, forming majority and minority classes, the dataset is so-called "imbalanced" (Chawla et al., 2004). Classifying algorithms are likely to ignore the minority classes, resulting in high performance scores for the majority classes but low scores for the minority classes(Liang, 2013). In many cases, the minority classes are the classes of interest (Moniz et al., 2017), making the accurate performance for these classes even more critical. Since fog rarely occurs, visibility datasets are often highly imbalanced, with many samples in good visibility categories. To avoid bias of the models towards good visibility, these datasets often need to be balanced.

There are multiple methods for dealing with imbalanced datasets. A popular strategy is to resample the input dataset (Moniz et al., 2017). In resampling, a dataset can be either under- or oversampled. A combination of both is also possible. Undersampling involves decreasing the size of the majority class by removing class samples, while in oversampling, more minority class samples are created (Liang, 2013). A risk of undersampling is that the resulting dataset is too small to train the ML algorithm accurately. On the other hand, a risk of oversampling is that the synthetically created samples do not add any new information for the model (Cateni et al., 2014). The majority class is still more broadly defined, making it easier for the model to generalize this class than any minority class.

2.5.6 Time Series Forecasting

The natural order and temporal correlation between variables often pose an extra challenge when using classifiers to make time series forecasts. Ordinary classification algorithms are usually incapable of adequately capturing this temporal correlation (Ruiz et al., 2021).

First of all, time series often display *concept drift* (Widmer & Kubat, 1996; Chawla et al., 2004), which describes changes in the distribution of the target variable in relation to the predictor variables (Moniz et al., 2017). In other words, the target will not always have the same relation to the predictor variables, as this relation depends on time. For this reason, 'date' and 'time' are often included in the algorithm as predictor variables. For visibility, concept drift can be described by the high seasonality, thus dependence on the time of the year, and the fact that fog often occurs during specific times of the day.

Furthermore, time series are often *autocorrelated*, implying that future values depend on past values. A way to address autocorrelation is to include lagged versions of the predictor variables (Surakhi et al., 2021).

Finally, when it comes to multi-step forecasting, a distinction can be made between *direct* and *recursive* forecasting (see Figure 2.10), which both have advantages and disadvantages. In direct forecasting, a different model is trained to independently forecast each future time step. Oppositely, in recursive forecasting, only one model sequentially forecasts each future time step, taking its own predictions as preceding time steps. An advantage of recursive forecasting is that it only requires one model compared to the multiple models needed for direct forecasting. However, recursive forecasting has the risk of *compounding errors*, as each step is affected by the error in the previous prediction. The best method depends on the situation, such as the input data type and computational power available.

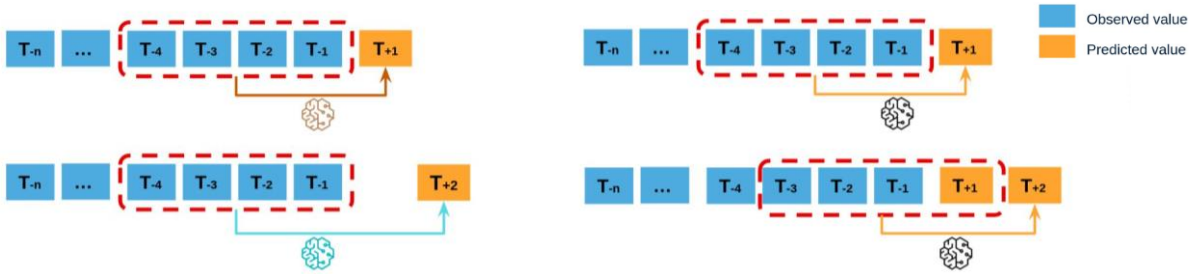

Figure 2.10: direct forecasting (left) versus recursive forecasting (right). Source: Spektor, 2023.

2.5.7 Sequence Models

Oppositely, sequence models are ML models specifically designed for sequential data like timeseries. These models focus on learning temporal dependencies of variables in the dataset, making them well-suited for tasks like timeseries forecasting.

The TFT is an example of a sequence model. However, this model incorporates more advanced mechanisms, like *attention layers* and *gating mechanisms*. The attention layers help the model 'pay more attention' to the most important parts of the input data by assigning weights to more relevant features or time points. The gating mechanisms act like control switches that regulate how much information is passed to the model, to avoid overfitting. The TFT is specifically powerful for forecasting at multiple forecasting horizons, as well as forecasting with multiple input and output variables. A unique aspect of this model is that you can input both known and unknown feature variables, making it very useful for real-life practices like meteorological forecasting.

TFT's use loss functions to optimize the model. A loss function measures how well the model's predictions align with observation, after which the model can adjust internal parameters to improve. Lower loss values indicate better performance, which is why models usually aim to minimize loss. The most common

loss function is the 'Quantile Loss' (Koenker & Bassett, 1978), which is used in regression tasks and predicts quantiles of the target variable. For classification, a common loss function is 'Cross Entropy' (Shannon, 1948), which measures how far the model's forecasted probability (between 0 and 1) is from the observation (1 for observed, 0 for not observed). A variant of the Cross Entropy function is the 'Focal Loss' function (Ross & Dollár, 2017), specifically designed to handle class imbalance. The function introduces an extra scaling factor, in which it down-weights loss on easy-to-predict samples while it increases the weight on harder, usually rare samples. If this scaling factor does not sufficiently handle the class imbalance, an additional weighting factor can be applied to each class.

2.5.8 Machine Learning and Visibility Forecasts

Many studies have shown that many machine learning types are capable of forecasting visibility conditions in different configurations. The following section provides an overview of earlier studies on machine learning and visibility forecasting. Specifically, since our research focusses on applying a decision tree and neural network to the problem, we focus on studies that were conducted on these two types of models.

Model performance is often evaluated with metrics or skill scores like the Probability of Detection (POD) and the Critical Success Index (CSI) which both score the model's performance between 0 (low) and 1 (perfect). The False Alarm Rate (FAR) shows the ratio of incorrectly forecasted fog events between 0 (low) and 1 (high). Finally, the Brier Score is a score that assesses the skill of a probabilistic forecast, with 0 being the perfect score and higher values indicating worse performance. For formulas on the calculation of these metrics, please refer to the Methods chapter.

The very first application of an ML algorithm for fog forecasting was by Koziara et al. (1983), using a linear approach to post-process NWM forecasts, outperforming competing models at the time (CSI 0.42 – 0.45, Brier Score 0.27 – 0.34). Several years later, Tag and Peak (1996) also applied a machine learning approach to marine fog forecasting (POD 0.543). They concluded machine learning is a possible viable tool for visibility forecasting, but only when used with "reliable data and sufficient cases of known outcome" (Tag & Peak, 1996).

Decision Trees

It wasn't until 2001 that the first decision tree was applied to the problem. Wantuch (2001) showed promising results of a short-term decision-tree-based model, which was quickly adopted by the Hungarian Meteorological Service. However, decision-tree forecasting didn't make its breakthrough until many years later. In 2015, Bartoková et al. (2015) proposed using a decision tree to improve the output of NWM for fog forecasting in Dubai, focusing on short-term predictions up to several hours. When the algorithm was used as a NWM post-processing technique it showed the highest results (POD 0.88).

Many other studies also used decision trees as post-processing techniques: Herman and Schumacher (2016) improved NWM forecasts on lead times of 1 to 12 hours for visibility forecasting in aviation, and Bari (2018) implemented similar methods to improve visibility forecasting on lead times of 3 to 12 hours. Kim et al. (2021) included ECMWF air pollutant data and focused on short-term predictions of 1 to 3 hours. The decision-tree-based fusion model by Yu et al. (2021) includes regional model output and also forecasted visibility for aviation on short lead times of 30 minutes to 3 hours (POD 0.13 – 0.72, FAR 0.28 – 0.67, CSI 0.22 – 0.50) and Kim et al. (2022) combined synoptic observations with NWM output to forecast visibility conditions up to 36 hours. Negishi and Kusaka (2022) used decision trees as part of a hybrid approach to improve radiation forecasts on short lead times (1 to 3 hours; CSI 0.382) in Japan, and Parde

et al. (2022) post-processed NWM output on lead times of 6 to 12 hours (POD 0.95, FAR 0.43, CSI 0.55, Brier Score 0.13). Finally, Thomasson (2023) applied decision trees as a post-processing technique on an NWM grid over Denmark, focusing on lead times of 1 to 12 hours (POD RFC 0.338).

Besides being used as a post-processing technique, decision trees have also been shown to be effective in independently forecasting visibility. For example, Dewi et al. (2020) used decision trees to predict fog at airports over 1 to 3 hours (POD 0.57 – 0.77). Han et al. (2021) used decision trees in forecasting fog dissipation on lead times of 1 to 3 hours (CSI 0.82 – 0.96) and Ortega et al. (2019) focused on the classification of visibility categories on lead times of 1 to 6 hours. Zhen et al. (2023) included decision trees in their fusion model for real-time fog prediction, focusing on lead times from 30 minutes to 12 hours (CSI 0.42 – 0.89). Penov and Guerova (2023) focused on high-accuracy, short-term forecasts for airports (POD 0.3, FAR 0.17, CSI 0.27) and Ohashi & Hara (2024) applied decision trees to forecast the expansion of morning sea fog over 1 to 6 hours (POD 0.733 – 0.848, Brier Score 0.125 – 0.83). Liu (2024)applied decision trees to forecast winter fog, focusing on complex terrain, over a horizon of 3 to 12 hours (POD 0.885, FAR 0.542, CSI 0.535). Almeida et al. (2023) showed the effectiveness of a decision tree in dynamically improving predictions with evolving weather conditions over 6 to 12 hours (POD 0.93 – 0.99, FAR 0.32 – 0.34, CSI 0.63 – 0.67) and Zhang et al. (2022) successfully applied decision trees over longer timescales (1 to 24 hours).

These studies show that decision trees have proven to be successful at forecasting multiple types of fog and visibility over multiple forecasting horizons both independently and as an NWM post-processing technique.

Neural Networks

Besides decision trees, many studies have focused on using neural networks for visibility forecasting. For example, Fabbian et al. (2007) used a classification neural network at Canberra Airport in Australia, Colabone et al. (2015) applied an artificial neural network for back-propagation on visibility forecasts with a 95% success percentage, Durán-Rosal et al. (2018) applied evolutionary neural networks on fog prediction at Valladolid airport in Spain and Miao et al. (2020) tested Long Short-Term Memory networks in China. Liu (2024)also applied neural networks to forecasting of winter fog over complex terrain and Negishi and Kusaka (2022) applied neural networks to radiation fog prediction. Other algorithms were also tested, including Support Vector Regressions (SVR), Extreme Learning Machines (ELM) (Cornejo-Bueno et al., 2017), ordinal classifiers (Guijo-Rubio et al., 2018) and fuzzy logic-based predictors (Miao et al., 2012) (POD 0.66-0.96). Boneh et al. (2015) applied a Bayesian network at Melbourne Airport, which successfully forecasted fog (POD 0.80-0.94) and is now operational.

The first case fusion-based neural network 'Temporal Fusion Transformer' being used to forecast visibility conditions was presented by Wehrli et al. (2024) at the EMS conference in 2024. Wehrli used the TFT to forecast visibility quantiles and found promising results of Brier Scores close to 0.01 on a forecasting horizon of 180 minutes. The results showed the TFT is a model well-capable of forecasting visibility conditions.

3 Data

The scope of this research is to design ML algorithms to forecast visibility conditions at Amsterdam Airport Schiphol. We train two models, namely the RFC and TFT. As described in the previous chapter, both models have been shown to be capable of accurately forecasting visibility conditions. In this section, we describe the data we used to train the models. We discuss the data source, any general pre-processing, and some dataset statistics.

3.1 Observational Data

We used observational data from 20 FD12P Present Weather Sensors alongside the runways of Schiphol, made available by the KNMI (Figure 3.1). The dataset runs from January 1st, 2012, to March 31st, 2017, with time stamps of 1 minute. All variables in the dataset are presented in Appendix C. Each measurement station has its own unique Location-ID, consisting of 'VAM', indicating the station is located in Amsterdam, followed by the two-digit runway number and a possible L/C/R indicating the left, center or right runway, and positional indicator, describing at what position along the runway the station is located. The positional indicators are 't' (touchdown zone), 'm' (middle zone), 'n' (north), 's' (south), 'e' (east) and 'w' (west).

Figure 3.1: the location of the 20 measurement stations along the runways of Schiphol Airport. FD12P Present Weather Sensors shown in blue, and the Automated Weather Sensor in orange. Stations are named according to their position relative to the closest runway, 't' meaning touchdown zone, 'm' middle zone, 'n' north, 's' south, 'e' east and 'w' west.

Not every sensor's dataset contained all variables of interest (for variables used please refer to the Methods chapter). For each BZO (Beperkt Zicht Omstandigheden) area, we created a dataset by averaging the available recordings of its sensors. All sensors weighed equally in the averaging, as we assumed that the sensors are relatively evenly distributed in both areas and the center of the BZO area is representative of the entire area. This resulted in two datasets with the same variables, one for each BZO area.

The dataset also contained many missing values. To impute these values, we used a Random Forest regressor. This is a common practice for imputing missing data in meteorological datasets (Gorshenin & Lukina, 2021). The imputation was executed using the Python class 'IterativeImputer' with its estimator 'RandomForestRegressor', both available in the scikit-learn library. The dataset's statistics showed minimal changes after imputing missing values, so we can assume the data structure was well preserved. These statistics, as well as the amount of missing values per variable are shown in Appendix D.

Table 3.1 quickly confirms a spatial pattern in low visibility circumstances. The stations in BZO area West, along the Polderbaan, (runway 18R or 36L, highlighted in orange) recorded the most low-visibility circumstances of all stations. Note that even though the VAM18Cm27 station does have a relatively low percentage of 'good' visibility recordings, the percentage of recordings in all BZO phases is still lower than the posts in BZO area West because the amount of recordings in 'Marginal' is relatively high. The table also clearly shows the highly imbalanced nature of the dataset, as low-visibility conditions only make up about 1-3% of all recordings.

Table 3.1: Distribution of the number of samples across the different visibility phases for all measurement stations, ordered by relative time spent in good visibility. The total sample amount is given for each station, along with the percentages of samples recorded in each visibility category. Stations in BZO area West are highlighted in orange. VAM18Ct was excluded since it did not contain any visibility recordings.

Location-ID	Total 1-min Samples	BZO D (%)	BZO C (%)	BZO B (%)	BZO A (%)	Marginal (%)	Good (%)
VAM18Rtw	2626232	0.79	0.45	0.29	0.96	8.38	89.14
VAM18Cm27	763554	0.35	0.28	0.18	0.86	9.12	89.20
VAM18Rms	2626217	0.73	0.47	0.28	0.95	8.27	89.30
VAM18Rmn	2626063	0.76	0.46	0.28	0.95	8.20	89.36
VAM36Lt	2616044	0.75	0.50	0.29	0.88	7.83	89.75
VAM18Rte	2626172	0.69	0.45	0.29	0.83	7.88	89.86
VAM27t	2607455	0.53	0.34	0.23	0.82	7.91	90.17
VAM36Ct	2625178	0.51	0.34	0.21	0.76	7.60	90.58
VAM18Ctpws	2625519	0.47	0.36	0.20	0.74	7.56	90.68
VAM06t	2626249	0.49	0.36	0.22	0.71	7.54	90.69
VAM18Lt	2626373	0.49	0.33	0.21	0.72	7.50	90.75
VAM22t	2626285	0.44	0.32	0.20	0.71	7.39	90.93
VAM27m	2604578	0.44	0.28	0.18	0.64	7.51	90.95
VAM09t	2595391	0.31	0.27	0.18	0.71	7.52	91.00
VAM36Rm	2626308	0.48	0.32	0.21	0.65	7.17	91.17
VAM36Rt06	2626294	0.46	0.34	0.21	0.67	7.15	91.19
VAM24t	2626323	0.39	0.29	0.17	0.59	7.19	91.37
VAM06m	2504369	0.36	0.30	0.20	0.65	7.06	91.44
VAM36Cd36R	2620475	0.43	0.31	0.19	0.64	6.90	91.54

3.2 NWM Data

The NWM data used for this research is the HARMONIE-AROME Cy43 reforecast made available by KNMI in the KNMI Data Platform (Tijm, 2024). The dataset contains meteorological variables for the near-surface boundary layer (up to 300 m). Appendix C shows a table with a complete overview of the variables in the dataset. The data spans from October 1st, 2020 to September 30th, 2023. The domain spans from 49.000 N to 56.002 N latitude and 0.000 E to 11.281 E longitude and has a resolution of 2 kilometers. We selected the domain cell containing the measurement station from which we collected the visibility observations (AWS data). The center of this grid cell is located at 52.312 N, 4.785 E.

Besides the latitude and longitude dimensions, the dataset contains two height dimensions. For temperature, the dimension has levels 0, 2, 50, 100, 200, and 300 (meters). For wind variables, the dimension has levels 0, 10, 50, 100, 200, and 300 (meters). We used 2-meter values for temperature variables and 10-meter values for wind variables.

A new run is produced every 6 hours, at 00 UTC, 06 UTC, 12 UTC and 18 UTC. Each run has a lead time of 60 hours with an output timestep of 1 hour. The first 6 hours of each run were maintained and merged to form a continuous time series.

3.3 AWS Data

The NWM dataset did not have the same time range as our observational dataset. For this reason, we had to collect additional observations for the time range of the NWM dataset. One of the sensors at Schiphol is an Automated Weather Sensor (AWS), which automatically makes recordings every minute. Hourly data of this sensor is made available by the KNMI from 1951 onwards (KNMI). We extracted data from this sensor from October 1st, 2020, to September 30th, 2023. This data contained the target variable (visibility) as well as additional observations of meteorological variables, so that the models could also be trained on historical observations besides NWM data. These variables were chosen to best match the variables in the observational dataset. A complete overview of the variables in this dataset is given in Appendix C.

3.4 Runway Visual Range and Cloud Base Height Data

Runway Visual Range (RVR) was not present in any dataset and cloud base height (CLB) was not present in the AWS dataset. These variables were necessary to train models to predict RVR and vertical visibility respectively. LVNL (Air Traffic Control the Netherlands) maintains records of these variables. This data was recorded by the same Present Weather Sensors as the observational data.

CLB was added to the AWS dataset as an extra column. RVR was added to the observational dataset and AWS dataset. Recordings were not regular, so their values were coupled to closest minute. If two RVR recordings were within a minute, their value was averaged. Furthermore, RVR was only recorded for active runways, and only once horizontal visibility drops below 1500m. Therefore, RVR values were imputed into the horizontal visibility data whenever they were available. This version of the dataset was only used when predicting 'RVR'. When the target was 'horizontal visibility', RVR was left out the dataset.

4 Methods

This subchapter describes our approach of training and testing the machine learning (ML) algorithms. In general, models were designed to forecast visibility into different categories based on the BZO (Beperkt Zicht Omstandigheden) phases (see Background Information – Low visibility Procedures at Amsterdam Airport Schiphol), over a forecasting horizon of 4 hours with timesteps of 10 minutes. This forecasting horizon was chosen to align with the forecasting horizon of the Decision Support Tool (DST; see Background Information – Low Visibility Procedures at Amsterdam Airport Schiphol), so forecasts could be easily implemented in DST.

We repeated all experiments for three targets: 'horizontal visibility', 'RVR' (Runway Visual Range) and 'vertical visibility'. The models classified output of horizontal visibility and RVR into the following classes:

G. 5000 m ≤ visibility
M. 1500 m ≤ visibility < 5000 m
A. 550 m ≤ visibility < 1500 m
B. 350 m ≤ visibility < 550 m
C. 200 m ≤ visibility < 350
D. 200 m > visibility

Or, similarly for vertical visibility, values were classified in the following classes:

G. 1000 ft ≤ visibility
M. 300 ft ≤ visibility < 1000 ft
a. 200 ft ≤ visibility < 300 ft
b. 200 ft > visibility

In the rest of this chapter we discuss the training and evaluating processes for both the Random Forest Classier (RFC) and the Temporal Fusion Transformer (TFT). For each model, we will first describe the pre-processing steps taken to prepare the dataset. Then, we describe the general set-up of the model, followed by the different experiments we executed to test the model's performance. Finally, we go into depth on the different evaluation metrics we used to assess the model's performance.

