UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA
DIPARTIMENTO DI INFORMATICA, SISTEMISTICA E COMUNICAZIONE
CORSO DI LAUREA MAGISTRALE IN DATA SCIENCE

Weather forecast quality assessment and power grid faults prediction

Supervisor: Prof. Matteo Palmonari
Co-supervisor: Dott. Volker Hoffmann

Candidate:
DAVID GOVI
NUM. 833653

Academic Year 2018/2019

"All models are wrong. Some are lethal."

Nassim Nicholas Taleb

Abstract

This work aims to predict large scale fault occurrences in the Norwegian power grid distribution using weather data. Being able to anticipate such faults means saving large amount of resources and avoiding disservices for costumers. Develope machine learning predictive models and improve what was already done at SINTEF requires to address a number of critical issues. First and foremost, the need to rely on weather forecast in order to produce predictions in advance. Especially in Norway-like countries weather is unstable, making forecasts not always reliable. Accounting for forecast error can improve machine learning model performances and increase reliability of thus made fault predictions. To evaluate quality of forecasts it is necessary to compare their values with real weather observation made available through weather stations distributed within Norway's boundaries. Forecast quality is not uniform in time and space and understanding its nature turns out to be a valuable piece of information. While evaluating the behaviour of forecast error in time requires analysing the right portion of data and isolating pattern of interest, evaluating the behaviour of forecast in different points in space benefits of a description of the area surrounding a certain point. This is obtained by integrating the so-called Digital Elevation Model. DEMs provide high resolution elevation data. Not only elevation is a valuable piece of information by itself, but it can also be used to calculate various terrain descriptors such as, for instance, slope and roughness. Time and terrain features, properly modeled, can explain a large portion of forecast error variance. Forecast quality is not the only concern while predicting faults. Other challenges, such as the impossibility to know the exact point where a fault occurred, the need to deal with high class imbalance and effects of seasonality on need to be addressed.
Eventually, using all findings out of this analysis, this thesis work succeed in its goal by improving the results of previous studies in terms of model performances and fault prediction accuracy.

Ringraziamenti

The end of a journey always requires a moment of recollection to sit and reflect on the path one have travelled.
I had the luck to count on great professors and personalities. Among these, I need to thank, from an academical and professional standpoint, Matteo Palmonari and Volker Hoffmann, for their patience, their support and to allow me to spend three months in a top-class research institute other than to serve as supervisors for this work. Thank you.
What touches my heart the most though is the number of incredible people I've been so lucky to meet during this six year journey. From people I met only once, to people I lost on the way, to those who are still by my side to this day. Thank you.
Especially to the latter, who always provided shelter and help to navigate life, thank you again. I owe you huge debt of gratitude.
Through Pisa, Leicester, Milano, Oslo, there were doubts, change of course, struggles, challenges, but I haven't heard of men growing through straight paths.

Contents

1

Introduction

Before starting to go through the methodological and technical aspects of this thesis it is necessary to present to the reader with the required background to better understand context and real life scenario where this very work fits.
This thesis intersects multiple topics. In this first chapter we introduce the idea of power grid and smart grid while making a link with weather, and so weather forecasts, with the aim of motivating where the problems here addressed originate from.

1.1 Context and background

Stable power provisions have a primary role in contemporary societies. Electricity is not only essential in everyday life, but also covers a central role in industrial sectors and in delivery of public and private services. To guarantee high quality of power distribution is of the utmost importance for modern societies.
Because of this condition of strong dependency, even short-term fluctuations in the power system may lead to relevant, or even dangerous, consequences on economical and societal levels [50, 39].

To safeguard power systems and ensure their correct functioning is a high priority for any country. Therefore, being able to detect faults and weaknesses in the power system before problems escalate and cause breakdowns can be absolutely invaluable [51].
The ultimate goal of this work is, indeed, to build machine learning models for prediction of fault occurrences in order to allow providers to prevent such breakdowns and/or to put into play alternative strategies and adapt their distribution structure.

1.1.1 Power system

The electrical power system, or electrical grid, can be divided into three major levels:

- **Power supply**: generators and power plants.
- **Power transmission**: transmission grid.
- **Power distribution**: distribution grid that leads the electricity to all receivers.

Electricity is produced by power plants, a sort of generators, then goes through the transmission grid that operates at high voltages in the range of 155.000 V to 765.000 V for distance that can arrive to cover 500 km. Then electricity passes to an electrical substation in order to transform the voltage to values transformed for private distribution, in the 2.000 V scope, on its path to the distribution grid.
Eventually, electrical power is delivered to the final consumer through the distribution grid [39].

There are various ways and methods of producing electrical energy. Deciding what type of power plant to use, in a certain environment, strongly depends on location-related circumstances. These can be, for example, access to primary sources, local requirements, demand, laws and regulations, and energy strategy of a given country. Regardless of the type of generating station, where good example are hydroelectric, thermal, nuclear or based on wind power, the main principles and pipeline of the transmission and distribution grid remains the same. [42]

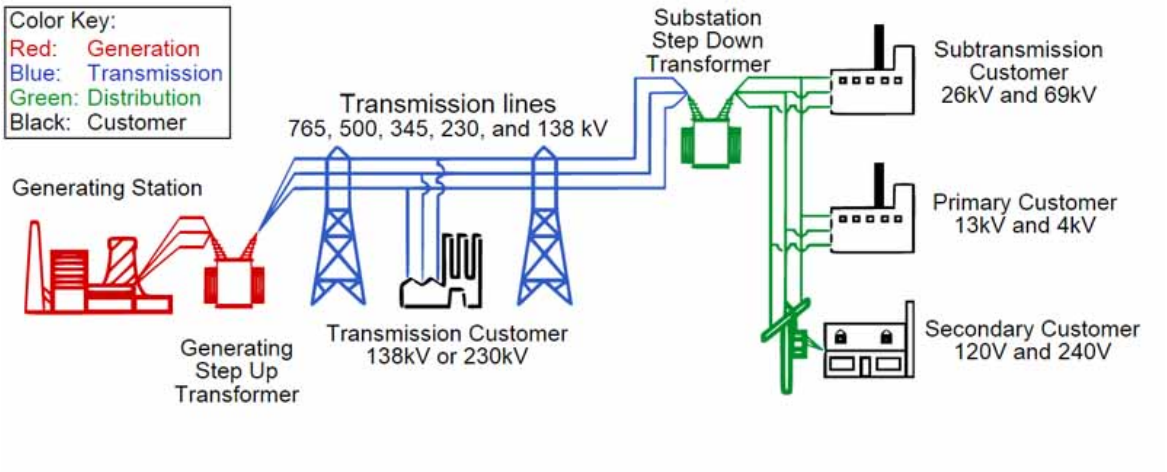

Figure 1.1 Power grid tree-level pipeline

In Norway, the main energy source for the generation of electricity is hydropower. Hydropower covers around 96% of the total annual power production in the country. Hydroelectric plants use terrain height differences and work by converting the potential energy of falling water into kinetic energy. Kinetic energy is then further converted to electricity with the help of water turbines [57]. If we include other renewable resources, thermal power and wind power, the percentage of electrical energy acquired from renewable energy sources, in Norway, reaches 98% [20].

The current state of the Norwegian energy production and consumption are a result of a reiterated effort and of strong social and political will in the country [19]. The conversion to mass-distributed electric engines of the transportation sector serves as a mirror for this willingness. Thanks to stimulating public policies and numerous incentives, Norway already has the largest number of electric vehicles per capita in the world [14]. Road transport is just an example though, as the use of electric power in the Norwegian transportation sector is also extended to maritime transport and new areas of life.

1.1.2 Smart grid

The importance of power systems makes them a focal point for research and innovation projects. In this sense, the idea of Smart Grid has been buzzing around for years [12].