We used a High-Performance Computing Cluster (HPC Cluster) to store data and execute scripts. Scripts were written in the programming language Python. Data handling and structuring using libraries Pandas (McKinney, 2010) and NumPy (Harris et al., 2020) . Calculations were mainly done using SciPy (Virtanen et al., 2020) and plots were made using Matplotlib (Hunter, 2007). Any other specific libraries used will be described in each step of the modeling process.

4.1 Random Forest Classifier

4.1.1 Pre-processing steps

Dataset Transformations

The observational dataset was transformed into a regular Pandas DataFrame instead of a time series. We added sine-transformed variables for 'Month', 'Date' and 'Time' as predictor variables, creating the variables 'Time_Sin' and 'Month_Day_Sin'. This ensured the RFC would still consider temporal and seasonal dependence. Although 'Years' were not defined as predictor variables, we kept the column so that data could still be ordered chronologically, and cross-validation could be performed based on years.

Lagged Features

To account for the autocorrelation in our observational dataset (see Background Information – Machine Learning), we included lagged versions of each predictor. By adding lags, the model can recognize dependencies on previous timesteps.

To determine the optimal number of lags, we calculated autocorrelation for visibility below 5000 meters (reduced visibility classes; 4.1.a and 4.1.c) and below 1500 meters (low-visibility classes; 4.1.b and 4.1.d). Autocorrelation values range from -1, indicating opposite correlation, and 1, indicating perfect correlation. The shaded region indicated the 95% confidence interval. If the autocorrelation is above the confidence interval, we can say that the timestep is significantly correlated with the timestep X-minutes before (the amount of minutes being the lag).

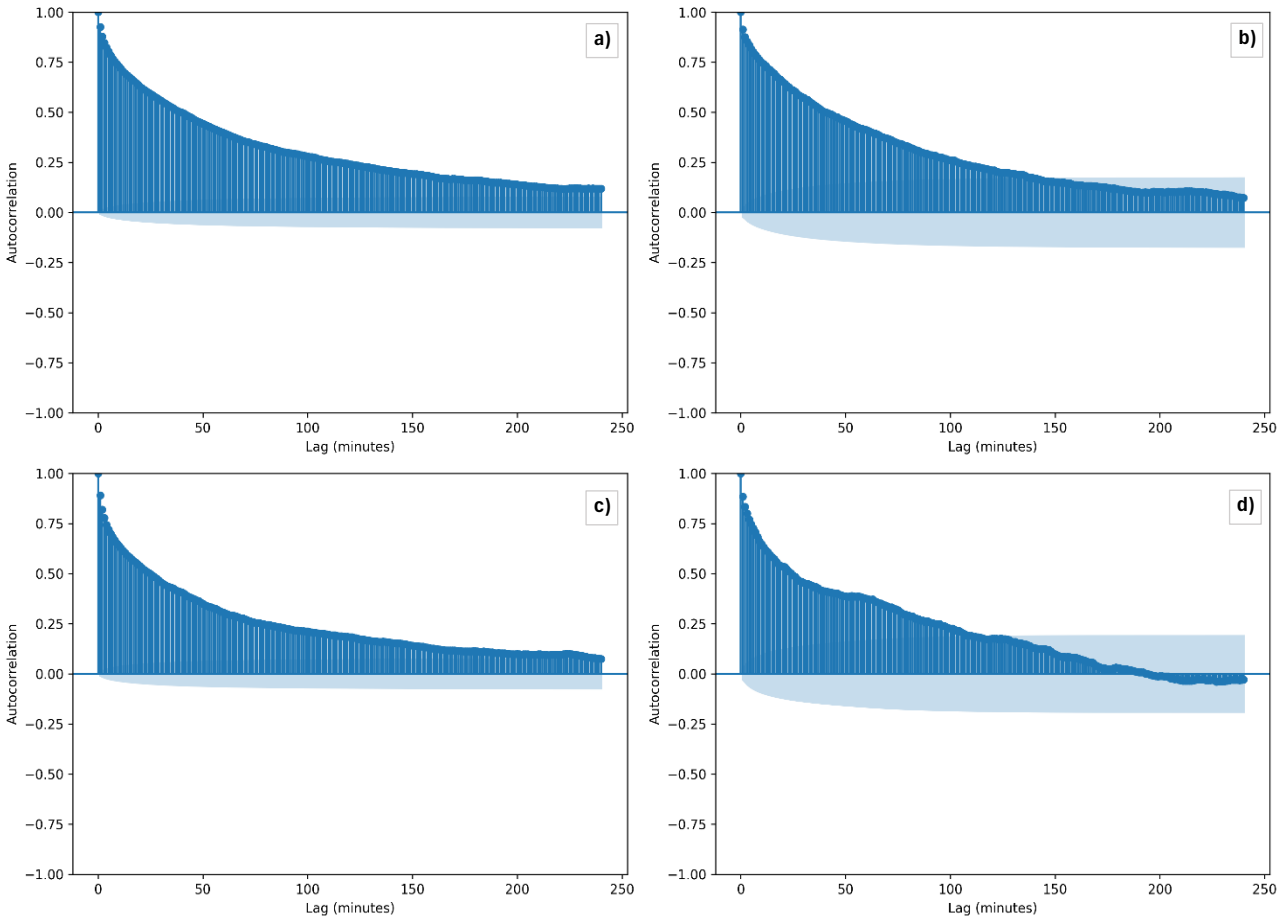

Figure 4.1: Autocorrelation for visibility values below 1500 meters (above) for BZO area West (left) and Center (right), and for visibility values below 800 meters (below) for BZO area West (left) and Center (right). Autocorrelation showed by bars and 95% confidence intervals in shaded blue. Autocorrelation above the confidence interval can be considered significant.

The figures show that autocorrelation remains significant for BZO West but drops below the confidence interval at about 130 minutes for BZO Center. To keep consistency between our models, we applied lags of up to 240 for both locations, in 10-minute increments.

We only used lags in our ML models that would be available in real time. If T is the current time, the forecast of T+10 utilized all lags up to T-240. However, for T+20, lag T-10 would not be available, so the model used all lags from T-20 to T-240. This extends over all lags.

Balancing techniques

The previous chapter on Data showed that our observational dataset was highly imbalanced over the visibility categories. Therefore, we used resampling techniques to balance the dataset. Moniz et al. (2017), found the combination of so-called "SMOTE" oversampling and random undersampling (SMOTE-RUS) performs best for Random Forest regression on time series. Castillo-Botón et al. (2022) found this method successful for Random Forest Classifiers.

SMOTE, or Synthetic Minority Over-sampling Technique, is the most common oversampling technique. It was developed by Chawla et al. (2002) and involves creating synthetic cases in the minority classes. The technique creates new samples by considering existing samples' characteristics. New samples are created on the 'line segments' between existing samples' positions. As the name implies, random undersampling includes randomly removing samples from a majority class. No characteristics or values of variables are considered (Castillo-Botón et al., 2022).

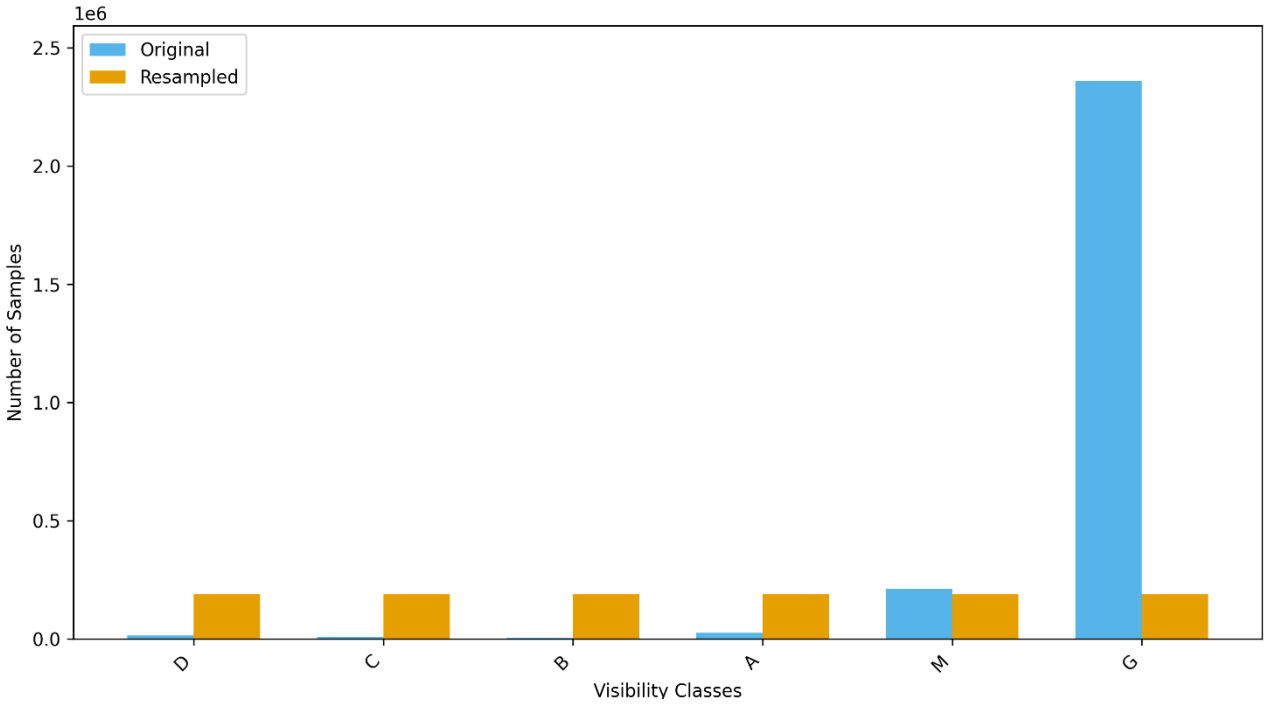

Figure 4.2: The distribution of horizontal visibility in the original dataset (blue) and the resampled dataset (orange)b) of 1-minute observations. The original dataset was highly imbalanced with the majority class being 10 to 100 times larger than the minority classes. In the resampled dataset, all classes have the same size.

We implemented both techniques using the Imbalanced-Learn Python library from Scikit-Learn (Pedregosa et al., 2011). The strategies were very effective for both horizontal visibility and vertical visibility (horizontal visibility: Figure 4.2; vertical visibility: Appendix C).

Cross-validation

As explained in the Background Information chapter, a dataset must be split into a training and test dataset. To ensure the entire dataset was used to train the model, we set up cross-validation. Herman and Schumacher (2016) state that at least three years of data is necessary to capture the relationship between visibility and the predictor variables for hourly datasets. We followed this recommendation since we used hourly data in the NWM post-processing experiments. Therefore, following Thomasson (2023), the cross-validation contained five folds, each of one year (see Figure 4.3). This implies that four years of data will be used to train the model in each iteration. The remaining year is used as the test set. This approach is justified by the climatology of the dataset, as all years show similar patterns in the occurrence of horizontal and vertical visibility (Appendix C). Furthermore, to avoid data leaking, we also follow Thomasson (2023) to imply a 72-hour gap, equal to 4320 1-minute timesteps, between each fold.

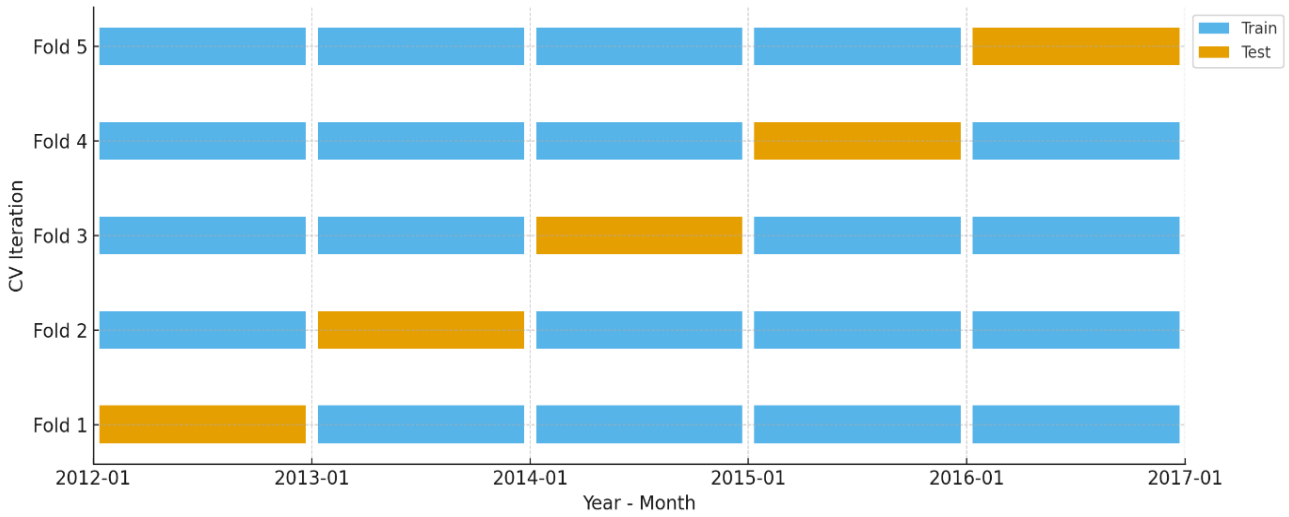

Figure 4.3: A schematic overview of the cross-validation scheme. Folds are divided by the separate years. In each fold one year forms the test set (orange), while the other four years are the training set (blue).

4.1.2 Model set-up

For training the RFC, we followed a two-phase approach after Bartok et al. (2022). In this approach, a rule-based system first distinguishes between class 'Good' and reduced visibility classes, after which the RFC is only trained on the reduced visibility classes. This approach minimizes the bias of the model towards class 'Good'. A visual overview of the model is shown in Figure 4.4, where the input and output are shown in blue, with characteristics of the input and output data in lighter blue. The main model components in orange are the "Rule-based system" and the "Random Forest Classifier". In green are the different visibility categories. We used the RandomForestClassifier class, available from the Scikit-learn Python package, which is an open-source, easy-to-implement package for machine learning in Python (Pedregosa et al., 2011).

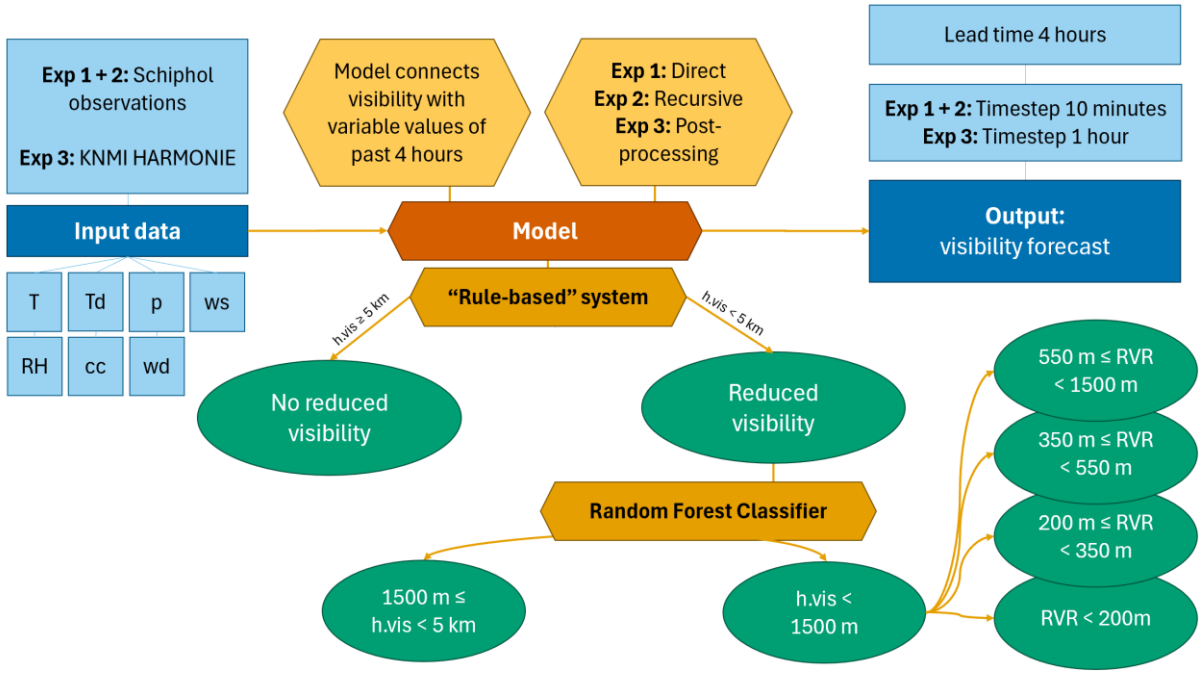

Figure 4.4: General example of the model setup with target 'RVR'. Input variables shown on the left (T = model components in the middle (orange), classified categories in green and output on the top right.

1. **"Rule-based" system:** Based on statistics of the variables in the dataset, determine whether a "No reduced visibility" event can be assumed. The minimum, maximum, lower 2.5% quantile, and upper 97.5% quantile of both no reduced visibility (h.vis ≥ 5 km) and reduced visibility (h.vis < 5km) are used. If "No reduced visibility" cannot be assumed, continue to phase 2.
2. **Random Forest Classifier:** The RFC is trained to classify classes 'Marginal', 'A', 'B', 'C' and 'D' based on the thresholds in visibility for the BZO categories (see Background Information – Low Visibility Procedures at Amsterdam Airport Schiphol).

4.1.3 Experiments

In this final section we describe the three experiments we executed to evaluate and compare different versions of the RFC. We discuss the variables used as predictors and different model settings. Each experiment was repeated for three targets: 'horizontal visibility', 'RVR' and 'vertical visibility'. The predictors used were based on earlier research on Random Forest Classifiers and visibility (Fabbian et al., 2007; Dewi et al., 2020; Bartok et al., 2022). For all experiments we used default hyperparameter settings of the RandomForestClassifier class from Scikit-learn (Pedregosa et al., 2011).

Experiment 1 – RFC-Direct

First, we trained a direct RFC, meaning the model trains predict each future timestep directly from the known values at timestep zero (see Background Information – Machine Learning). To achieve this, a separate classifier was trained for each timestep on the forecasting horizon. We trained a total of 24 models; one for each 10-minute timestep within 4 hours. The predictors are presented in Table 4.1.

Experiment 2 – RFC-Recursive

In the second experiment, we trained a recursive RFC. This model recursively predicts each timestep based on the previous one, and thus only consists of 1 model, rather than 24 separate ones. We executed this experiment to compare the performance of this simple, quick approach to direct RFC, which requires more computational power and time. The predictors are presented in Table 4.1.

Experiment 3 – RFC-HARMONIE

Finally, we also want to be able to compare our model's performance to that of a NWM post-processing technique, as these models were found to perform well (see Background Information – Machine Learning). Therefore, we perform an additional experiment where we use our RFC as a post-processing technique on the NWM model 'HARMONIE'. As predictors, we use the same set as in the first experiment, but with additional HARMONIE variables (Table 4.1).

Since the HARMONIE dataset did not overlap with our observations, our observational predictors and targets came from a different dataset: the AWS sensor (see: Data) and from LVNL (for cloud base height).

Table 4.1: An overview of the different predictors used for the three experiments executed for the Random Forest Classifier.

Experiment	Predictors	
1 – RFC-Direct	• 2-m 1-min average air temperature • 2-m 1-min average dew point temperature • 2-m 1-min average relative humidity • 2-m 1-min average wind speed • 2-m 1-min average wind direction • 1-min average rainfall intensity	• 1-min average surface air pressure • Cloud base height at time of observation • Month_Day_Sin • Time_Sin
2 – RFC-Recursive	*Same as Experiment 1 for T+10*	For recursive timesteps: • Class of previous timestep • Month_Day_Sin • Time_Sin
3 – RFC-HARMONIE	Observational variables: • 1.5-m air temperature at the time of observation • 1.5-m dew point temperature at the time of observation • 1.5-m relative humidity at the time of observation • 2-m wind direction, average over last 10 minutes of the hour • 2-m 1-hour average wind speed • Hourly global radiation • Hourly precipitation amount • Air pressure reduced to mean sea level at time of observation • Cloud base height at time of observation	HARMONIE variables: • 2-m air temperature at time of forecast • 2-m dew point temperature at time of forecast • 10-m wind speed at time of forecast • 10-m wind direction at time of forecast • Total cloud cover at time of forecast • 2-m relative humidity at time of forecast • Surface visibility at time of forecast • Surface air pressure at time of forecast

4.2 Temporal Fusion Transformer

4.2.1 Pre-processing

In the following subchapter we describe the general model set-up and experiments we executed for the TFT. Our goal was to assess the performance of the TFT in forecasting BZO phases using the Focal Loss function. This loss function was specifically designed for class imbalance (see Background Information – Sequence Models). Since our dataset was highly unbalanced, we thought the Focal Loss function would be a suitable loss function for our classification task.