Smart Grid is a term that encloses the whole range of technological tools and innovative solutions for the electricity grids of the future. The main idea behind this concept is to

address the problem of stability and reliability of power networks involving advanced domain knowledge and computational power. Smart Grids use digital technology to collect valuable insights about the state of transmission and distribution lines relying on a variety of sensors, sniffer, detectors, and utilizes cutting-edge computational power for data analysis and to further improve its managing ability. Besides, Smart Grid shows real advantages in ensuring more effective transmission of electricity and faster restoration in case of power breakdowns. It can also reduce operational, human resources related and management costs, improve security and allow for better interaction between customers and providers [12].

Power supply network of all kinds are known to be fault prone. Again, having to rely on them so heavily makes ensuring their stability and reliability the key challenge, not only for the future but for the present. Being able to locate power grid flaws and disturbances as soon as they happen, if non before, with the best possible accuracy is of the utmost importance. Improved timing and accuracy of proposed solutions, thanks to Smart Grid related technologies, in case of a fault helps to reduce the effect on other vital parameters of the power grid. Isolating the issue and limiting the potential risk of domino-like consequences prevents damages to not immediately hit components of the system. Wasn't it enough, this also drastically reduces costs.

Increasingly more complex power transmission set-up require more efficient and accurate methods of fault prediction and prevention. Through Smart Grid solutions, high-resolution data become available and lays the ground for introducing more sophisticated ideas for faults detection in the power grid.
It is indeed from this latter idea and from the concept of Smart Grid that this project takes inspiration.