In other fields of study, this function was successful in forecasting highly imbalanced classes without the dataset being resampled first. Therefore, to assess the pure performance of Focal Loss in handling class imbalance in visibility classes, we decided to not over- or undersample our datasets for training the TFT.

The TFT expects input data at the same intervals as the desired output. Since we wanted to create a forecast with timesteps of 10 minutes, the 1-minute observational data was aggregated to 10 minutes. We aggregated the data by taking the mean of all values within each 10-minute interval.

4.2.2 Model set up

To train the TFT, we used the TemporalFusionTransformer class in the Pytorch Forecasting libraries to set up the model (Paszke et al., 2017). These libraries are useful for our study because they are specifically designed for the easy application and training of deep learning models. We also executed hyperparameter tuning to find the most optimal set of hyperparameters for each experiment, using the Optuna hyperparameter optimization framework (Akiba et al., 2019). Optuna is particularly useful for hyperparameter tuning because it is quick and easy to implement. The best hyperparameters were defined on the combination that resulted in the lowest loss for performance on the test set. An overview of the best hyperparameters in each experiment can be found in Appendix G. For any variables or hyperparameters not mentioned in this section, we used default settings of the TemporalFusionTransformer class.

In the TFT, a predictor variables are categorized as either ‘continuous’ (numerical values), ‘categorical’ (categories) or ‘static’ (not changing over time). Moreover, a variable can be ‘known’ or ‘unknown’, depending on whether the values of this variable are known at the forecast lead time. An example of a known feature is ‘time of day’. An overview of all predictors in our model can be found in Table 4.3.

The TFT also allows for adding lagged variables as predictors. However, TFT is designed to be more aware of temporal dependencies through its attention heads (see: Background Information – Machine Learning). As this is the first time Focal Loss is being applied to this context, our aim was to investigate the performance of a baseline TFT. Therefore, we were interested in whether this attention heads mechanism was sufficiently capable of identifying temporal dependencies without lags.

The predictor variables differ slightly from the variables we used in the RFC models. For training the TFT we followed the predictors used by Wehrli et al. (2024) (Table 4.3), to allow for direct comparison with their TFT model. Specifically, grass temperature, 2-hour air temperature anomaly, and 12-hour accumulated precipitation are added. Wind direction is no longer considered.

4.2.3 Experiments

Experiment 1 – TFT-Focal-Loss

Our first experiment consisted of fitting the TFT with the Focal Loss function. Focal Loss is not included in the Pytorch Forecasting libraries by default. We manually set up the loss function by basing it on the MultiHorizonMetric class. The complete code for the Focal Loss function can be found in Appendix F.

Experiment 2 – TFT-Weighted

The Focal Loss function allows for additional weighting of classes, where class weights can be manually specified. In this experiment, we applied weight to less-abundant classes emphasize rare cases, to investigate whether this improves the model's capability of forecasting rare classes.

We decided to weight classes by either doubling (2), adding a higher order of magnitude (10) or both (20). The weights were assigned based on class-wise performance in Experiment 1. We chose this straightforward approach, because it makes it easier to observe how the added weighting affected model performance. The weights for both horizontal visibility classes and vertical visibility classes are shown in Table 4.2.

Table 4.2: The assigned class weights for experiment 2, per BZO phase, for both horizontal visibility and vertical visibility.

Target Visibility/BZO Phase	Good	Marginal	A	B	C	D
Horizontal Visibility	1	2	10	20	10	10
Vertical Visibility	1	2	10	20	-	-

Experiment 3 – TFT- HARMONIE

Finally, we also experimented with using the TFT as a NWM post-processing technique. Since the HARMONIE dataset is a forecast, we can consider them 'known' for future timesteps. Therefore, we included HARMONIE variables as known, continuous variables.

Table 4.3: The predictors for each experiment of the Temporal Fusion Transformer, per experiment and per predictor type.

	Predictors		
Experiment	**Known, categorical**	**Known, continuous**	**Unknown, continuous**
1 – TFT-Focal-Loss	• Season	• Hour_Sin • Hour_Cos • Day_Sin • Day_Cos • Time_idx	• 2-m 1-min average air temperature • 2-hour air temperature deviation • 2-m 1-min average dew point temperature • 1-min average grass temperature • 2-m 1-min average relative humidity • 2-m 1-min average wind speed • 1-min average surface air pressure • 12-hour precipitation sum • Cloud base height at time of observation
2 – TFT-Weighted	*Same as Exp. 1*	*Same as Experiment 1*	*Same as Experiment 1*
3 – TFT-HARMONIE	*Same as Exp. 1*	*Same as Exp 1.* + HARMONIE variables: • 2-m air temperature at time of forecast • 2-m dew point temperature at time of forecast • 10-m wind speed at time of forecast • Total cloud cover at time of forecast • 2-m relative humidity at time of forecast • Surface air pressure at time of forecast	• 1.5-m air temperature at the time of observation • 1.5-m dew point temperature at the time of observation • 1.5-m relative humidity at the time of observation • 2-m 10-min average wind direction • 2-m 1-hour average wind speed • Hourly global radiation • Hourly precipitation amount • Air pressure reduced to mean sea level at time of observation • Cloud base height at time of observation

4.3 Evaluation metrics

We will use several evaluation metrics to assess the models' performance. In this subchapter, we will explain each metric, how it is calculated, and what aspect of the model's performance it represents.

Many metrics are calculated using a so-called contingency table (Table 4.4). The columns show whether an event was observed (o=1) or not observed (o=0). Similarly, the rows indicate whether the event was forecasted (f=1) or not forecasted (f=o). Together, these possibilities make up the four cells in Table 4.4.

Table 4.4: A contingency table.

		Observed	
		Positive (o = 1)	Negative (o = 0)
Forecasted	Positive (f = 1)	True Positive (Hit, H)	False Positive (False Alarm, F)
	Negative (f = o)	False Negative (Miss, M)	True Negative (Correct Negative, CN)

The total number of samples N is retrieved by adding all cells together.

$$N = H + M + F + CN \tag{4.1}$$

Since our models handle multi-class problem, use the 'One-vs-Rest' approach for evaluation (Castillo-Botón et al., 2022; Thomasson, 2023). In this approach, we simplify the evaluation into a binary classification for each class. We treat the class of interest as the 'positive' class, while all other classes are 'negative'. This way, all metrics are calculated separately for each class.

4.3.1 Metrics for the Random Forest Classifier

1. Probability of Detection

The Probability of Detection, or POD, measures the fraction of actual events that were correctly forecasted in a specific class. It indicates the model's capability of correctly detecting the events within a class. It ranges from 0 to 1, 0 indicating the poorest performance and 1 the best possible performance. POD is a key quality measure in evaluating meteorological models (Roebber, 2009). The POD is given by:

$$POD = \frac{H}{H + M} \tag{4.2}$$

2. Accuracy

Finally, accuracy measures how often the model is correct in general, not just considering observed cases. It thus also takes into account the 'correct negatives' or a correctly forecasted *absence* of a class. Accuracy ranges from 0 to 1, with 0 being the lowest score and 1 being the highest. While it is a useful metric for general performance, it is more quickly influenced by imbalanced datasets, since it is easier to correctly forecast the absence of a class if it is rarely observed. Accuracy is given by:

$$Accuracy = \frac{H + CN}{N} \tag{4.3}$$

3. Critical Success Index

The Critical Success Index, or CSI, is similar to POD in the sense that is also accounts for correct detection of observed events. However, it also takes into account both misses and false alarms. This metric penalizes the model for forecasting a class when it did not occur, e.g. a false alarm. This index therefore provides a more complete overview of the model's performance. It ranges from 0 to 1, 0 being the worst performance and 1 being the best. The CSI is given by:

$$CSI = \frac{H}{H + M + F} \tag{4.4}$$

4. False Alarm Ratio

Finally, the False Alarm Ratio solely measures the amount of false alarms relative to the total amount of forecasts. It ranges from 0 to 1, 1 being the worst performance and 0 the best. The FAR is given by:

$$FAR = \frac{F}{F + M} \tag{4.5}$$

4.3.2 Metrics for the Temporal Fusion Transformer

As the Temporal Fusion Transformer outputs probabilities for each class, we cannot make use of the contingency table as easily. A conversion is necessary to compute the metrics that we used for the Random Forest Classifier.

Since the output consists of probabilities for each class, we can label the class with the highest forecasted probability as 'forecasted'. This way, we flatten the output into a deterministic forecast and we can still make use of the contingency table to calculate POD, CSI and Accuracy.

In the case of using 0.5 as a threshold, a probability of 51% is treated equally as a probability of 99%. However, this approach does not fully evaluate the quality of the probabilities that the model produces. Therefore, we introduce some additional metrics to evaluate this model.

1. Brier Score

The Brier Score is a metric specifically designed for probabilistic forecasting. The score penalizes the model based on how far the forecasted probabilities are away from the observations. Similar to the contingency table, an observed event is denoted by '1' and the absence of the class by '0'. The forecasted probability lies somewhere between 0 and 1. By calculating the Brier Score, we penalize the model both for overconfident, incorrect predictions (observed = 0, forecasted > 0) as well as underconfident correct predictions (observed = 1, forecasted < 1). The Brier Score is given by Equation 4.6, where f is the forecast probability ranging between 0 and 1, and o is either 1 or 0 depending on whether the class was observed or not. A perfect score for the Brier Score is 0.

$$BS = \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2 \tag{4.6}$$

2. Brier Skill Score

An additional metric building on the Brier Score is the Brier Skill Score. This score is one of the most common evaluation metrics for forecasting (Wilks, 2010). It compares the Brier Score of the model to the Brier Score of some type of reference forecast. It provides context on how well the model performs compared to a simple forecast. The Brier Skill Score is given by:

$$BSS = 1 - \frac{BS_{model}}{BS_{reference}} \tag{4.7}$$

The reference forecast can be any type of forecast. A common approach that we implemented is to use the climatology of the dataset, where the probability of a class is simply the percentage of its occurrence. Additionally, we also used a persistence forecast, which uses the occurrence of classes at one timestep as the forecast for the following timestep. A BSS above 0 indicates the model performs better than the reference, while a BSS below 0 indicates worse performance.

5 Results

This chapter discusses the performance of the RFC and the TFT in forecasting BZO phases based on horizontal visibility (subchapter 5.1), RVR (subchapter 5.2) and vertical visibility (subchapter 5.3). We will present the evaluation metrics described in subchapter 4.3 to assess the models' performances. We present the evaluation metrics of all experiments in one table. Plots per metric over all lead times can be found in Appendix H. We also compare our findings with those from earlier research.

5.1 Horizontal Visibility

In this subchapter we describe the performance of the RFC and the TFT that forecasted BZO phases based on horizontal visibility. We assess the RFC experiments 'RFC-Direct', 'RFC-Recursive' and 'RFC-HARMONIE', followed by the TFT experiments 'TFT-Focal-Loss', 'TFT-Weighted' and 'TFT-HARMONIE'.

5.1.1 Random Forest Classifiers

We start by discussing the performance of RFCs in forecasting BZO phases based on horizontal visibility. We assess the performance using the POD, Accuracy, CSI, FAR (Table 5.1) and the confidence interval of POD (Figure 5.1 and Figure 5.2).

Table 5.1: The Probability of Detection (POD), Accuracy, Critical Success Index (CSI) and False Alarm Ratio (FAR) for the Random Forest Classifier forecasting BZO phases based on horizontal visibility, for all three RFC-experiments. First value is the score at the first lead time (10 minutes for experiment 1 and 2, 1 hour for experiment 3). Second value in brackets is the average score over all lead times up to 4 hours. The best value for the specific category, over all experiments and both locations, is given in green/bold. The worst value for the specific category, over all experiments and both locations, is given in orange/italic.

	Metric	**POD**		**Accuracy**		**CSI**		**FAR**	
Experiment	**Category /Location**	**West**	**Center**	**West**	**Center**	**West**	**Center**	**West**	**Center**
Experiment 1: RFC-Direct	Good	0.921 (0.810)	0.938 (0.832)	0.971 (*0.815*)	0.976 (0.825)	0.855 (0.611)	0.876 (0.620)	0.077 (0.288)	0.070 (*0.294*)
	Marginal	0.780 (0.596)	0.865 (0.608)	0.869 (0.627)	0.884 (*0.671*)	0.532 (0.305)	0.570 (0.329)	0.374 (*0.613*)	0.373 (0.580)
	A	0.616 (*0.189*)	0.505 (0.228)	0.760 (0.810)	0.788 (0.788)	0.330 (*0.109*)	0.300 (0.131)	0.583 (0.776)	0.577 (0.750)
	B	0.108 (*0.018*)	0.163 (0.038)	0.881 (**0.936**)	0.865 (0.918)	0.090 (*0.015*)	0.135 (0.030)	0.642 (0.847)	0.529 (0.831)
	C	0.377 (*0.112*)	0.502 (0.169)	0.848 (**0.892**)	0.811 (0.857)	0.276 (*0.083*)	0.304 (0.101)	0.478 (0.722)	0.560 (0.715)
	D	0.653 (0.188)	0.563 (*0.140*)	0.908 (0.879)	0.899 (0.884)	0.544 (0.149)	0.481 (*0.115*)	0.232 (*0.635*)	0.235 (0.574)
Experiment 2: RFC-Recursive	Good	0.972 (*0.564*)	**0.981** (0.605)	0.986 (0.846)	**0.989** (0.871)	0.919 (*0.551*)	**0.935** (0.594)	0.056 (0.039)	0.047 (**0.032**)
	Marginal	0.906 (*0.364*)	**0.944** (0.401)	**0.926** (0.750)	0.915 (0.765)	**0.674** (*0.285*)	0.654 (0.315)	**0.275** (0.460)	0.319 (0.432)
	A	**0.659** (0.305)	0.586 (0.299)	0.821 (0.777)	**0.837** (0.772)	**0.380** (0.168)	0.375 (0.176)	0.524 (0.737)	**0.490** (0.721)
	B	0.260 (0.280)	**0.323** (0.268)	0.835 (0.821)	0.859 (0.822)	0.208 (0.093)	**0.277** (0.110)	0.488 (*0.871*)	**0.336** (0.841)
	C	0.504 (0.392)	**0.612** (0.340)	0.855 (0.817)	0.860 (0.802)	0.365 (0.181)	**0.423** (0.169)	0.424 (0.752)	**0.421** (*0.756*)
	D	**0.737** (0.559)	0.733 (0.549)	**0.923** (0.849)	0.932 (0.849)	0.615 (0.325)	**0.643** (0.322)	0.207 (0.575)	**0.161** (0.592)
Experiment 3: RFC-HARMONIE	Good	0.927 (0.921)		0.947 (0.948)		0.753 (0.753)		0.196 (0.191)	
	Marginal	0.709 (0.711)		0.752 (0.750)		0.328 (0.327)		0.618 (0.620)	
	A	0.226 (0.216)		*0.739* (0.745)		0.125 (0.123)		*0.781* (0.779)	
	B	0.059 (0.072)		*0.807* (0.810)		0.046 (0.057)		0.839 (0.791)	
	C	0.275 (0.290)		0.802 (*0.801*)		0.188 (0.195)		0.626 (0.623)	
	D	0.382 (0.373)		0.813 (*0.808*)		0.250 (0.242)		0.571 (0.586)	

Performance per Class

The best-performing class over all experiments is 'Good', (initial POD of 0.921 or higher, initial Accuracy of 0.947 or higher, initial CSI of 0.752 or higher), closely followed by 'Marginal' (initial POD of 0.709 or higher, initial Accuracy 0.752 or higher, initial CSI of 0.328 or higher). These POD scores were similar to scores found by Bartok et al. (2022), who achieved an overall POD of 0.84 for a model nowcasting visibility conditions. Even though the dataset was balanced, our RFC models still seem biased towards the majority classes, because scores of other classes are substantially lower. After 'Good' and 'Marginal', the order classes in terms of initial POD is 'D' (0.382 – 0.737), 'A' (0.226 – 0.659), 'C' (0.275 – 0.612) and finally 'B' (0.059 – 0.323). These scores are somewhat lower than POD scores found in other research on independent decision tree forecasting (e.g. POD: 0.57 – 0.77 (Dewi et al., 2020); 0.885 (Liu et al., 2024); 0.733 – 0.848 (Ohashi & Hara, 2024). This bias towards majority classes is likely caused by the broader range of values and conditions in majority classes, making forecasts in these classes easier for the model.

Similar patterns are observed in the other metrics, showing substantially lower performance for class 'B' than other classes. For example, the FAR for class 'B' is especially high, ranging from 0.336 to 0.871 over all experiments. This order of performance does not follow the natural order of the classes, indicating the model finds it more difficult to forecast intermediate classes (like 'B') than extreme cases, like those in class 'D'. This suggests there might be a stronger correlation between atmospheric conditions and visibility during those extreme cases. A feature correlation analysis could confirm or refute this hypothesis. Castillo-Botón et al. (2022) also trained a RFC to forecast 5 visibility classes and obtained POD scores of 0.87, 0.78, 0.41, 0.31 and 0.93, respectively. This indicates that their model also struggled to forecast intermediate classes but found extreme low-visibility cases relatively easy to detect. The patterns in our results were similar to this finding.

We can conclude that RFC models struggle to forecast rare, intermediate classes in horizontal visibility, underperforming slightly to earlier literature, and are more successful in forecasting extreme low-visibility classes and majority classes. The latter is likely partially caused by a residual bias towards majority classes that remains even after balancing.

Performance per Experiment

We found that the initial POD, CSI and Accuracy scores are generally higher than the average scores over all lead times. This is expected behavior for a forecast, as uncertainty increases at longer lead times. However, there are notable differences between the experiments. Specifically, RFC-Direct has much lower average POD scores compared to the initial POD than the two other experiments. For example, the average POD for class A, West was 0.189 in RFC-Direct and 0.305 in RFC-Recursive, even though both experiments had similar initial scores (0.616 and 0.659 respectively). This was a surprising result, as we expected more propagating errors in the recursive model, but rather this model showed better performance at longer lead times. A notable observation was that the POD and CSI showed a spike in the second timestep for the recursive models, which we could not explain (Appendix H, Figures H.7 and H.9).

Additionally, we found higher initial results for other metrics in the recursive model (e.g. CSI, Marginal, West: 0.674, versus RFC-Direct: 0.532; CSI, class B, Center: 0.277 versus RFC-Direct: 0.135). These metrics showed a similar performance decrease over longer lead times in both experiments (e.g. CSI Good, Center: 0.876 to 0.620 in RFC-Direct, 0.935 to 0.594 in RFC-Recursive). This indicates that only the detection rate decreases more in RFC-Direct over time, while the number of correct negatives and false alarms decreases similarly in both experiments. To summarize, RFC-Recursive performs best in predicting visibility initially and also shows higher detection rates than RFC-Direct over longer lead times. Initial scores for RFC-Direct are slightly lower, and detection rates decrease faster over time than for RFC-Recursive (see also plots over lead times; Appendix H: Figures H.1 – H.12,).

For RFC-HARMONIE, scores decrease the least and sometimes even slightly increase over longer lead times (e.g. POD Class C: 0.275 to 0.290; CSI Class B: 0.046 to 0.057; see Appendix H: Figures H.13 – H.15). This aligns with the stable performance of the HARMONIE model over 4 hours (Appendix B). Still, RFC-HARMONIE's performance is higher than the HARMONIE model (see Appendix B), indicating the post-processing did improve HARMONIE output. However, generally, RFC-HARMONIE performs worse than the other two RFC experiments and also underperforms compared to earlier research in the low-visibility classes (POD low-visibility 0.057 – 0.382, versus POD 0.13-0.72 (Yu et al., 2021); POD 0.338 (Thomasson, 2023)). None of the category's best values are found in this experiment (no bold green values), and there are 5 out of 24 worst scores. Most striking is the model's performance in RFC-HARMONIE for class 'B'. Its POD (0.059) decreased to less than half of the first two experiments (RFC-Direct: 0.163, RFC-Recursive: 0.323), similar to the CSI (0.046; RFC-Direct 0.135, RFC-Recursive: 0.277).