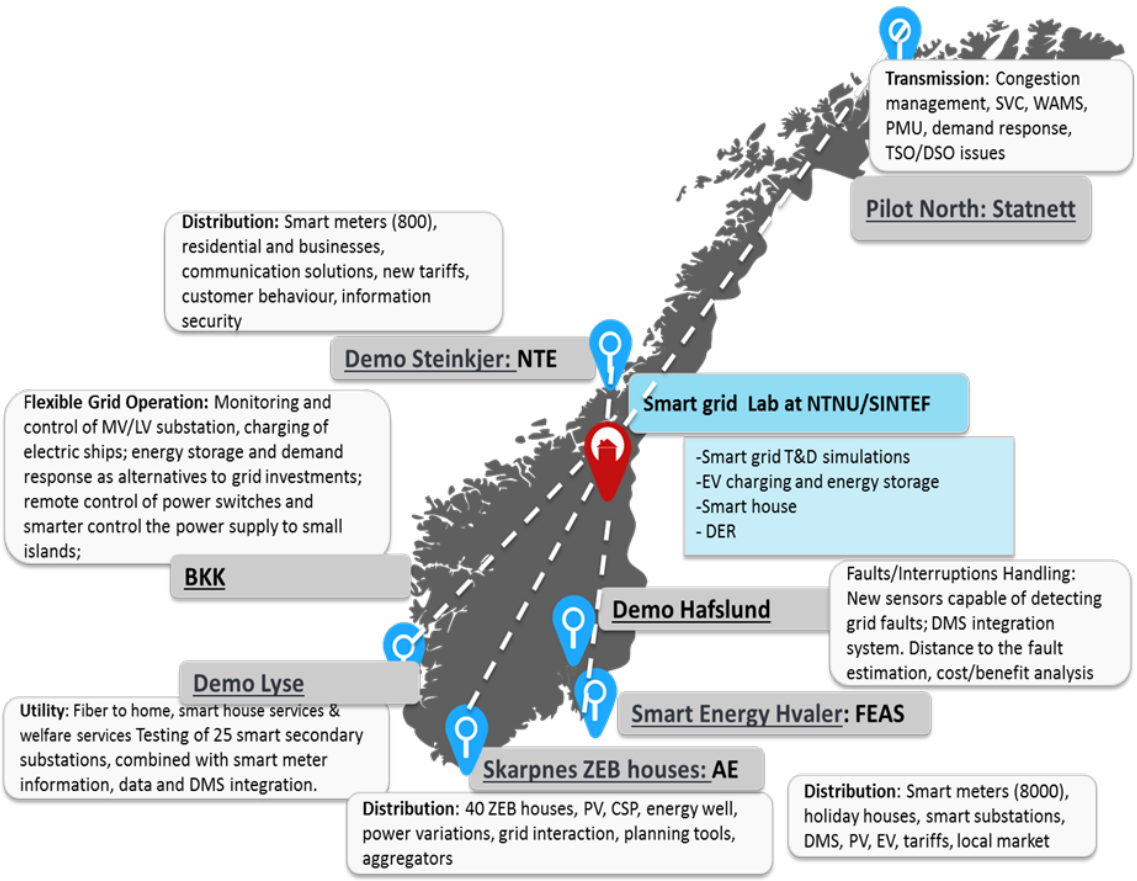

Figure 1.2 Demo Norway for Smart Grids

Norway has already established its position in the Smart Grid landscape. Through several scientific projects with the goal of supporting and coordinating a variety of international activities [9].

One of the most important achievements of The Norwegian Smartgrid Centre (NSGC) is Demo Norway for Smart Grids. The aim of the project is to carry out research projects and test the functioning of Smart Grid within both real life and laboratory environment. As a part of the initiative under NSGC, the National Smart Grid Laboratory (NSGL) gives its contribution to development work on eight power system sites located on national territory. Solutions regarding network communication between a power node and consumers, smart metering, micro-grid, and smart home concepts, among other things, are tested for constant evaluation and in order to gather important information to further develop the idea of Smart Grid in Norway [39].

1.1.3 Faults in power systems

After going through the whole description of what power systems are, how they work and how we can find inspiration within the idea of Smart Grid for developing smart algorithms to prevent faults, we need to describe faults themselves.
Faults are any shortcomings, deviations or malfunctioning that make a device unable to perform the function it is intended to in the power system [13]. The occurrence of a fault can have effect on the mechanical and physical components of the power grid as well as on the final product received by the customer. Within the power system, an unexpected failure could damage or even disable a part of the structure. On the customer hand, visible consequences can result to be a flickering light, damages to the property, especially considering electrical appliances, or hours of disservice.
The gravity of the consequences of faults or slowdowns in the power distribution highly depends on the type of occurred fault.

In this study the following types of fault are considered:

- **Earth fault**: The occurrence of an unwanted conductive path between a live conductor and the ground, e.g. through a faulty insulation, vegetation or conductive structures such as cranes, ladders [10].
- **Voltage dip**: A sudden and rapid reduction in effective voltage followed by a recovery. In a system with one supply voltage it is said that a voltage dip occurs when its value drops into the range of 5-90% of the agreed voltage level, with duration from 10 ms to 1 min. In case of more than one supply voltage, a dip occurs when at least one of the supply voltages falls below 90% of the agreed level [10, 47].
- **Rapid voltage change (RVC)**: A change in the effective voltage within ± 10% of the agreed voltage level, which occurs quicker than 0.5% of the agreed voltage level per second [47].
- **Voltage interruption**: Non-delivery of electrical energy to one or more end users, where all supply voltages are below 5% of the agreed voltage level. The interruptions are classified in long term interruptions when > 3 min and short interruptions when <= 3 min [47].

1.1.4 The influence of weather conditions on the power system

According to the annual reports provided by the governative organization Statnett, which is the organization that runs as an operator for the power grid in Norway, weather conditions are the most common triggering causes of faults in the power system in Norway.
Figure 1.3 provides insights on the statistics about reported operational disturbances caused by faults in the distribution networks. The detection and covering of disturbances is performed using a software approved in accordance with the use of Fault And Supply Interruption Information Tool (FASIT). FASIT is a system for the definition of guidelines for the standardized registration and reporting system of failures and interruptions in the power system.[47, 58]

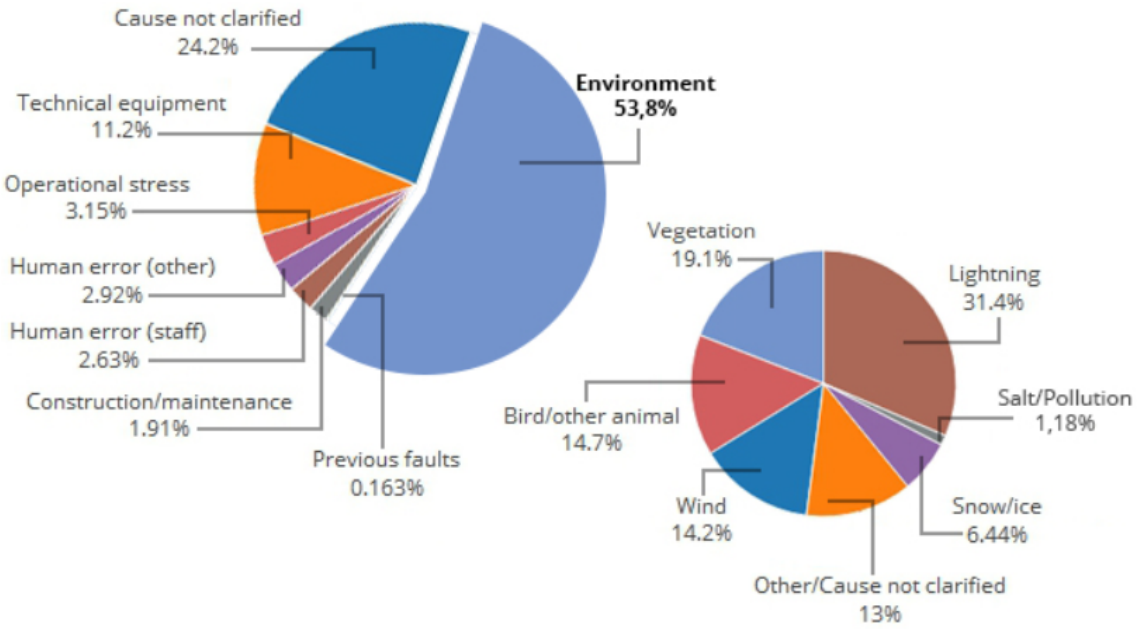

Figure 1.3 Fault causes

Environmental causes are certified to constitute 53.8% of registered faults. Faults due to human error, for example, are difficult to study, not to mention the 24.2% of faults that is not even properly classified. This hints that weather is one, if non the, of the most interesting aspects to analyze in order to understand what fault are caused by.

1.2 Related work

While scientific literature about fault detection is a boundless sea, it is possible to resume the most important studies regarding predicting fault on power systems. We look into the research related to fault prediction based on disturbances recorded by power system sensors and on studies on weather-based prediction. Eventually we discuss what has been done at SINTEF.

1.2.1 Weather-based prediction of power system faults

Most research related to the effects of weather on the occurrence of fault on the power grid are concerned about extreme weather conditions such as hurricanes [44, 23], earthquakes [54, 55] or storms [69, 2]. The influence of common weather conditions on the power grid remains not studied and not investigated well enough. There is, of course, a number of studies that goes through such a topic. From all of them we can obtain valuable indications about how to shape our work.

In their study [69], Zhu et al. develop a model for predicting storm-related faults on the power grid. The forecast they use as data is based on discretized storm states defined by wind speed and air temperature. The problem of the model developed by Zhu et al. is that it relies on real-time observational data at hourly intervals. As intuitive as it is, it is difficult to prevent faults and take appropriate action when relying on real time weather data. Besides, the real aim of their project to predict the number of outages per hour. The problem of outages position, though, is not addressed. Altough interesting, this model is limited in its practical applications.