Relatively, the performance of low-visibility classes decreased much more than class 'Good', which scores were barely affected (e.g. POD 0.938 to 0.921, Accuracy 0.976 to 0.948). Thomasson (2023), who also post-processed NWM data using ML, also recognized this behavior. His model showed a lower POD for fog-like classes only (0.227) than the general model performance (0.338). There are very few exceptions in our model, where RFC-HARMONIE showed higher results (e.g. Accuracy Marginal: 0.750 versus RFC-Direct: 0.671; FAR class D: 0.586 versus RFC-Recursive: 0.592; FAR class C: 0.623 versus RFC-Recursive: 0.756). However, these differences are very minimal and the general model performance of RFC-HARMONIE is lower than the other experiments. Therefore, we can conclude that including HARMONIE data into the RFC model did not improve its performance, and that independent versions of the RFC perform just as well, or better.

Performance per Location

POD and CSI scores of BZO phases 'Good', 'Marginal', 'B' and 'C', are generally higher in BZO Center in both RFC-Direct and RFC-Recursive. Oppositely, class 'A' and 'D' score higher in BZO West (e.g. RFC-Direct, initial: West vs Center: 0.616 versus 0.505, 0.653 versus 0.563 respectively). This is likely because there were more training samples of these classes in BZO West. In terms of Accuracy and FAR, these patterns are slightly less distinct. For example, the Accuracy for class 'B' and 'C' is actually higher in BZO West in RFC-Direct (Class B: 0.881 (West) and 0.865 (Center); Class C: 0.848 (West) and 0.811 (Center)). Similarly, in terms of FAR, some cases also show better performance of class 'B' and 'C' in BZO West (e.g. RFC-Direct, class C: 0.478 (West) and 0.560 (Center)).

Despite this, metrics generally portray a similar pattern in location-wise performance. BZO West has higher performance in classes 'A' and 'D', while BZO Center performs better in the other classes. It could mean that these classes are more distinctly defined, are less affected by noise and exhibit fewer short-term changes over time at BZO Center, making it easier for the model to predict these classes over time. However, we did not find evidence of this in the class distributions (Appendix C). We hypothesize that the intermediate classes 'B' and 'C' are coupled to more distinct conditions in BZO Center, because it experiences less fluctuations in meteorological variables as it is in a more urbanized area than BZO West. Oppositely, we hypothesize that BZO West performs better in class 'D', which is often related to thick fog. Thick fog is more likely to occur in BZO West because of its location (see Background), which is why we suspect the model finds it easier to detect this extreme class here.

A key takeaway is that the RFC models show distinct differences in performance for the two locations 'BZO West' and 'BZO Center', whether that may be due to their geographical properties or due to differences in the relative occurrence of BZO phases. This indicates it is useful to make separate forecasts for the two locations, and that a single forecast for the airfield would possibly miss out on nuances in fog occurrence between the locations.

Detection Certainty

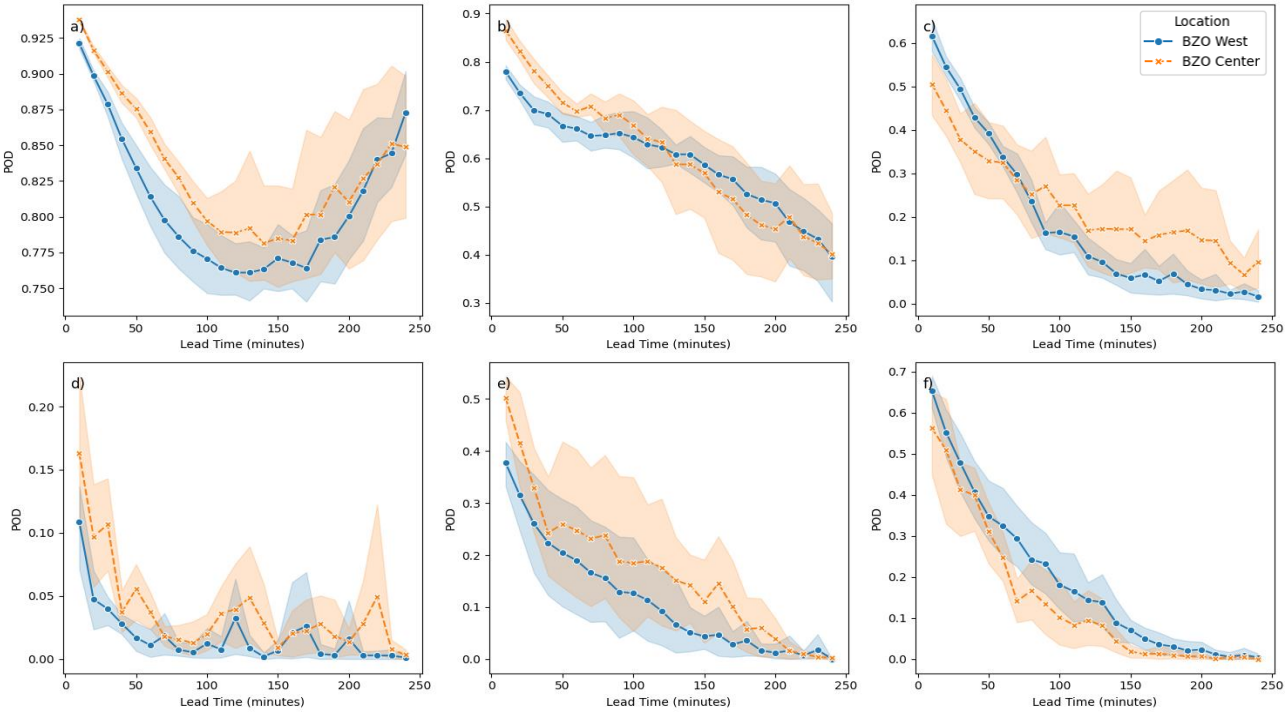

Figure 5.1: The POD (Y-axis) for phase 'Good' (a), 'Marginal ' (b), 'A' (c), 'B' (d), 'C' (e), and 'D' (f) for the direct Random Forest Classifier (RFC-Direct) forecasting BZO phases based on horizontal visibility over the lead time of 4 hours (X-axis). Both locations BZO West (blue) and BZO Center (orange) are shown. The shaded areas show the 95% confidence interval.

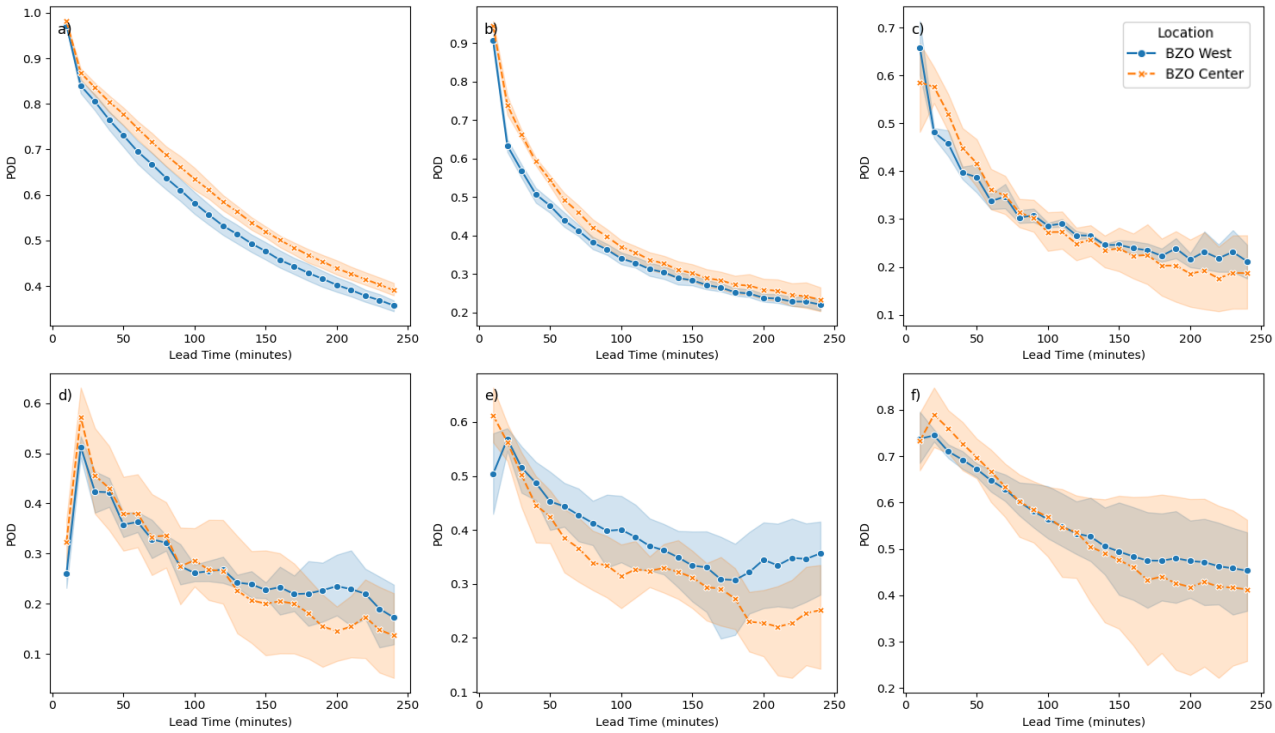

Figure 5.2: The POD (Y-axis) for phase 'Good' (a), 'Marginal ' (b), 'A' (c), 'B' (d), 'C' (e), and 'D' (f) for the recursive Random Forest Classifier (RFC-Recursive) forecasting BZO phases based on horizontal visibility over the lead time of 4 hours (X-axis). Both locations BZO West (blue) and BZO Center (orange) are shown. The shaded areas show the 95% confidence interval.

The confidence interval of the POD (Figure 5.1, 5.2) is generally narrowest for class 'Good' with its widest POD range slightly over 0.1. Class 'Marginal' and 'D' show slightly wider intervals, both ranging somewhere between 0.1 and 0.2. This supports our earlier observation that the RFC models find extreme cases easier to predict than other low-visibility classes. Classes 'A' and 'C' show the widest interval of about 0.2 and 0.3 respectively, reflecting less certainty from the model. Finally, class 'B' exhibits a spiky and inconsistent pattern. It indicates the model is struggling to find a stable pattern for this class. The recursive model also shows a distinct oscillating behavior in class 'A' and 'B', which indicates that the model struggles to distinguish between these classes, likely finding it difficult to classify edge cases.

The confidence intervals are narrower at shorter lead times for RFC-Recursive, but widen more over the longer lead times than RFC-Direct. This observation aligns with the generally lower performance of RFC-Recursive over longer lead times in terms of POD, which we found above, and also aligns with patterns that one would expect from a recursive model, as uncertainty increases in every future timestep. Moreover, a notable difference is the confidence interval in class 'D', which is much narrower in RFC-Direct than in RFC-Recursive, especially for BZO Center. Despite the higher POD scores in RFC-Recursive the model still appears to have large uncertainty modelling class D, which contradicts our hypothesis that the RFC models find extreme cases easier to predict.

Generally, the confidence intervals are narrower in location BZO West. This is most apparent in classes 'Marginal', 'A' in RFC-Direct and 'A', 'B', and 'D' for RFC-Recursive. For low-visibility classes, it could indicate that the slightly higher abundance of data in BZO West improved the model's confidence. Another possible location is that BZO Center experiences a slightly more diverse range of conditions, specifically in classes like 'Marginal' and 'A'. If conditions for low-visibility are more sharply defined in BZO West, it raises the model's confidence in forecasts in this area. We already hypothesized this to be the case earlier, and a feature correlation analysis would give more insight in this.

The main takeaway from this is that even though recursive models generally display higher detection rates, their uncertainty does increase more over longer lead times. Oscillating behavior in intermediate classes also supports our earlier finding that the models struggle in forecasting intermediate classes. Moreover, uncertainty is generally lower in BZO West, which suggests that the mechanisms for fog formation are more clear to the model in this location.

5.1.2 Temporal Fusion Transformers

In the following section we will discuss the performance of the TFT fitted with a Focal Loss function in forecasting BZO phases based on horizontal visibility for the experiments 'TFT-Focal-Loss', 'TFT-Weighted' and 'TFT-HARMONIE'. We will assess the performance of the models as a deterministic model using the POD and CSI, and as a probabilistic model using Brier Score, BSS Climatology and BSS Persistence.

Performance per Class

When assessing the TFT models as a deterministic model a notable finding is the low performance of low-visibility classes. We leave out scores of 0.0, which occur when a class never received the highest probability, and therefore was never forecasted in the deterministic forecast. Still, we find very low scores for these classes among all experiments (e.g. POD class B, TFT-Weighted: 0.013; POD class C, TFT-Focal-Loss: 0.006). Oppositely, class 'Good' receives high scores, similar to scores we found in the RFC experiments (e.g. POD 0.934 to 0.993, CSI 0.922 to 0.977). For class 'Marginal', scores of TFT-Focal-Loss are similar to the RFC experiments (POD around 0.7 and CSI around 0.6), but TFT-Weighted and TFT-HARMONIE show lower values (POD 0.381 for TFT-Weighted, CSI 0.355 and 0.257 for TFT-Weighted and TFT-HARMONIE respectively). The high POD and CSI scores for classes 'Good' and 'Marginal' indicate the model is likely biased towards these classes since the dataset is not balanced.

Table 5.2: The Probability of Detection (POD), Critical Success Index (CSI) and Brier Score, Climatological Brier Skill Score (BSS Climatology) and Persistence Brier Skill Score (BSS Persistence) for the Temporal Fusion Transformer forecasting BZO phases based on horizontal visibility, for all three TFT-experiments. First value is the score at the first lead time (10 minutes for experiment 1 and 2, 1 hour for experiment 3). Second value in brackets is the average score over all lead times up to 4 hours. The best value for the specific category, over all experiments, is given in green/bold. The worst value for the specific category, over all experiments, is given in orange/italic. Values of 0.0 excluded.

Experiment	Category	POD	CSI	Brier Score	BSS Climatology	BSS Persistence
Experiment 1: TFT-Focal-Loss	Good	**0.993** (0.981)	**0.977** (0.945)	**0.030** (0.054)	**0.661** (0.395)	0.0 (-3.295)
	Marginal	**0.769** (0.536)	**0.626** (0.397)	**0.040** (0.058)	**0.410** (0.158)	0.0 (-2.544)
	A	0.145 (0.084)	**0.115** (0.063)	0.009 (0.010)	**0.188** (0.136)	0.0 (-0.773)
	B	0.0 (0.0)	0.0 (0.0)	0.003 (0.003)	**0.091** (0.052)	0.0 (0.066)
	C	**0.006** (0.0)	**0.006** (0.0)	0.004 (0.004)	**0.151** (0.125)	0.0 (-0.510)
	D	**0.639** (0.543)	**0.237** (0.187)	0.004 (0.004)	**0.181** (0.106)	0.0 (-1.971)
Experiment 2: TFT-Weighted	Good	0.934 (*0.926*)	0.922 (*0.903*)	0.092 (*0.110*)	-0.031 (*-0.239*)	0.0 (*-7.645*)
	Marginal	0.674 (0.562)	0.355 (0.281)	0.068 (*0.077*)	0.011 (-0.122)	0.0 (*-3.673*)
	A	**0.275** (0.261)	0.111 (0.101)	0.013 (*0.014*)	-0.112 (*-0.157*)	0.0 (*-1.360*)
	B	0.013 (**0.014**)	0.011 (0.012)	*0.004* (*0.004*)	*-0.274* (-0.271)	0.0 (*-0.236*)
	C	0.0 (0.0)	0.0 (0.0)	*0.005* (*0.005*)	0.013 (0.016)	0.0 (*-0.689*)
	D	0.233 (*0.122*)	0.100 (0.072)	0.005 (0.005)	-0.094 (-0.062)	0.0 (-2.506)
Experiment 3: TFT-HARMONIE	Good	0.972 (0.985)	0.931 (0.922)	0.064 (0.081)	0.183 (-0.031)	0.0 (-0.636)
	Marginal	0.381 (*0.161*)	0.257 (*0.123*)	0.061 (0.072)	0.041 (*-0.131*)	0.0 (-0.295)
	A	0.017 (*0.011*)	0.016 (*0.010*)	**0.007** (0.007)	-0.042 (-0.074)	0.0 (**0.282**)
	B	0.0 (0.0)	0.0 (0.0)	**0.002** (**0.002**)	0.015 (0.011)	0.0 (**0.357**)
	C	0.0 (0.0)	0.0 (0.0)	**0.001** (**0.001**)	0.0 (*-0.005*)	0.0 (**0.331**)
	D	0.203 (0.130)	0.082 (*0.058*)	*0.008* (*0.008*)	-0.038 (*-0.096*)	0.0 (*-0.388*)

However, we observe a strikingly different pattern when we assess the model's performances as a probabilistic model, using Brier Scores. Regarding the Brier Score, the best-performing classes are classes 'B', 'C' and 'D' with scores between 0.001 and 0.005. For context, the only TFT that has been applied to visibility forecasting, as a regressor, obtained a Brier Score of 0.01 for visibility below 1000 meters (Wehrli et al., 2024). Our model outperforms this TFT in the low-visibility classes. Oppositely, the worst performing classes in our TFT are classes 'Good' and 'Marginal' with scores ranging from 0.03 to 0.110 and 0.04 to 0.077, respectively. Even though these scores are above 0.01, they are still well below Brier Scores obtained by other visibility classifiers (e.g. 0.125 – 0.83 (Ohashi & Hara, 2024)). Therefore, we can say that our TFT models are very capable of forecasting visibility probabilities, specifically for the low-visibility classes.

We can conclude that TFTs with a Focal Loss function are highly biased towards majority classes 'Good' and 'Marginal' in the deterministic form. However, as a probabilistic model, the models are very capable of forecasting visibility categories, specifically the low-visibility classes.

Performance per Experiment

Even though Brier Scores indicate very high performance of the TFT models, the models do not always show skill when compared to a climatological or persistence forecast. In this case, we find large differences in the different experiments. The BSS climatology is positive in all classes for TFT-Focal-Loss, indicating better performance than a climatological forecast. However, the BSS climatology is only positive for a few classes in TFT-Weighted (Marginal, initial: 0.011; C, initial: 0.013) and TFT-HARMONIE (Good, initial: 0.0183; Marginal, initial: 0.041; B, initial: 0.015; B, average: 0.011). These BSS values are generally lower than the BSS values for the HARMONIE-QRF forecast, especially at short lead times (see Background Information – Low Visibility Procedures at Amsterdam Airport Schiphol; class Marginal 0.306 – 0.689, class A 0.188 – 0.661, class B 0.157 – 0.643, class C 0.007 – 0.403). The only class that showed better skill than the HARMONIE-QRF model was class 'D', whose scores ranged from -0.114 to 0.074 for the HARMONIE-QRF model but were higher in our TFT (0.106 to 0.181). Additionally, the BSS persistence is generally negative in all our experiments, indicating the TFT models perform worse than a persistence forecast. However, in TFT-HARMONIE, there are three classes with a positive BSS Persistence: 'A' (average, 0.282), 'B' (average, 0.357) and 'C' (average, 0.331).

Other scores also show apparent differences between the experiments. TFT-Weighted generally shows the lowest performance across all metrics and classes. This suggests that adding weights to the categories in the TFT disturbed the ability of the model to detect natural patterns in class occurrence. The weights may have increased the model's confidence in low-visibility scores too much, resulting in the lowest Brier Scores for class 'A' (0.013/0.014), 'B' (0.004) and 'C' (0.005) in this experiment. TFT-Focal-Loss shows the highest performance as a deterministic forecast (highest POD/CSI values) and also shows the best skill compared to a climatological forecast. However, TFT-HARMONIE seems to have improved Brier Scores of intermediate classes 'A', 'B' and 'C', as well as improved the model's skill compared to the persistence forecast in these classes. Introducing HARMONIE data seems to have improved the model's ability to distinguish between intermediate classes.

To summarize, TFT models with a Focal Loss function show high performance as a probabilistic model but low performance as a deterministic model. As a deterministic model, the models are highly biased towards majority classes. Adding weights to the classes did not improve performance. Adding HARMONIE data slightly improved the performance of intermediate classes 'B', 'C', and 'D'.

5.2 Runway Visual Range

The following subchapter describes the performance of the RFC that forecasted BZO phases based on RVR. We only discuss RFC since there was insufficient data to train the TFT on RVR. Introducing RVR as a target resulted in fewer data points for low-visibility classes, so class 'C' and 'D' were combined. RVR data was only available below 1500 meters, so we will focus our analysis on the low-visibility classes.

Table 5.3: The Probability of Detection (POD), Accuracy, Critical Success Index (CSI) and False Alarm Ratio (FAR) for the Random Forest Classifier forecasting BZO phases based on RVR, for the first two RFC-experiments. First value is the score at the first lead time (10 minutes). Second value in brackets is the average score over all lead times up to 4 hours. The best value for the specific category, over all experiments and both locations, is given in green/bold. The worst value for the specific category, over all experiments and both locations, is given in orange/italic.

	Metric	POD		Accuracy		CSI		FAR	
Experiment	**Category /Location**	**West**	**Center**	**West**	**Center**	**West**	**Center**	**West**	**Center**
Experiment 1: RFC-Direct	Good	0.920 (0.829)	0.917 (0.847)	0.965 (0.813)	0.961 (*0.803*)	0.854 (0.640)	0.841 (0.637)	0.077 (0.262)	0.091 (*0.278*)
	Marginal	0.794 (0.532)	0.809 (0.488)	0.900 (*0.706*)	0.908 (0.766)	0.627 (*0.322*)	0.655 (0.325)	0.250 (*0.559*)	0.226 (0.516)
	A	0.751 (0.416)	0.796 (0.518)	0.771 (0.760)	0.801 (*0.740*)	0.433 (0.239)	**0.487** (0.297)	0.495 (0.620)	**0.445** (0.576)
	B	0.300 (*0.066*)	0.397 (0.091)	0.804 (**0.888**)	0.780 (0.880)	0.195 (*0.045*)	0.255 (0.070)	0.637 (*0.839*)	0.592 (0.728)
	C	0.330 (0.054)	0.135 (*0.012*)	0.866 (0.894)	0.865 (**0.922**)	0.300 (0.049)	0.125 (*0.011*)	0.241 (*0.773*)	0.381 (0.504)
Experiment 2: RFC-Recursive	Good	0.973 (*0.574*)	**0.979** (0.558)	**0.982** (0.826)	0.981 (0.815)	**0.916** (0.561)	0.913 (*0.546*)	0.060 (0.039)	0.069 (**0.035**)
	Marginal	0.896 (*0.460*)	**0.917** (0.517)	0.934 (0.776)	**0.941** (0.803)	0.733 (0.345)	**0.756** (0.380)	0.197 (0.448)	**0.187** (0.434)
	A	0.734 (*0.382*)	**0.811** (0.386)	**0.814** (0.751)	0.782 (0.742)	0.441 (*0.230*)	0.427 (0.249)	0.474 (*0.637*)	0.526 (0.601)
	B	**0.422** (0.392)	0.346 (0.396)	0.801 (0.786)	*0.750* (0.788)	**0.299** (0.151)	0.223 (0.167)	**0.494** (0.799)	0.616 (0.767)
	C	0.467 (**0.540**)	0.149 (0.495)	0.866 (*0.820*)	0.827 (0.824)	**0.410** (0.257)	0.146 (0.211)	**0.239** (0.674)	0.290 (0.742)

Performance per Class

First, we can compare the scores of these experiments to those we performed on horizontal visibility earlier. Scores (POD, CSI and Accuracy) for class 'Good' and 'Marginal' are similar (e.g. POD West: initial 0.920 to 0.921, average 0.829 to 0.810 (RFC-Direct); initial 0.973 to 0.972, average 0.574 to 0.564 (RFC-Recursive)). For 'A', performance greatly improved (e.g. POD Center: initial 0.505 to 0.796, average 0.228 to 0.518 (RFC-Direct); initial 0.586 to 0.811, average 0.299 to 0.386 (RFC-Recursive)). For class 'B' scores also improved (e.g. POD West: initial 0.108 to 0.300, average 0.018 to 0.066 (RFC-Direct); initial 0.260 to 0.422, average 0.280 to 0.392 (RFC-Recursive). However, class 'A' and 'B' did show an increase in FAR when RVR was introduced (e.g. FAR Center, RFC-Recursive: 0.490 to 0.526 (class 'A') 0.336 to 0.616 (class 'B')).

Introducing RVR as a target resulted in a different performance order than for horizontal visibility experiments. In terms of POD, the best-performing classes are still 'Good' (0.547 – 0.979) and 'Marginal' (0.460 – 0.917). However, in the case of RVR, these classes are followed by 'A' (0.382 – 0.734), 'B' (0.066 – 0.422) and 'C' (0.012 – 0.540), resembling the natural order of the classes. We no longer observe a substantially higher detection rate of extreme low-visibility cases, which are now part of class 'C'. We hypothesize that since RVR data included much fewer cases of low-visibility cases, the model did not have sufficient samples to accurately recognize patterns for these cases, resulting in lower detection rates for class 'C' (0.012 – 0.540) than class 'D' in the horizontal visibility experiments (0.140 – 0.737).

In terms of the other metrics, e.g. for FAR, the order of performance slightly changes to 'Good' (0.025 – 0.278), 'Marginal' (0.187 – 0.559), 'C' (0.239 – 0.773), 'A' (0.445 – 0.637) and 'B' (0.494 – 0.839). Interestingly, this pattern is similar to the pattern we observed in all metrics for horizontal visibility. Therefore, introducing RVR only resulted in a different class order of detection rate but did not influence correct negatives or false alarms as much.

Performance per Experiment

The recursive model shows the highest performance overall. 16 out of 20 'best values' are found in RFC-Recursive, and with few exceptions, all metrics score better in this experiment. This is similar behavior to what we observed in the horizontal visibility models. Introducing RVR, therefore, did not influence the performance of the direct and recursive RFCs relative to each other.

Performance per Location

In classes 'Good', 'Marginal' and 'A', there are minimal differences in initial scores between the two locations. In the horizontal visibility experiments, differences were more apparent. RVR appears to vary less over the two locations, causing similar performance in the two locations. This consistency between the locations likely results from the adjustable runway light luminance that is included in RVR. For example, if visibility is lower in one location (e.g. BZO West), the runway lights are likely increased in brightness, while in better visibility conditions (e.g. BZO Center) lights are adjusted less. This reduced the relative difference in RVR between the two locations.

This hypothesis holds for the first three classes, but we observe a different pattern for classes 'B' and 'C'. These classes both consistently perform better in area West (e.g. RFC-Recursive, POD, class B: 0.422 (West) versus 0.346 (Center); class C: 0.467 (West) versus 0.149 (Center); FAR class B 0.494 (West) versus 0.616 (Center). This is likely caused by the more persistent fog in BZO West (see Appendix E). If visibility is extremely low in one location (e.g. BZO West), runway lights will reach their maximum brightness. As a result, the difference in RVR between the locations will become more pronounced, and the model performs better in more persistent foggy conditions at BZO West.

Detection Uncertainty

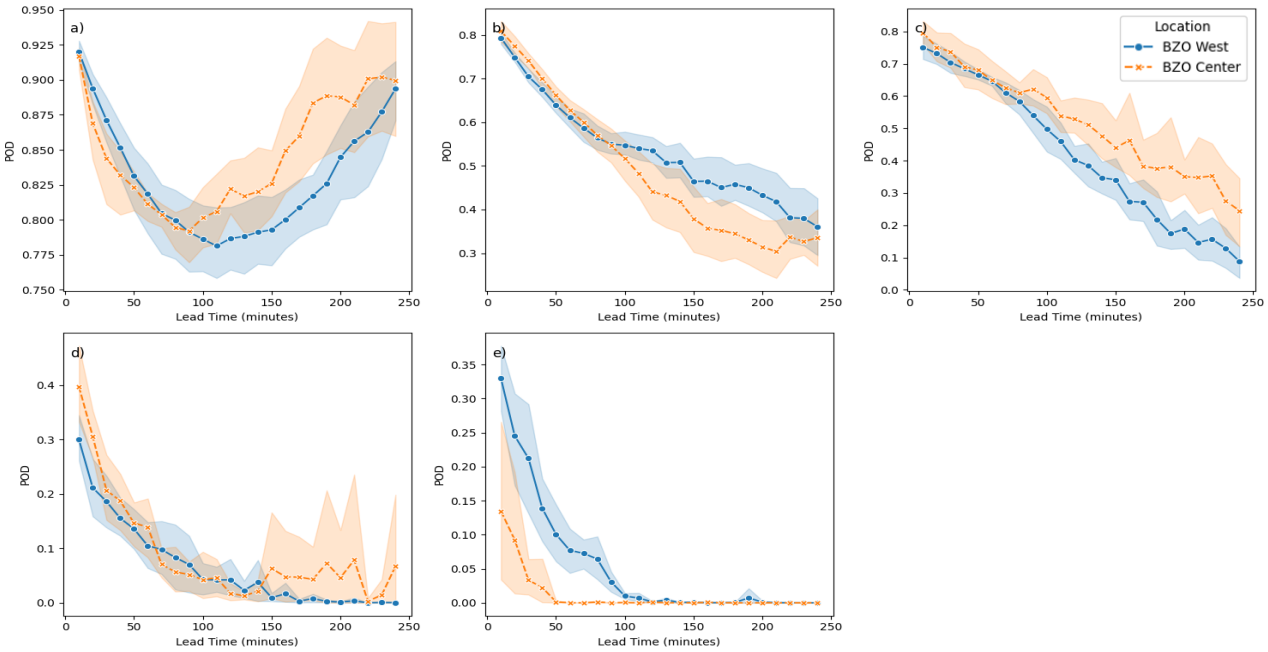

Figure 5.3: The POD for phase 'Good' (a), 'Marginal ' (b), 'A' (c), 'B' (d) and 'C' (e) for the direct Random Forest Classifier (Experiment 1) forecasting BZO phases based on RVR. Both locations BZO West (blue) and BZO Center (orange) are shown. The shaded areas show the 95% confidence interval.

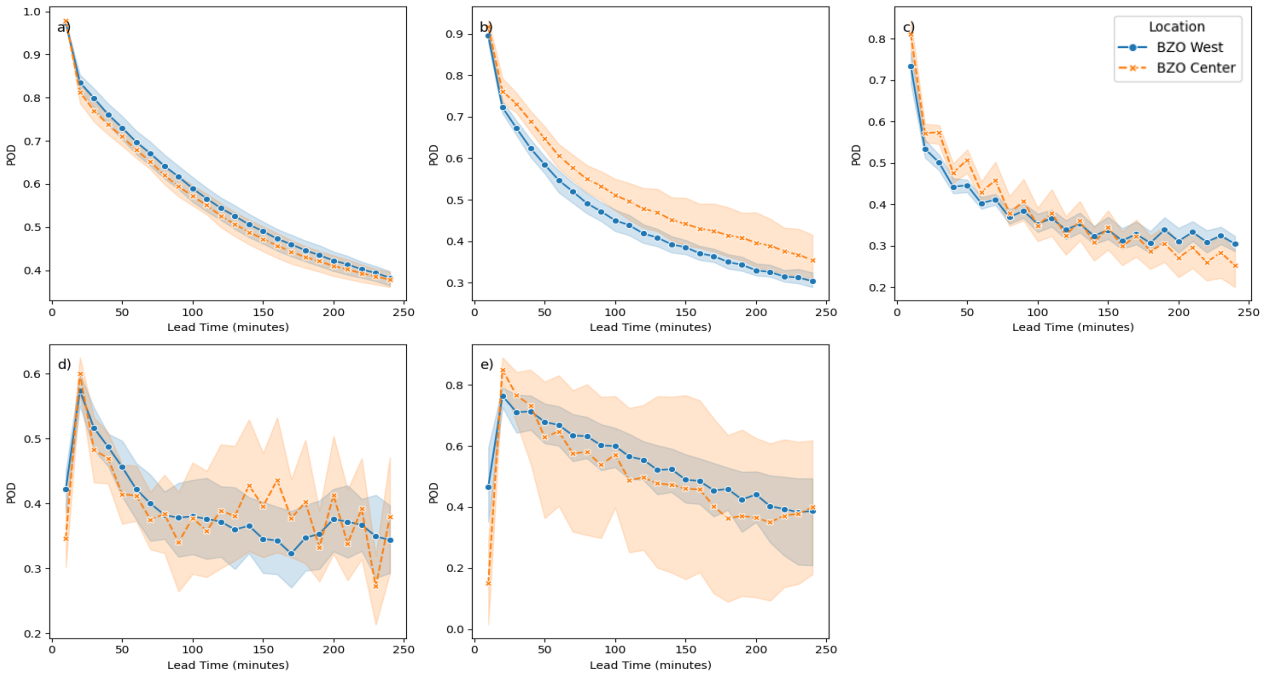

Figure 5.4: The POD for phase 'Good' (a), 'Marginal ' (b), 'A' (c), 'B' (d) and 'C' (e) for the recursive Random Forest Classifier (Experiment 2) forecasting BZO phases based on RVR. Both locations BZO West (blue) and BZO Center (orange) are shown. The shaded areas show the 95% confidence interval.

The confidence intervals of the RVR experiments (Figure 5.3 and 5.4) show similar patterns over the lead time as horizontal visibility experiments. The difference between the locations is less apparent than in horizontal visibility for classes 'Good', 'Marginal' and 'A', which we also observed above. An interesting difference, however, is the confidence in predictions at BZO Center for RFC-Direct. The model shows spikey and inconsistent behavior. Classes 'A' and 'B' also show a more prominent oscillating behavior than in horizontal visibility. We hypothesize that the smoother nature of the RVR dataset (due to the adjustable runway light luminance) made it more difficult for the model to distinguish intermediate classes, especially for edge cases. Additionally, we observe a larger difference in the confidence interval between BZO Center and BZO West for class 'C'. The confidence interval this class is much wider compared to that of BZO West than in the previous experiments. After introducing RVR, there were very few extreme low-visibility samples left, which is why we merged classes 'C' and 'D'. This was mostly the case in BZO Center, since there was already a lower abundance of low-visibility cases in this location. We hypothesize the further decrease in samples caused the higher uncertainty of the models in BZO Center.

To summarize, RVR mainly caused the RFCs to be more confused about edge cases in intermediate classes since the data was more smoothed in these ranges. Additionally, it increased the difference in uncertainty between BZO West and BZO Center by further decreasing the amount of extremely low visibility samples.

5.3 Vertical Visibility

This final subchapter will discuss the results of the RFC and TFT in forecasting BZO phases based on vertical visibility. Vertical visibility is only determining for classes 'Good', 'Marginal', 'A' and 'B', so results are only plotted for these classes (see Background Information – Low Visibility Procedures at Amsterdam Airport Schiphol).

5.3.1 Random Forest Classifiers

The following sections present and discuss the performance of RFC's forecasting BZO phases based on vertical visibility. We discuss the three experiments 'RFC-Direct', 'RFC-Recursive' and 'RFC-HARMONIE' and assess the performance with the POD, Accuracy, CSI, FAR and the confidence interval of POD.

Table 5.4: The Probability of Detection (POD), Accuracy, Critical Success Index (CSI) and False Alarm Ratio (FAR) for the Random Forest Classifier forecasting BZO phases based on vertical visibility, for all three RFC-experiments. First value is the score at the first lead time (10 minutes for experiment 1 and 2, 1 hour for experiment 3). Second value in brackets is the average score over all lead times up to 4 hours. The best value for the specific category, over all experiments and both locations, is given in green/bold. The worst value for the specific category, over all experiments and both locations, is given in orange/italic. Experiment 3 results are given under location 'Center', since the AWS measurement station is located in BZO area Center.

Metric		**POD**		**Accuracy**		**CSI**		**FAR**	
Experiment	**Category /Location**	**West**	**Center**	**West**	**Center**	**West**	**Center**	**West**	**Center**
Experiment 1: RFC-OBS-Direct	Good	0.937 (0.963)	**0.964** (0.962)	0.849 (0.714)	0.908 (0.726)	0.740 (0.683)	**0.822** (0.687)	0.221 (*0.299*)	0.152 (0.293)
	Marginal	0.567 (0.124)	0.698 (0.236)	0.763 (0.826)	0.820 (0.808)	0.386 (0.089)	0.519 (0.173)	0.453 (0.734)	0.323 (0.593)
	A	0.182 (*0.046*)	0.257 (0.056)	0.881 (0.920)	0.858 (0.903)	0.144 (0.034)	0.168 (0.036)	0.530 (0.685)	0.642 (0.688)
	B	0.612 (0.151)	0.616 (*0.118*)	0.905 (0.883)	0.909 (0.888)	0.519 (0.125)	0.530 (*0.100*)	0.221 (0.601)	0.214 (0.491)
Experiment 2: RFC-OBS-Recursive	Good	0.813 (*0.324*)	0.870 (0.347)	0.913 (*0.541*)	**0.941** (0.588)	0.665 (*0.308*)	0.746 (0.339)	0.215 (0.099)	0.160 (0.035)
	Marginal	0.758 (0.081)	0.778 (*0.055*)	0.941 (0.747)	**0.952** (*0.718*)	0.616 (0.053)	**0.654** (*0.041*)	0.233 (0.906)	**0.196** (*0.932*)
	A	**0.775** (0.234)	0.628 (0.054)	0.884 (*0.677*)	**0.927** (0.813)	**0.654** (0.065)	0.503 (*0.031*)	**0.195** (0.921)	0.286 (*0.955*)
	B	0.933 (0.709)	**0.962** (0.946)	0.945 (0.681)	**0.947** (*0.553*)	0.864 (0.235)	**0.910** (0.222)	0.079 (0.741)	**0.056** (*0.775*)
Experiment 3: RFC-HARMONIE	Good		0.876 (0.878)		0.930 (0.927)		0.758 (0.750)		**0.151** (0.162)
	Marginal		**0.813** (0.807)		0.808 (0.807)		0.517 (0.513)		0.412 (0.414)
	A		0.215 (0.214)		0.754 (0.754)		0.180 (0.178)		0.478 (0.480)
	B		0.729 (0.726)		0.824 (0.824)		0.511 (0.510)		0.364 (0.366)

Performance per Class

The highest-performing class was 'Good' (POD: 0.324 – 0.964, Accuracy; 0.541 – 0.941, CSI: 0.308 – 0.822, FAR: 0.151 – 0.299), followed by 'Marginal' (POD: 0.055 – 0.775, Accuracy: 0.718 – 0.952, CSI: 0.041 – 0.654, FAR: 0.196 – 0.932), 'B' (POD: 0.118 – 0.962, Accuracy: 0.553 – 0.947, CSI: 0.100 – 0.910, FAR: 0.056 – 0.775) and finally 'A' (POD: 0.046 – 0.775, Accuracy: 0.677 – 0.927, CSI: 0.031 – 0.654, FAR: 0.195 – 0.955). The performance of the models is generally lower than the performance of RFCs in forecasting horizontal visibility and RVR, and with that automatically also lower than findings from other research. Especially at longer lead times, some scores are exceptionally low (e.g. CSI, RFC-Recursive, class A, Center: 0.031; POD, RFC-Direct, class A, West: 0.046; see also Appendix H: Figures H.47 – H.49).

This order does not match the natural order of the classes, as intermediate class 'A' is the worst-performing class. We also saw this pattern in our horizontal visibility and RVR experiments, where intermediate classes performed worse than the extremely low-visibility classes. These findings indicate that it is not only fog formation that makes intermediate classes challenging to forecast for the RFCs, but rather, this pattern is present for all three targets.