In the work by Karin Alvehag and L. Soder [1], the model built predicts the system reliability but is again based on extreme weather conditions. These conditions are only expressed in terms of high winds and lightning. The advantage of their model is that, being built for a very specific geographical location, covers an area small for which is able to provide good predictive performances.

Kankala et al., S. Das, P. Kankanala and A. Pahwa [26] use a neural network model to predict the number of power outages per day. Here the model is trained only on the daily wind speed of gust and lightning data.

1.2.2 State of art at SINTEF

As previously mentioned, SINTEF is not new to this sort of research. The studies described so far have already been taken into account by K. Michalowska for her master thesis [39]. In her study, Michalowska focused on two main research aspects: the possibility to predict faults occurrences based on monitored power flow expressed in effective voltage and the possibility to predict faults based on weather conditions.
The first mentioned aspect's results were found to be not satisfying by the author and are not directly relevant for the purpose of this work. On the other hand, the second aspect represent a baseline for the current research.
Models trained on weather forecast obtained significantly better performance than the baseline models, which was a random classifiers.
Seasonal models were also built with faults labeled with daily resolution with the idea of addressing seasonal differences within weather time series.
The best performance was achieved for the prediction of all types of faults, the same seen in subsection 1.1.3, in autumn and winter and for the prediction of Rapid Voltage Change in spring and summer. The type of faults the occurrence of which seemed to be the most associated with weather conditions, and which was also the easiest to predict, was, indeed, RVC.

The model with the best performance obtained a F1-score value equal to 0.38, while the baseline was equal to 0.28. The generally low scores were due to the inherent properties of the weather observations, such as high overlap between the instances of the positive and the negative class, and the class imbalance, with days corresponding to a fault being the minority class. Even though seasonal models outperform a random classifier, their ability to predict faults was found in need of some improvement. One of the main findings of the study is that models based on the weather are not particularly effective due to the inherent properties of the weather data. Not only, in fact, fault occurrences depend on weather condition, but also on season of the year and type of fault itself.

1.3 Problem definition and research questions

At SINTEF the idea of Smart Grid is taken in high consideration and it has become an important research point of emphasis. After going through some of the most interesting studies that was possible to found in scientific literature, this work lines up with previous works with the aim of improving current results and address encountered issues.
Previous works stumbled upon several criticalities that are to be faced in this thesis in order to eventually improve model performances in predicting fault occurrences over the power grid. These criticalities can be resumed as follow:

- The will to make weather-based prediction of faults in advance, in order to be able to properly react, imposes to use forecast data instead of observational data. Forecast, though, always have a margin of error and it is hard to know how close they are to real weather.
- Faults data are registered and available around substations. Substations structure cover large areas, while faults information are only indicated at the central spatial coordinates of the corresponding grid. This means that if a fault occurred on a branch of a grid sub-level we still receive fault information labeled at the center of the sub-level itself. This entails the impossibility to know the exact position where a fault occurred.

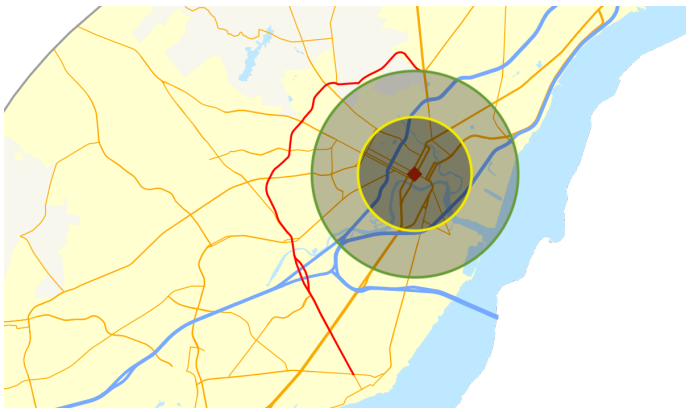

Figure 1.4 Possible fault registration area

As we shall see in the next chapter weather forecast values are expressed every 2.5 *km*. This makes the choice of which weather cell to use, and thus which values, completely arbitrary.