Performance per Experiment

RFC-Recursive clearly outperformed other experiments in forecasting horizontal visibility and this is also the case vertical visibility. 12 out of 16 'best values' are found in the RFC-Recursive experiment and in many cases scores in this experiment are better than in RFC-Direct. An important note, however, is that false alarms in this experiment are very high, especially at longer lead times (Center, Marginal: 0.932; Center A: 0.955). Moreover, the average scores over longer lead times decline quickly in RFC-Recursive and end up being lower than RFC-Direct (e.g. CSI class 'Marginal' in Center; 0.041 (RFC-Recursive) versus 0.173 (RFC-Direct)). This pattern is different than what we observed in the horizontal visibility and RVR experiments, where RFC-Recursive scored well initially as well as over longer lead times. It indicates the recursive RFCs are more capable of forecasting horizontal visibility, and thus fog, than vertical visibility at longer lead times.

However, in the case of vertical visibility, RFC-HARMONIE shows much better results. Even though it only contains 2 out of 16 'best values', its scores are much closer to the RFC-Recursive experiment than in horizontal visibility and RVR. At longer lead times its scores are often better. For example, the average CSI for class 'B' is 0.510, while it is 0.222 in RFC-Recursive and 0.100 in RFC-Direct. This implies that including NWM data does aid the RFC model in forecasting vertical visibility over longer lead times.

Performance per Location

We found that the models performed better in location 'Center' than in 'West' (e.g. POD RFC-Direct: Good 0.937 (West) versus 0.964 (Center), Marginal 0.567 (West) versus 0.698 (Center); FAR RFC-Recursive Good 0.215 (West) versus 0.160 (Center), B 0.079 (West) versus 0.056 (Center)). This is a surprising result. Vertical visibility is mainly influenced by clouds, and clouds are generally less related to small-scale variations in conditions than fog (Bony et al., 2015). Therefore, one would not expect vertical visibility to differ greatly between the two locations. We also did not find differences in the occurrence of BZO phases based on vertical visibility between the locations (see Appendix E: Figure E.2). A likely explanation is that meteorological variables, apart from visibility, were more variable in BZO West because of its rural location. The higher variance in the dataset could cause lower performance of the RFC because it is not able to generalize relationships between the variables and visibility as well. A more extensive dataset analysis could confirm this. The main takeaway from this is that the difference in model performance between the two BZO locations is not limited to the modelling of fog, but also applies to vertical visibility.

Detection Certainty

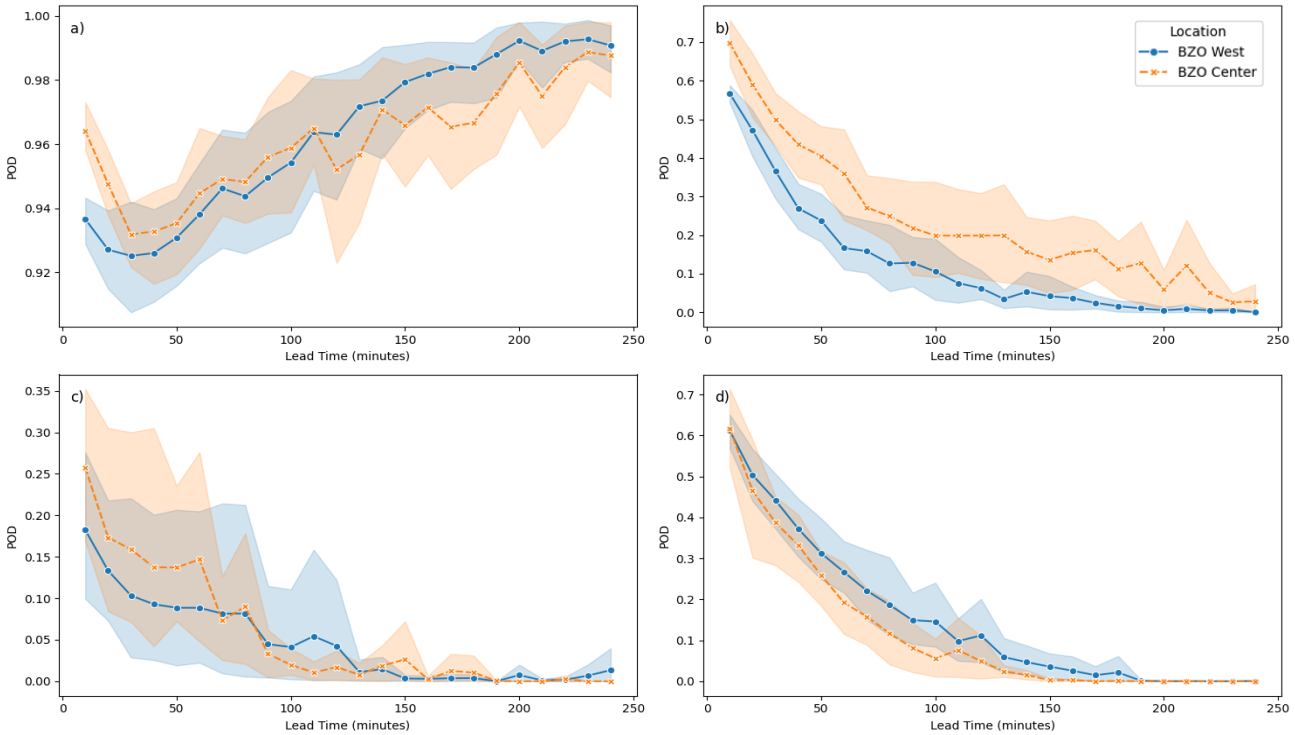

Figure 5.5: The POD for phase 'Good' (a), 'Marginal ' (b), 'A' (c), and 'B' (d) for the direct Random Forest Classifier (Experiment 1) forecasting BZO phases based on vertical visibility. Both locations BZO West (blue) and BZO Center (orange) are shown. The shaded areas show the 95% confidence interval.

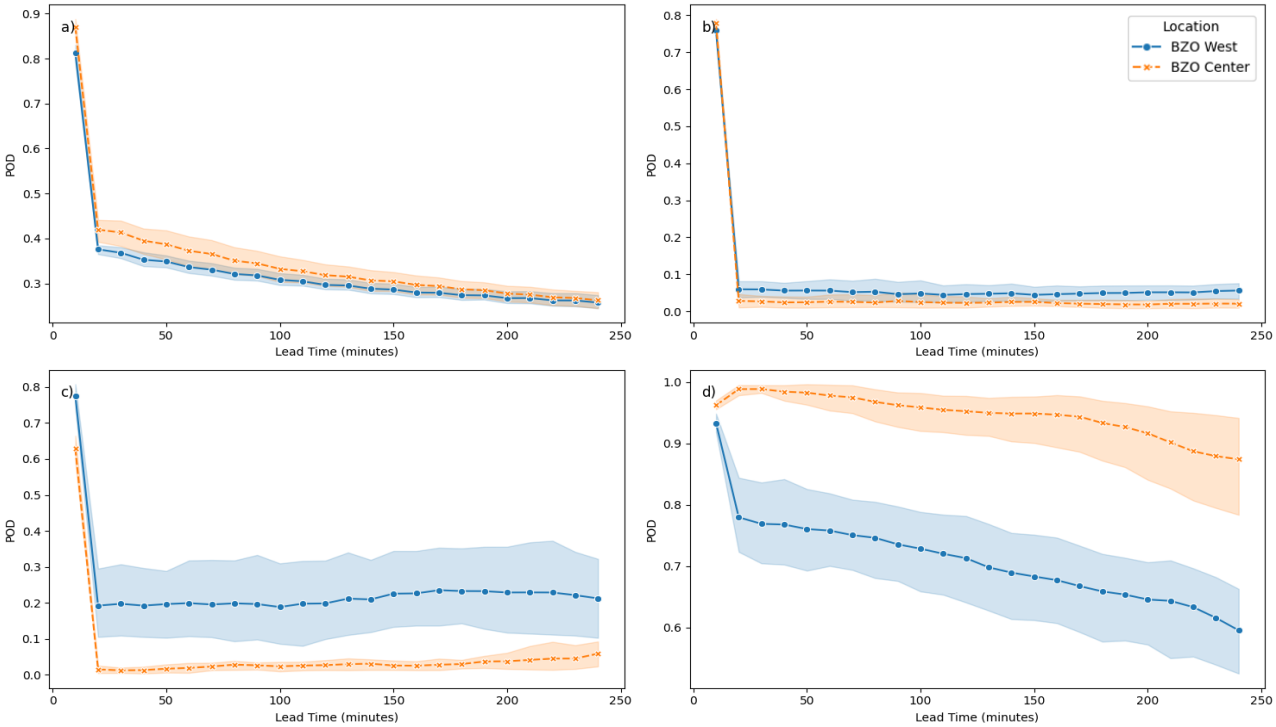

Figure 5.6: The POD for phase 'Good' (a), 'Marginal ' (b), 'A' (c), and 'B' (d) for the recursive Random Forest Classifier (Experiment 2) forecasting BZO phases based on vertical visibility. Both locations BZO West (blue) and BZO Center (orange) are shown. The shaded areas show the 95% confidence interval.

The confidence intervals of the POD of RFC-Direct are generally wide across all lead times, indicating substantial uncertainty in the model. For class 'A', the intervals are specifically wide at shorter lead times. Oppositely, in RFC-Recursive, the intervals are narrow, but POD scores are generally low, specifically for class 'A' in BZO Center and class 'Marginal'. The narrow interval indicates the model is confident about predictions, even though they are incorrect. High confidence in incorrect predictions can occur when an ML model uses features that are not representative of the target. In this case, this means that the previously classified visibility class is generally not representative of the class in the next timestep. This is somewhat to be expected, as clouds, and with that vertical visibility, usually show patterns on larger temporal scales than 10 minutes (Bony et al., 2015). An interesting finding is that the recursive RFC is capable of detecting class 'B', especially in BZO Center (average POD 0.946). We could not find an explanation for this pattern. To summarize, the direct RFCs are relatively uncertain in their predictions, while recursive RFCs are confident in predictions but are often incorrect.

5.3.2 Temporal Fusion Transformers

Finally, we will discuss the performance of the TFTs in forecasting BZO phases based on vertical visibility. Similar to the horizontal visibility subchapter, we discuss the experiments 'TFT-Focal-Loss', 'TFT-Weighted' and 'TFT-HARMONIE'. We assess deterministic performance using the POD and CSI, and assess the probabilistic forecasts using Brier Score, BSS Climatology and BSS Persistence.

Table 5.5: The Probability of Detection (POD), Critical Success Index (CSI), Brier Score, Climatological Brier Skill Score (BSS Climatology) and Persistence Brier Skill Score (BSS Persistence) for the Temporal Fusion Transformer forecasting BZO phases based on vertical visibility, for all three TFT-experiments. First value is the score at the first lead time (10 minutes for experiment 1 and 2, 1 hour for experiment 3). Second value in brackets is the average score over all lead times up to 4 hours. The best value for the specific category, over all experiments, is given in green/bold. The worst value for the specific category, over all experiments, is given in orange/italic.

Experiment	Category	POD	CSI	Brier Score	BSS Climatology	BSS Persistence
Experiment 1: TFT-Focal-Loss	Good	0.821 (0.843)	0.816 (0.836)	0.170 (0.174)	-8.679 (-8.827)	0.0 (-59.836)
	Marginal	0.283 (**0.307**)	0.030 (0.040)	0.067 (0.067)	*-6.423* (-6.347)	0.0 (*-13.775*)
	A	0.0 (0.0)	0.0 (0.0)	**0.004** (0.004)	-0.099 (-0.070)	0.0 (-0.575)
	B	0.440 (0.327)	0.019 (0.015)	0.052 (0.053)	-9.919 (-10.062)	0.0 (-35.183)
Experiment 2: TFT-Weighted	Good	0.543 (*0.428*)	0.543 (*0.427*)	0.407 (*0.435*)	-22.252 (*-23.551*)	0.0 (*-151.159*)
	Marginal	*0.002* (0.048)	*0.0* (0.005)	0.051 (**0.043**)	-4.699 (-3.687)	0.0 (-8.353)
	A	0.0 (0.0)	0.0 (0.0)	0.007 (*0.008*)	-0.848 (*-0.932*)	0.0 (*-1.818*)
	B	**0.982** (0.880)	0.013 (*0.009*)	0.229 (*0.256*)	-46.762 (*-52.192*)	0.0 (*-173.489*)
Experiment 3: TFT-HARMONIE	Good	0.987 (**0.991**)	**0.883** (0.881)	**0.085** (0.092)	**0.196** (0.131)	0.0 (**-0.753**)
	Marginal	0.127 (0.061)	**0.115** (0.057)	0.075 (*0.080*)	**0.138** (0.086)	0.0 (**-0.497**)
	A	0.0 (0.0)	0.0 (0.0)	0.006 (0.006)	**-0.012** (-0.018)	0.0 (**0.200**)
	B	0.039 (*0.037*)	**0.030** (0.028)	**0.016** (0.017)	**0.050** (0.038)	0.0 (**-0.509**)

Performance per Class

As a deterministic forecast, the best-performing class by far is class 'Good', outperforming all other classes in both metrics in both TFT-Focal-Loss and TFT-HARMONIE. Only in TFT-Weighted, the POD of class 'B' is substantially higher (0.982 versus 0.543 for 'Good') but the CSI is still much lower for class 'B' (0.013 versus 0.543 for 'Good'). Class 'A' was not forecasted in the deterministic forecast at all. While class 'B' shows higher POD values than 'Marginal' (e.g. 0.982 versus 0.002 in TFT-Weighted); the opposite is true for the CSI (0.030 for class 'B' versus 0.115 for 'Marginal' in TFT-HARMONIE). This implies that the model seems capable of detecting class 'B', especially in the weighted experiment, but introduces many false alarms for this class, reflected in low CSI scores.

For the probabilistic forecast, patterns are different. The best-performing class is 'A' (Brier Score 0.004 – 0.008), followed by 'B' (0.016 – 0.043), 'Marginal' (Brier Score 0.043 – 0.080) and finally 'Good' (Brier Score 0.085 – 0.435). Class 'B' does perform slightly worse than 'Marginal' in terms of Brier Skill Scores. Low-visibility categories perform best, which we also saw in the horizontal visibility experiments. Even though the Brier Scores are slightly higher than horizontal visibility, class 'A' outperforms the TFT by Wehrli et al. (2024) across all experiments, and all reduced visibility classes perform better than the Brier Score found by Parde et al. (2022) (0.13). However, compared to earlier experiments on horizontal visibility and RVR, the patterns in class-wise performance are less distinct.

Performance per Experiment

The order of performance differs greatly between the different experiments. The most prominent difference is that of class 'B' in TFT-Weighted. Introducing class weights has made the model overly biased towards this class, resulting in high detection rates, but also a high number of false alarms. However, for other classes, the effect of weights is different. For classes 'Good' and 'Marginal', both POD and CSI scores worsened from TFT-Focal-Loss to TFT-Weighted. This can be explained by the fact that class 'B' received a much higher weight (20) than class 'Good' (1) and 'Marginal' (2), causing the model to assign lower probabilities to these classes, resulting in lower deterministic performance.

We observe similar patterns in the Brier Score; which are higher in TFT-Weighted than in TFT-Focal Loss for classes 'Good' (0.407 versus 0.170), 'A' (0.007 versus 0.004) and 'B' (0.256 versus 0.053). Brier Skill Scores all also lowered in the TFT-Weighted experiment. The only exception is class 'Marginal', which performed slightly better in TFT-Weighted in terms of Brier Score (0.051 versus 0.067), BSS Climatology (-4.699 versus -6.423) and BSS Persistence (-8.353 versus -13.775). The weights appear to have very slightly improved the accuracy of probabilities for 'Marginal'. This suggests that weights do not necessarily always have a negative impact on model performance. Experimenting with different class weights could result in better performances for other classes.

The most prominent result of these experiments is the performance of TFT-HARMONIE. This experiment contains 14 out of 20 best values over all experiments. Moreover, it is the only experiment that has positive skill scores for some experiments (e.g. BSS Climatology Good: 0.196, Marginal: 0.138, B: 0.050; BSS Persistence A: 0.200). In terms of the CSI, the model also scores best over all experiments, both at short as well as longer lead times. We already saw a relatively better performance of RFC-HARMONIE for vertical visibility than for horizontal visibility, and this is even more prominent for the TFT. From this, we can conclude that using HARMONIE data as input for the TFT is useful when forecasting vertical visibility.

To summarize, introducing weights generally worsened performance of both the deterministic and probabilistic forecast. Only class 'B' received high scores for the deterministic forecast, because the model got overly confident in predicting this class. The model significantly improved in the TFT-HARMONIE experiment, which is the only experiment that shows positive skill towards a climatological or persistence forecast.

6 Discussion

In this study, we trained two types of ML algorithms, the Random Forest Classifier (RFC) and the Temporal Fusion Transformer (TFT), to forecast visibility categories at Amsterdam Airport Schiphol on lead times of up to 4 hours. Our aim was to assess the performance of independent RFC and TFT models in forecasting pre-defined visibility categories, at multiple locations and on relatively short lead times. In this section, we answer our research questions, discuss our main findings and how our work relates to the existing literature. We also discuss the limitations of this study and our recommendations for future research in section 6.1.

Random Forest Classifiers versus Temporal Fusion Transformers

Our first research question "How does a Random Forest Classifier perform in forecasting pre-defined visibility categories for two locations at Amsterdam Airport Schiphol, over a forecasting horizon of 4 hours?" can be answered as follows: our RFCs show relatively high performance at short lead times, but quickly decline over time. Independent, recursive models show the highest performance over all.

The RFC is a well-known, successful ML algorithm, recognized for its effectiveness in nowcasting and forecasting visibility (e.g. Dewi et al., 2020; Bartok et al., 2022; Castillo-Botón et al., 2022) as well as post-processing NWM data (e.g. Bartokóva et al., 2015; Thomasson, 2023). Consistent with this earlier research, our RFC models demonstrated high performances at short lead times, specifically in horizontal visibility BZO phases 'Good' and 'Marginal'. However, results quickly declined over the forecasting horizon, underperforming compared to earlier research on independent models.

Performance was highest for the most-abundant classes, but low-visibility classes generally show lower initial performance. This indicates that even after balancing classes, the model is biased towards majority classes. Additionally, recursive RFCs showed a higher performance on both short and longer lead times of up to 4 hours than direct RFCs. To the author's knowledge, there are no studies that benchmarked the use of direct and recursive approaches for the same visibility forecasting problem. Our finding that recursive RFCs perform better than direct RFCs, could be useful for future studies in deciding what approach is more suitable for their forecasting problem.

Our second research question "How does a Temporal Fusion Transformer perform in forecasting pre-defined visibility categories for two locations at Amsterdam Airport Schiphol, over a forecasting horizon of 4 hours?" can be answered as follows: the TFT shows accurate results across the entire forecasting horizon for all classes, when used as a probabilistic model. Performance is best for low-visibility classes. The high performance of TFTs in visibility forecasting shows the potential of the application of these relatively new ML models in the meteorological field.

The TFT is a relatively new model, introduced by Ross and Dollár (2017). The model was designed for multi-variate time-series forecasting and was only recently applied as a regressor to forecasting visibility conditions by Wehrli et al. (2024). Following their promising results, we applied the TFT to a similar forecasting problem, but as a classifier. To the author's knowledge, this was the first introduction of a TFT as a classifier for predicting visibility conditions. We also applied a relatively new loss function 'Focal Loss', designed to handle class imbalance. We found that TFTs showed high performance as a probabilistic forecast, specifically in low-visibility categories, outperforming the regressor by Wehrli et al. (2024). Our model also outperformed research studies on decision tree-based visibility forecasting (Ohashi & Hara, 2024). Moreover, our TF showed better skill in the most extreme low-visibility class 'D'

than the current forecast for Schiphol, HARMONIE-QRF. The application of a TFT could result in significant improvements in forecasting extremely low visibility at Schiphol.

The Focal Loss function also has the opportunity to assign weights to classes, which generally did not improve performance. We suspect that assigning weights skewed the natural balance of the dataset. Weighting made the TFT model too confident in rare classes, leading to many false positives. Lower probabilities reflect the natural low occurrence of fog events better. However, there were a few exceptions, indicating that the right balance in class weighting could positively influence predictions.

Class-wise performance

Our study was one of the few examples that assessed the performance of ML models per class. Our RFCs scored best at the boundaries of the visibility range, like class 'Good' and 'D'. This pattern was recognized by Castillo-Botón et al. (2022), whose model also struggled to forecast intermediate classes. Although model showed slightly higher performance overall, it is important to note that Castillo-Botón et al. (2022) defined classes based on a statistical analysis of the dataset. Oppositely, our classes were pre-defined based on LVNL (Air Traffic Control the Netherlands) regulations (see: Background Information – Low Visibility Operations at Amsterdam Airport Schiphol). This increases the complexity of the forecasting task, because distinctions between intermediate classes might be less apparent and, therefore, more challenging for the ML algorithm to detect. To the author's knowledge, there is no other research on independent RFCs that forecast visibility conditions in pre-defined classes. Our results indicate that models predict classes slightly less accurately when using pre-defined classes, but show similar patterns in relative class performance.

Conversely, our TFTs showed higher performance for low-visibility conditions, likely due to the probabilistic nature of the predictions. The probabilities allowed for more nuanced predictions, capturing the uncertainty inherent in rare events. We cannot compare our findings on class-wise performance since there has not been other research on TFT classifiers for forecasting visibility conditions.

Location-wise performance

Earlier research on machine learning (ML) algorithms for visibility conditions always focused on forecasting for either one location (Bartok et al., 2022; Castillo-Botón et al., 2022; Wehrli et al., 2024) or a NWM gridded domain ((Thomasson, 2023). In this research, we aimed to produce forecasts for two separate locations that were very closely related but showed differences in fog occurrence.

We found some differences in performances between the two forecasting locations (BZO West and BZO Center). BZO West generally showed higher performance for low-visibility classes. This is likely related to the slightly higher abundance of these classes in this location, as BZO West contained 1% more data in low-visibility classes. It suggests that models could improve further if more low-visibility data is available.

Furthermore, predictions were generally more confident in BZO West. We hypothesized this could be due to greater persistence or variability in low-visibility classes, but BZO West actually showed shorter class durations (see Appendix E), and the data in this location was less variable. Possibly, the higher confidence is also related to the higher abundance of low-visibility data in BZO West, but not all results confirmed this idea. Additionally, a possible explanation is that boundaries of conditions related to the classes were more distinctly defined in BZO West. We did not perform analyses on this.

A key takeaway from our research is that ML methods were able to distinguish between two locations that were very close in terms of distance but showed differences in meteorological conditions and fog occurrence. It shows the potential of using ML models for forecasts on small spatial scales.

Observational versus Numerical Weather Model input data

As explained (see Background Information – Machine Learning), many ML algorithms have been applied to visibility forecasting as post-processing techniques for Numerical Weather Models (NWM) (e.g. Bartoková et al., 2015; Thomasson, 2023). This research was mainly aimed to independently forecast visibility using ML, but we also performed experiments on using NWM data as input.

Our research question "How does the performance of each model change when including NWM data as input, e.g. using the model as a post-processing technique?" can be answered as follows: models that forecasted horizontal visibility with NWM data as input generally showed a similar or lower performance than independent ML algorithms, specifically at short lead times. This is an important result of this research, as it highlights that NWM data is not always necessary to produce accurate visibility forecasts. It shows the potential of using independent ML algorithms for visibility forecasting. Using ML models independently, without the input of NWM models, maximizes the benefits of using ML: it is quick, cheap, and easy to implement.

Horizontal Visibility versus RVR versus Vertical Visibility

Finally, we benchmarked the performance of models using different targets, namely horizontal visibility, RVR and vertical visibility. All targets have been investigated before (e.g. Boneh et al., 2015; Colabone et al., 2015; Guijo-Rubio et al., 2018) but only a few studies conducted studies with more than one target (e.g. Wehrli et al., 2024), To the author's knowledge we were the first to directly compare the performances of identical models these three different targets.

Generally, models performed best in forecasting BZO phases that were based on horizontal visibility. Introducing RVR into the dataset improved performance of intermediate classes like class 'A', but the sparseness of extreme values caused overall performance in low-visibility classes to decrease.

The models generally struggled with accurately forecasting BZO phases based on vertical visibility. These performances were lower than those found in other research (Wehrli et al., 2024). We suspect our data was of insufficient quality for accurate prediction of vertical visibility (see Limitations and Recommendations for Future Research). The introduction of HARMONIE did show the most improvement for vertical visibility forecasts compared to other targets. We suspect this because vertical visibility is more dependent on large-scale meteorological dynamics (Tiedtke, 1993; Matveev, 2012). Therefore, models struggle when using short-term observations as input only, but improve when using more large-scale and smoothed data like NWM data. An important takeaway for future research is that ML models show more potential for forecasting horizontal visibility and generally require larger-scale data when forecasting vertical visibility.

6.1 Limitations and Recommendations for Future Research

In this section, we elaborate on the limitations of this research. We base the limitations on four stages in the model training process: the quality of the input data, the preparation of the input data, the training and tuning of the models, and the final evaluation of the Random Forest Classifier (RFC) and the Temporal Fusion Transformer (TFT). We also discuss implications and recommendations for future research.

Data Quality

A key limitation of this study was the relatively short time range of the dataset. Only 5 years were available, of which 4 were used for training and 1 for evaluation. Herman and Schumacher (2016) concluded 3 years is the minimum amount of data needed to train a model to forecast visibility conditions. Even though our dataset should be sufficient, longer periods of data would have been beneficial for capturing rare events. The 5-year dataset also did not contain enough samples to train the TFT (see Results – Runway Visual Range). Moreover, the dataset used to train models on HARMONIE data was only 3 years long. As 1 year was used for evaluation, only 2 years remained for training the models. This was likely too short for the model to capture all meaningful relationships in the variables. For future research, we recommend expanding the dataset to expose the model to a sufficient amount of rare events.

Additionally, this HARMONIE dataset spanned from 2020 to 2023, while the observational dataset had a time range of 2012 through 2017. For this reason, we were not able to use the same observational variables as targets for these models. The distribution of fog events did not differ significantly (see Appendix E), but small variations might have slightly impacted model training. This therefore did not allow for a complete fair comparison of model performance. Future studies that aim to benchmark ML model performances on different types of input data should ensure these datasets have the same time coverage.

Both datasets contained a substantial amount of missing variables, mostly in the observational dataset. We used imputation methods to complete the dataset, which showed minimal differences in the dataset's statistics. However, the missing data may have prevented the model from understanding relationships between the target and variables that contained a large number of imputed variables. A longer dataset would also minimize this effect.

Initially, this research's aim was to produce visibility forecasts for multiple locations at Amsterdam Airport Schiphol. After reviewing the dataset, it appeared that data was not consistently available across all sensors. Because of that reason, we had to aggregate the data to form only two locations. This involved a simple aggregation of averaging all available variables over the BZO area. This process may have removed fine spatial and temporal details. Future research could look into more graceful methods for aggregating this data to maintain more details.

Finally, the availability of RVR data was most limited. RVR data was only recorded when horizontal visibility dropped below 1500 meters, and it was only recorded at active runways. Because we aimed to train the models on this variable, a full dataset was necessary. The issue was resolved by imputing RVR values into the horizontal visibility data whenever available. This approach may have altered the statistics of the dataset by skewing its distribution to higher values. When training on RVR, future studies should aim to create a full dataset of the variable, for example by manually calculating it using runway light luminance.

Dataset Preprocessing

To handle class imbalance, necessary to reduce bias in training RFCs, we applied balancing techniques to this dataset. The creation of synthetic samples in low-visibility classes compromised the temporal continuity of the data, likely affecting time-dependent predictions by obstructing the model in finding relevant temporal dependencies. The results also showed that the models were still biased towards majority classes, even after applying balancing techniques. Future research should look into applying

more extensive resampling techniques to further address this residual bias. Additionally, models like a TFT generally do not require balanced datasets. However, there are examples of research where balancing techniques have been applied before training a TFT (Zhang et al., 2022; Anjum et al., 2023; Luo et al., 2024). Future research might consider including balancing techniques, but maintaining the temporal continuity of the dataset would be challenging.

We recommend experimenting with ranges of lagged variables in RFC's and TFT's. Performance of the RFC models quickly deteriorated over lead times, which could be improved by adding more historical information. Including lags in TFT's could increase their awareness of diurnal and seasonal patterns.

For the TFT models, it was necessary to aggregate data into 10-minute intervals. We did so by averaging values over each 10-minute interval. This aggregation reduced the relative frequency of rare events, possibly making it more difficult for the models to train on these classes. Additionally, the larger timesteps could have smoothed the data too much, limiting the model's ability to capture short-term fluctuations. We recommend to do a more extensive research into the effect of possibly smoothing data to reduce the effects of noisiness, while still maintaining relevant short-term fluctuations.

Finally, we were limited in the scope of this study to execute extensive preprocessing steps on the datasets like outlier removal, noise reduction or further smoothing. As an example, another factor that likely limited reduced visibility were high rainfall intensities. It is recommended to take a more thorough approach to preprocessing, reducing the possible effects of noise on the model's performance.

Model Tuning

The limited scope of this research also limited our ability to thoroughly tune hyperparameters or conduct feature importance analyses, both of which could have significantly enhanced model performance. We performed some hyperparameter tuning for the TFT models, but as these models contain a vast amount of hyperparameters, a more thorough approach could still yield better results.

In addition, our experiments regarding class weighting were limited. We applied one set of class weights to the Focal Loss function, which appeared to negatively affect results. However, it might be worth experimenting with other sets of class weights. Our class weights were only one order of magnitude, which did not fully reflect the level of imbalance in the dataset. A more systematic approach could improve the model's ability to handle rare cases. However, since our results showed negative impact by class weighting, we do not expect that any class weighting will cause major improvements.

Finally, we did not explore the possibility of TFTs producing multiple outputs. This feature allows the model to produce probabilities for multiple targets simultaneously, possibly capturing more interdependencies between targets. We recommend future studies explore this option further.

Model Evaluation

The methods for calculating evaluation metrics may have slightly skewed results. The One-vs-Rest approach, used to calculate metrics for RFCs, likely impacted performance in boundary cases. In these cases, the model is forced to choose one class as a forecasted class, even though conditions may have been very similar to the adjacent class. This approach makes the results less nuanced.

When transforming the probabilistic forecast of a TFT to a deterministic forecast, we did so by taking the class with the highest probability as the 'predicted class'. In cases where the probabilities for multiple classes were close, this approach may have resulted in lower performance scores for classes that were ranked second or third. Alternatively, we recommend to use probability threshold. It allows the deterministic forecast to forecast multiple classes for a timestep, if multiple classes exceeded the threshold. This provides a more nuanced view of the model's certainty and performance.

7 Conclusion

In this study, we developed two types of classifying machine learning algorithms – Random Forest Classifiers (RFC) and Temporal Fusion Transformers (TFT) to forecast visibility conditions at Amsterdam Airport Schiphol over a lead time of 4 hours with timesteps of 10 minutes. We evaluated the models' performances using horizontal visibility, Runway Visual Range (RVR) and vertical visibility as determining variables for the target classes, focusing on performance in low-visibility classes.

In general, RFCs proved successful in predicting visibility conditions at short-time scales, particularly for high-visibility classes 'Good' and 'Marginal' and extreme classes like BZO phase 'D'. However, their performance declined quickly over the forecasting horizon, especially for intermediate classes. Recursive models generally performed better than direct models.

The TFTs, fitted with a custom loss function, performed well at classifying visibility categories. The models were explicitly capable of predicting low-visibility class probabilities, outperforming earlier research for these classes. High-visibility classes 'Good' and 'Marginal' also showed high performance. Assigning weights to the loss function worsened the model's performance.

All models performed better when forecasting horizontal visibility over vertical visibility. Introducing RVR as a target improved class performance of intermediate classes. Introducing NWM as input data sometimes improved performance of persistent classes, but negatively impacted predictions for rare events. Vertical visibility predictions improved most from including NWM data, because this target variable is more dependent on large-scale dynamics.

To conclude, this study introduced a the novel use of a Temporal Fusion Transformer fitted with a custom Focal Loss function to forecast visibility categories. We found this approach to be successful, performing similar or better than earlier research. We recommend continuing research on this model type, including more extensive hyperparameter tuning, feature importance analyses and graceful handling of class imbalance, and exploring the multi-output possibilities of the TFT to further refine forecasts.

References

Akiba, Takuya and Sano, Shotaro and Yanase, Toshihiko and Ohta, Takeru and Koyama, & Masanori. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. 25th International Conference on Knowledge Discovery and Data Mining,

Almeida, M. V., França, G. B., de Almeida, V. A., & Bonnet, S. M. (2023). Fog Forecast Model based on Machine Learning.

Anjum, N., Sathi, K. A., Hossain, M. A., & Dewan, M. A. (2023). A Temporal Transformer-Based Fusion Framework for Morphological Arrhythmia Classification. *Computers*, *12*(3).

Balali, F., Nouri, J., Nasiri, A., & Zhao, T. (2020). Machine Learning Principles. In F. Balali, J. Nouri, A. Nasiri, & T. Zhao (Eds.), *Data Intensive Industrial Asset Management: IoT-based Algorithms and Implementation* (pp. 115-157). Springer International Publishing. https://doi.org/10.1007/978-3-030-35930-0_8

Bari, D. (2018, 29 Oct.-1 Nov. 2018). Visibility Prediction Based on Kilometric NWP Model Outputs Using Machine-Learning Regression. 2018 IEEE 14th International Conference on e-Science (e-Science),

Bari, D., Bergot, T., & Tardif, R. (2023). Fog Decision Support Systems: A Review of the Current Perspectives. *Atmosphere*, *14*(8).

Bartok, J., Šišan, P., Ivica, L., Bartoková, I., Malkin Ondík, I., & Gaál, L. (2022). Machine Learning-Based Fog Nowcasting for Aviation with the Aid of Camera Observations. *Atmosphere*, *13*(10). https://www.mdpi.com/2073-4433/13/10/1684#B21-atmosphere-13-01684

Bartoková, I., Bott, A., Bartok, J., & Gera, M. (2015). Fog Prediction for Road Traffic Safety in a Coastal Desert Region: Improvement of Nowcasting Skills by the Machine-Learning Approach. *Boundary-Layer Meteorology*, *157*(3), 501-516. https://doi.org/10.1007/s10546-015-0069-x

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*(3), 1937-1967. https://doi.org/10.1007/s10462-020-09896-5

Bergot, T., & Koracin, D. (2021). Observation, Simulation and Predictability of Fog: Review and Perspectives. *Atmosphere*, *12*(2), 235. https://www.mdpi.com/2073-4433/12/2/235

Bergot, T., Terradellas, E., Cuxart, J., Mira, A., Liechti, O., Mueller, M., & Nielsen, N. W. (2007). Intercomparison of Single-Column Numerical Models for the Prediction of Radiation Fog. *Journal of Applied Meteorology and Climatology*, *46*(4), 504-521. https://doi.org/https://doi.org/10.1175/JAM2475.1

bewoners aanspreekpunt schiphol. (2022). *Jaarrapportage 2022*. https://bezoekbas.nl/wp-content/uploads/2023/03/BAS-jaarrapportage-2022Definitief.pdf

Boneh, T., Weymouth, G. T., Newham, P., Potts, R., Bally, J., Nicholson, A. E., & Korb, K. B. (2015). Fog Forecasting for Melbourne Airport Using a Bayesian Decision Network. *Weather and Forecasting*, *30*(5), 1218-1233. https://doi.org/https://doi.org/10.1175/WAF-D-15-0005.1

Bony, S., Stevens, B., Frierson, D. M. W., Jakob, C., Kageyama, M., Pincus, R.,...Webb, M. J. (2015). Clouds, circulation and climate sensitivity. *Nature Geoscience*, *8*(4), 261-268. https://doi.org/10.1038/ngeo2398

Breiman, L. (1984). *Classification and regression trees*. Routledge.

Bureau of Transportation Statistics. (2024). *On-Time Performance - Reporting Operating Carrier Flight Delays at a Glance*. U.S. Department of Transportation. Retrieved June 18 from https://www.transtats.bts.gov/OT_Delay/ot_delaycause1.asp?6B2r=FE&20=E

Cannemeijer, F., & Stalenhoef, A. H. C. (1977). *Occurrence and advation of fog at Amsterdam/Airport Schiphol*. https://cdn.knmi.nl/knmi/pdf/bibliotheek/knmipubWR/WR77-12.pdf

Castillo-Botón, C., Casillas-Pérez, D., Casanova-Mateo, C., Ghimire, S., Cerro-Prada, E., Gutierrez, P. A.,...Salcedo-Sanz, S. (2022). Machine learning regression and classification methods for fog events prediction. *Atmospheric Research*, *272*, 106157. https://doi.org/https://doi.org/10.1016/j.atmosres.2022.106157

Cateni, S., Colla, V., & Vannucci, M. (2014). A method for resampling imbalanced datasets in binary classification tasks for real-world problems. *Neurocomputing*, *135*, 32-41. https://doi.org/https://doi.org/10.1016/j.neucom.2013.05.059

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, *6*(1), 1–6. https://doi.org/10.1145/1007730.1007733

Colabone, R. d. O., Ferrari, A. L., Vecchia, F. A. d. S., & Tech, A. R. B. (2015). Application of Artificial Neural Networks for Fog Forecast. *Journal of Aerospace Technology and Management*, *7*. https://www.scielo.br/j/jatm/a/zdVF6BYgGvCx8ss9YXygFSM/?format=html&lang=en

Cornejo-Bueno, L., Casanova-Mateo, C., Sanz-Justo, J., Cerro-Prada, E., & Salcedo-Sanz, S. (2017). Efficient Prediction of Low-Visibility Events at Airports Using Machine-Learning Regression. *Boundary-Layer Meteorology*, *165*(2), 349-370. https://doi.org/10.1007/s10546-017-0276-8

Coy, S. (2006). A global model for estimating the block time of commercial passenger aircraft. *Journal of Air Transport Management*, *12*(6), 300-305. https://doi.org/https://doi.org/10.1016/j.jairtraman.2006.07.005

De Villiers, M., & Van Heerden, J. (2007). Fog at Abu Dhabi international airport.

Dewi, R., Prawito, & Harsa, H. (2020). Fog prediction using artificial intelligence: A case study in Wamena Airport. *Journal of Physics: Conference Series*, *1528*(1), 012021. https://doi.org/10.1088/1742s6596/1528/1/012021

Dholakiya, P. (2023). *SMOTE(Synthetic Minority Over-sampling Technique)*. Medium. Retrieved July 24, 2024 from https://medium.com/@parthdholakiya180/smote-synthetic-minority-over-sampling-technique-4d5a5d69d720

Dijkstra, F. (2024). Talk about thesis progress. In V. Buis (Ed.).

Dione, C., Haeffelin, M., Burnet, F., Lac, C., Canut, G., Delanoë, J.,...Toledo, F. (2023). Role of thermodynamic and turbulence processes on the fog life cycle during SOFOF3D experiment. *EGUsphere*, *2023*, 1-46. https://doi.org/10.5194/egusphere-2023-1224

Dissanayaka, D. M. M. S., Adikariwattage, V., & Pasindu, H. R. (2019, 2019/10). Evaluation of Emissions from Delayed Departure Flights at Bandaranaike International Airport (BIA). Proceedings of the 11th Asia Pacific Transportation and the Environment Conference (APTE 2018),

Durán-Rosal, A. M., Fernández, J. C., Casanova-Mateo, C., Sanz-Justo, J., Salcedo-Sanz, S., & Hervás-Martínez, C. (2018). Efficient fog prediction with multi-objective evolutionary neural networks. *Applied Soft Computing*, *70*, 347-358. https://doi.org/https://doi.org/10.1016/j.asoc.2018.05.035

Duynkerke, P. G. (1999). Turbulence, Radiation and fog in Dutch Stable Boundary Layers. *Boundary-Layer Meteorology*, *90*(3), 447-477. https://doi.org/10.1023/A:1026441904734

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In I. El Naqa, R. Li, & M. J. Murphy (Eds.), *Machine Learning in Radiation Oncology: Theory and Applications* (pp. 3-11). Springer International Publishing. https://doi.org/10.1007/978-3-319-18305-3_1

Fabbian, D., de Dear, R., & Lellyett, S. (2007). Application of Artificial Neural Network Forecasts to Predict Fog at Canberra International Airport. *Weather and Forecasting*, *22*(2), 372-381. https://doi.org/https://doi.org/10.1175/WAF980.1

Federal Aviation Administration. Visibility. In *Pilot/Controller Glossary*. Retrieved August 19, 2024, from https://www.faa.gov/air_traffic/publications/atpubs/pcg_html/glossary-v.html#:~:text=VISIBILITY%20%5BICAO%5D%2D%20The%20ability,of%20an%20aircraft%20in%20flight.

Benefit-Cost Analysis, (2022). https://www.faa.gov/regulations_policies/policy_guidance/benefit_cost

Gautam, R., & Singh, M. K. (2018). Urban Heat Island Over Delhi Punches Holes in Widespread Fog in the Indo-Gangetic Plains. *Geophysical Research Letters*, *45*(2), 1114-1121. https://doi.org/https://doi.org/10.1002/2017GL076794

GMAP. *Parking Spots & Runways Schiphol Airport*. Global Military Aviation Photography. Retrieved July 10th from https://gmap.nl/parking-spots-runway-schiphol/

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Goodwin, L., & Pisano, P. (2003). Weather-related crashes on US highways in 2001. *Mitretek Systems, Inc., Rep. prepared for US Dept. of Transportation*, *8*.

Gorshenin, A. K., & Lukina, S. S. (2021, 2021//). On the Efficiency of Machine Learning Algorithms for Imputation in Spatiotemporal Meteorological Data. Advances in Artificial Systems for Medicine and Education IV, Cham.

Guijo-Rubio, D., Gutiérrez, P. A., Casanova-Mateo, C., Sanz-Justo, J., Salcedo-Sanz, S., & Hervás-Martínez, C. (2018). Prediction of low-visibility events due to fog using ordinal classification. *Atmospheric Research*, *214*, 64-73. https://doi.org/https://doi.org/10.1016/j.atmosres.2018.07.017

Guo, L., Guo, X., Luan, T., Zhu, S., & Lyu, K. (2021). Radiative effects of clouds and fog on long-lasting heavy fog events in northern China. *Atmospheric Research*, *252*, 105444. https://doi.org/https://doi.org/10.1016/j.atmosres.2020.105444

Han, J. H., Kim, K. J., Joo, H. S., Han, Y. H., Kim, Y. T., & Kwon, S. J. (2021). Sea Fog Dissipation Prediction in Incheon Port and Haeundae Beach Using Machine Learning and Deep Learning. *Sensors*, *21*(15).

Hang, C., Nadeau, D. F., Gultepe, I., Hoch, S. W., Román-Cascón, C., Pryor, K.,...Pardyjak, E. R. (2016). A Case Study of the Mechanisms Modulating the Evolution of Valley Fog. *Pure and Applied Geophysics*, *173*(9), 3011-3030. https://doi.org/10.1007/s00024-016-1370-4

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D.,...Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2

Herman, G. R., & Schumacher, R. S. (2016). Using Reforecasts to Improve Forecasting of Fog and Visibility for Aviation. *Weather and Forecasting*, *31*(2), 467-482. https://doi.org/https://doi.org/10.1175/WAF-D-15-0108.1

Hingmire, D., Vellore, R., Krishnan, R., Singh, M., Metya, A., Gokul, T., & Ayantika, D. C. (2022). Climate change response in wintertime widespread fog conditions over the Indo-Gangetic Plains. *Climate Dynamics*, *58*(9), 2745-2766. https://doi.org/10.1007/s00382-021-06030-1

Huang, H., & Chen, C. (2016). Climatological aspects of dense fog at Urumqi Diwopu International Airport and its impacts on flight on-time performance. *Natural Hazards*, *81*(2), 1091-1106. https://doi.org/10.1007/s11069-015-2121-z

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90-95. https://doi.org/10.1109/MCSE.2007.55

Annex 3 - Meteorological Service for International Air Navigation, 1 § 1 5 (2018a). https://www.icao.int/airnavigation/IMP/Documents/Annex%203%20-%2075.pdf

Operation of Aircraft, § Annex 6 (2018b). https://ffac.ch/wp-content/uploads/2020/09/ICAO-Annex-6-Operation-of-Aircraft-Part-I-International-commercial-air-transport.pdf

Inoue, M., Fraser, A. D., Adams, N., Carpentier, S., & Phillips, H. E. (2015). An Assessment of Numerical Weather Prediction–Derived Low-Cloud-Base Height Forecasts. *Weather and Forecasting*, *30*(2), 486-497. https://doi.org/https://doi.org/10.1175/WAF-D-14-00052.1

Izett, J. G., van de Wiel, B. J. H., Baas, P., & Bosveld, F. C. (2018). Understanding and Reducing False Alarms in Observational Fog Prediction. *Boundary-Layer Meteorology*, *169*(2), 347-372. https://doi.org/10.1007/s10546-018-0374-2

Izett, J. G., van de Wiel, B. J. H., Baas, P., van Hooft, J. A., & Schulte, R. B. (2019). Dutch fog: On the observed spatio-temporal variability of fog in the Netherlands. *Quarterly Journal of the Royal Meteorological Society*, *145*(723), 2817-2834. https://doi.org/https://doi.org/10.1002/qj.3597

Keith, R., & Leyton, S. M. (2007). An Experiment to Measure the Value of Statistical Probability Forecasts for Airports. *Weather and Forecasting*, *22*(4), 928-935. https://doi.org/https://doi.org/10.1175/WAF988.1

Kim, B.-Y., Belorid, M., & Cha, J. W. (2022). Short-Term Visibility Prediction Using Tree-Based Machine Learning Algorithms and Numerical Weather Prediction Data. *Weather and Forecasting*, *37*(12), 2263-2274. https://doi.org/https://doi.org/10.1175/WAF-D-22-0053.1

Kim, B.-Y., Cha, J. W., Chang, K.-H., & Lee, C. (2021). Visibility Prediction over South Korea Based on Random Forest. *Atmosphere*, *12*(5).

KNMI. *Daggegevens van het weer in Nederland*. KNMI. Retrieved July 21st, 2024 from https://www.knmi.nl/nederland-nu/klimatologie/daggegevens

KNMI. *Dagwaarnemingen van weerstations* (KNMI. https://www.daggegevens.knmi.nl/

KNMI. *Dichte mist*. https://www.knmi.nl/kennis-en-datacentrum/waarschuwingen/zicht#:~:text=Mist%20is%20beperking%20van%20het,dan%20200%20meter%20is%20gevaarlijk.

KNMI. *Klimaatviewer*. Retrieved July 24th, 2024 from https://www.knmi.nl/klimaat-viewer

Koenker, R., & Bassett, G. (1978). Regression Quantiles. *Econometrica*, *46*(1), 33-50. https://doi.org/10.2307/1913643

Koziara, M. C., Renard, R. J., & Thompson, W. J. (1983). Estimating Marine Fog Probability Using a Model Output Statistics Scheme. *Monthly Weather Review*, *111*(12), 2333-2340. https://doi.org/https://doi.org/10.1175/1520-0493(1983)111<2333:EMFPUA>2.0.CO;2

Lakra, K., & Avishek, K. (2022). A review on factors influencing fog formation, classification, forecasting, detection and impacts. *Rendiconti Lincei. Scienze Fisiche e Naturali*, *33*(2), 319-353. https://doi.org/10.1007/s12210-022-01060-1

Leung, A. C. W., Gough, W. A., & Butler, K. A. (2020). Changes in Fog, Ice Fog, and Low Visibility in the Hudson Bay Region: Impacts on Aviation. *Atmosphere*, *11*(2), 186. https://www.mdpi.com/2073-4433/11/2/186

Li, X., & Pu, Z. (2024). Effects of surface moisture flux on the formation and evolution of cold fog over complex terrain with large-eddy simulation. *Quarterly Journal of the Royal Meteorological Society*, *150*(762), 3013-3027. https://doi.org/https://doi.org/10.1002/qj.4748

Liang, G. (2013, 2013//). An Effective Method for Imbalanced Time Series Classification: Hybrid Sampling. AI 2013: Advances in Artificial Intelligence, Cham.

Liu, G. (2024). Predicting Winter Fog over Complex Terrain using Machine Learning. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www-old.cs.utah.edu/docs/techreports/2024/UUCS-24-002.pdf

Beperkt Zicht Omstandigheden (BZO), (2022).

Luo, H., Zheng, Y., Chen, K., & Zhao, S. (2024). Probabilistic Temporal Fusion Transformers for Large-Scale KPI Anomaly Detection. *IEEE Access*, *12*, 9123-9137. https://doi.org/10.1109/ACCESS.2024.3353201

LVNL. (2022, October 31 2022). *Fewer delayed flights during fog due to new LVNL procedure*. LVNL. Retrieved May 8 2024 from

LVNL. (2023). *Innovation Leads to Fewer Delays for Airlines and Travellers*. LVNL. Retrieved June 17, 2023 from https://en.lvnl.nl/news/innovation-leads-to-fewer-delays-for-airlines-and-travellers

Martinet, P., Cimini, D., Burnet, F., Ménétrier, B., Michel, Y., & Unger, V. (2020). Improvement of numerical weather prediction model analysis during fog conditions through the assimilation of ground-based microwave radiometer observations: a 1D-Var study. *Atmos. Meas. Tech.*, *13*(12), 6593-6611. https://doi.org/10.5194/amt-13-6593-2020

Mastorakis, G. (2018). Human-like machine learning: limitations and suggestions. *arXiv preprint arXiv:1811.06052*.

Matveev, L. T. (2012). *Cloud dynamics* (Vol. 2). Springer Science & Business Media.

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. https://doi.org/10.25080/Majora-92bf1922-00a

MetOffice. *How we measure visibility*. Crown. Retrieved 2024 from https://www.metoffice.gov.uk/weather/guides/observations/how-we-measure-visibility

Miao, K.-c., Han, T.-t., Yao, Y.-q., Lu, H., Chen, P., Wang, B., & Zhang, J. (2020). Application of LSTM for short term fog forecasting based on meteorological elements. *Neurocomputing*, *408*, 285-291. https://doi.org/https://doi.org/10.1016/j.neucom.2019.12.129

Miao, Y., Potts, R., Huang, X., Elliott, G., & Rivett, R. (2012). A Fuzzy Logic Fog Forecasting Model for Perth Airport. *Pure and Applied Geophysics*, *169*(5), 1107-1119. https://doi.org/10.1007/s00024-011-0351-x

Moniz, N., Branco, P., & Torgo, L. (2017). Resampling strategies for imbalanced time series forecasting. *International Journal of Data Science and Analytics*, *3*(3), 161-181. https://doi.org/10.1007/s41060-017-0044-3

Mullers, S. (2023). *An investigation of the unnecessary costs caused by regulations for weather reasons* [Internship Report]. LVNL

VU Amsterdam.

Negishi, M., & Kusaka, H. (2022). Development of statistical and machine learning models to predict the occurrence of radiation fog in Japan. *Meteorological Applications*, *29*(2), e2048. https://doi.org/https://doi.org/10.1002/met.2048

Office of Aviation Consumer Protection. (2023). *Air Travel Consumer Report*. U. S. D. o. Transportation. https://www.transportation.gov/resources/individuals/aviation-consumer-protection/air-travel-consumer-reports-2023

Ohashi, Y., & Hara, K. (2024). AI-Driven Forecasting for Morning Fog Expansion (Sea of Clouds). *Weather and Forecasting*, *39*(10), 1387-1398. https://doi.org/https://doi.org/10.1175/WAF-D-23-0237.1

Ortega, L., Otero, L. D., & Otero, C. (2019, 8-11 April 2019). Application of Machine Learning Algorithms for Visibility Classification. 2019 IEEE International Systems Conference (SysCon),

Parde, A. N., Ghude, S. D., Dhangar, N. G., Lonkar, P., Wagh, S., Govardhan, G.,...Jenamani, R. K. (2022). Operational Probabilistic Fog Prediction Based on Ensemble Forecast System: A Decision Support System for Fog. *Atmosphere*, *13*(10).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z.,...Lerer, A. (2017). Automatic differentiation in PyTorch. NIPS 2017 Workshop,

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O.,...Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Pejovic, T., Williams, V. A., Noland, R. B., & Toumi, R. (2009). Factors Affecting the Frequency and Severity of Airport Weather Delays and the Implications of Climate Change for Future Delays. *Transportation Research Record*, *2139*(1), 97-106. https://doi.org/10.3141/2139-12

Penov, N., & Guerova, G. (2023). Sofia Airport Visibility Estimation with Two Machine-Learning Techniques. *Remote Sensing*, *15*(19).

Rodríguez-Sanz, Á., Cano, J., & Rubio Fernandez, B. (2022). Impact of weather conditions on airport arrival delay and throughput. *Aircraft engineering and aerospace technology*, *94*(1), 60-78.

Roebber, P. J. (2009). Visualizing Multiple Measures of Forecast Quality. *Weather and Forecasting*, *24*(2), 601-608. https://doi.org/https://doi.org/10.1175/2008WAF2222159.1

Román-Cascón, C., Yagüe, C., Sastre, M., Maqueda, G., Salamanca, F., & Viana, S. (2012). Observations and WRF simulations of fog events at the Spanish Northern Plateau. *Adv. Sci. Res.*, *8*(1), 11-18. https://doi.org/10.5194/asr-8-11-2012

Román-Cascón, C., Yagüe, C., Steeneveld, G.-J., Morales, G., Arrillaga, J. A., Sastre, M., & Maqueda, G. (2019). Radiation and cloud-base lowering fog events: Observational analysis and evaluation of WRF and HARMONIE. *Atmospheric Research*, *229*, 190-207. https://doi.org/https://doi.org/10.1016/j.atmosres.2019.06.018

Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

Ross, T.-Y., & Dollár, G. (2017). Focal loss for dense object detection. proceedings of the IEEE conference on computer vision and pattern recognition,

Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M., & Bagnall, A. (2021). The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, *35*(2), 401-449. https://doi.org/10.1007/s10618-020-00727-3

Salcedo-Sanz, S., Pérez-Aracil, J., Ascenso, G., Del Ser, J., Casillas-Pérez, D., Kadow, C.,...Restelli, M. (2022). Analysis, characterization, prediction and attribution of extreme atmospheric events with machine learning: a review. *arXiv preprint arXiv:2207.07580*.

Schiphol. (2017). *Waarom vlieg ik toch altijd vanaf de Polderbaan?* Retrieved June 19 from https://nieuws.schiphol.nl/waarom-vlieg-ik-toch-altijd-vanaf-de-polderbaan/

Schiphol, L. (2016). 1.2.3 Beperkt Zicht Omstandigheden (BZO). In A. Operations, A. Management, F. Brigade, & H. O. A. A. Schiphol (Eds.), *Handboeken Business Area Aviation*. Amsterdam Airport Schiphol.

Schultz, M. G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L. H.,...Stadtler, S. (2021). Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *379*(2194), 20200097. https://doi.org/10.1098/rsta.2020.0097

Shafer, A., & Victor, D. (1997). The Past and Future of Global Mobility. *Scientific American Magazine*, *277*(4), 1. https://doi.org/10.1038/scientificamerican1097-58

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379-423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Spektor, I. (2023). *Direct vs Recursive Multi-Step Forecasting*

Steeneveld, G. J., Ronda, R. J., & Holtslag, A. A. M. (2015). The Challenge of Forecasting the Onset and Development of Radiation Fog Using Mesoscale Atmospheric Models. *Boundary-Layer Meteorology*, *154*(2), 265-289. https://doi.org/10.1007/s10546-014-9973-8

Stolaki, S., Kazadzis, S., Foris, D., & Karacostas, T. S. (2009). Fog characteristics at the airport of Thessaloniki, Greece. *Natural hazards and earth system sciences*, *9*(5), 1541-1549.

Surakhi, O., Zaidan, M. A., Fung, P. L., Hossein Motlagh, N., Serhan, S., AlKhanafseh, M.,...Hussein, T. (2021). Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm. *Electronics*, *10*(20).

Tag, P. M., & Peak, J. E. (1996). Machine Learning of Maritime Fog Forecast Rules. *Journal of Applied Meteorology and Climatology*, *35*(5), 714-724. https://doi.org/https://doi.org/10.1175/1520-0450(1996)035<0714:MLOMFF>2.0.CO;2

Tapiador, F. J., Sánchez, J.-L., & García-Ortega, E. (2019). Empirical values and assumptions in the microphysics of numerical models. *Atmospheric Research*, *215*, 214-238. https://doi.org/https://doi.org/10.1016/j.atmosres.2018.09.010

Thomasson, A. (2023). Improving Visibility Forecasts in Denmark Using Machine Learning Post-processing. In. Uppsala: Department of Earth Sciences, Uppsala University.

Tiedtke, M. (1993). Representation of Clouds in Large-Scale Models. *Monthly Weather Review*, *121*(11), 3040-3061. https://doi.org/https://doi.org/10.1175/1520-0493(1993)121<3040:ROCILS>2.0.CO;2

Tijm, S. (2024). *Weather model - HARMONIE-AROME Cy43 reforecast Netherlands meteorological parameters* [Dataset]. https://dataplatform.knmi.nl/dataset/harmonie-arome-cy43-p1-reforecast-1-0

Tonkelaar, J. F., den. (n.d.). *De afname van de frekwentie van de stralinstmist in Schiphol gedurende het tijdvak 1 januari 1949 t/m 31 december 1959, suggesties voor een mogelijke oorzaak*. https://cdn.knmi.nl/knmi/pdf/bibliotheek/knmipubVerslag/Verslag78.pdf

van der Velde, I. R., Steeneveld, G. J., Wichers Schreur, B. G. J., & Holtslag, A. A. M. (2010). Modeling and Forecasting the Onset and Duration of Severe Radiation Fog under Frost Conditions. *Monthly Weather Review*, *138*(11), 4237-4253. https://doi.org/https://doi.org/10.1175/2010MWR3427.1

van Oldenborgh, G. J., Yiou, P., & Vautard, R. (2010). On the roles of circulation and aerosols in the decline of mist and dense fog in Europe over the last 30 years. *Atmos. Chem. Phys.*, *10*(10), 4597-4609. https://doi.org/10.5194/acp-10-4597-2010

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,...SciPy, C. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261-272. https://doi.org/10.1038/s41592-019-0686-2

Wantuch, F. (2001). Visibility and fog forecasting based on decision tree method. *Idojárás*, *105*, 29-38. https://www.researchgate.net/publication/228420283_Visibility_and_fog_forecasting_based_on_decision_tree_method

Wehrli, K., Attinger, R., Barras, H., Landmann, J. M., Aznar-Siguan, G., Exterde, S.,...Stocker, C. (2024). Using machine learning to enhance visibility predictions at Zurich Airport. EMS Annual Meeting 2024, Barcelona, Spain.

Westcott, N. E. (2007). Some aspects of dense fog in the Midwestern United States. *Weather and Forecasting*, *22*(3), 457-465.

Widmer, G., & Kubat, M. (1996). Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, *23*(1), 69-101. https://doi.org/10.1023/A:1018046501280

Wolters, D., Schmeits, M., & Whan, K. (n.d.). *Probabilistic forecasting of cloud base height and visibility using quantile regression forests, based on NWP and observation features*. KNMI.

Wu, Y., Abdel-Aty, M., & Lee, J. (2018). Crash risk analysis during fog conditions using real-time traffic data. *Accident Analysis & Prevention*, *114*, 4-11. https://doi.org/https://doi.org/10.1016/j.aap.2017.05.004

Yang, L., Ding, S., Liu, J. W., & Zhang, S. P. (2023). Effects of longwave radiative cooling on advection fog over the Northwest Pacific Ocean: Observations and large eddy simulations. *EGUsphere*, *2023*, 1-27. https://doi.org/10.5194/egusphere-2023-1494

Yu, Z., Qu, Y., Wang, Y., Ma, J., & Cao, Y. (2021). Application of Machine-Learning-Based Fusion Model in Visibility Forecast: A Case Study of Shanghai, China. *Remote Sensing*, *13*(11).

Zhang, W., Qiu, F., Wang, S., Zeng, H., Zhang, Z., An, R.,...Ding, Y. (2022). Transformer-based multimodal information fusion for facial expression analysis. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Zhen, M., Yi, M., Luo, T., Wang, F., Yang, K., Ma, X.,...Li, X. (2023). Application of a Fusion Model Based on Machine Learning in Visibility Prediction. *Remote Sensing*, *15*(5).

Zhou, B., Du, J., Gultepe, I., & Dimego, G. (2012). Forecast of Low Visibility and Fog from NCEP: Current Status and Efforts. *Pure and Applied Geophysics*, *169*(5), 895-909. https://doi.org/10.1007/s00024-011-0327-x