- Fault data have high class imbalance issues being the negative class the large majority of observed occurrences. As we shall see in section 4.5, class imbalance ratio, which is already very high for most algorithms, will only further increase when trying to augment fault data resolution from daily to hourly. Whatever the ratio, this is an issue that requires to be addressed by implementing ad hoc solutions or by training algorithms that are class-imbalance immune.
- Model performances still depend on season and period of the year. Weather data follow, of course, seasonal trends. Developing different models for different seasons seems to be a dispersive solution, as seasonality needs to be addressed in a more compact way.
- Models so far developed have tried to predict the day when an hypothetical fault occurred. Increasing fault data resolution would, as mentioned, inexorably raise class imbalance ratio but, on the other hand, create more observation that could positively reflect on model performances.

Taking the baton from all reviewed work and considering their achievements and criticalities, we finally define the two main research question that shape this work.

Research question 1:

How good weather forecasts are?

Weather forecast are the primary source of information we are going to use for this project. Having an understanding of their quality, their reliability and of their inner nature is nothing but fundamental.

Research question 2:

How can our understanding of weather data translate into new methodologies and help improve model performances?

Improvement of model performances passes by augmenting fault data resolution, handling class imbalance and addressing missing position of fault occurrences. How can our knowledge about weather data and weather forecasts improve and innovate the way we address such issues?

2

Weather forecast quality assessment

In this chapter we start focusing on how weather forecasts are produced, post-processed and evaluated. At first we review how weather prediction models produce forecasts and how final post-processed products take shape. We then move forward to resume best practices for forecast quality evaluation and to see how MET already performs an evaluation of its own forecast before performing our own.

2.1 Focus on weather forecast

In order to evaluate weather forecast it is of primary importance to understand how weather prediction models are produced, how they work and where they come from.
Weather models, known formally as Numerical Weather Prediction [4] are at the core of modern weather forecasts. Weather models are simulations of the future state of the atmosphere through time. Millions of observations are used as initial conditions of trillions of calculations, producing a picture of what the atmosphere might look like at some time in the future [4]. Supercomputers are used to do these calculations at affordable speeds, to enable simulations to cover the entire globe, and extend up to two weeks into the future [40].

There are two general types of weather models, global models and regional models. Global models produce forecast output for the whole globe, generally extending a week or two into the future. Because these models cover a wider area, and a longer timespan, they are generally run at a lower resolution, both spatially (fewer forecast points per given area) and temporally (fewer time points get a forecast). These models are generally fairly accurate in predicting large scale patterns, but all of them become less accurate through time. The ECMWF (European Centre for Medium-Range Weather Forecasts) is generally considered to be the most accurate global model, with the US's GFS (Global Forecast System) slightly behind [65]. Regional models on the other hand have much higher resolutions, but only cover some part of the globe, and only provide forecasts a couple days out in time. The advantage with these models is that their higher resolution lets them foresee features that the global models miss, most notably including thunderstorms. For "region" we intend continental areas such as Europe, Asia, America. Local models instead are more localized and refer to areas such as Germany, or Scandinavia.

2.1.1 AROME-MetCoOp: A Nordic Weather Prediction Model

Forecast used in this study respond to a local weather predictive model for the entire territory of Scandinavia. Since October 2013, a convective-scale weather prediction model has been used operationally to provide short-term forecasts covering large parts of the Nordic region. The model is operated by a cooperative effort, resulting into the Meteorological Cooperation on Operational Numerical Weather Prediction (MetCoOp) between the Norwegian Meteorological Institute and the Swedish Meteorological and Hydrological Institute. The core of the model is based on the convection-permitting Applications of Research to Operations at Mesoscale (AROME). AROME is a forecast model developed by Météo-France [43].
The AROME-MetCoOp model covers large parts of the Scandinavian countries with a horizontal resolution of 2.5 km. More than half of its domain though is over open water. The horizontal grid, with dimensions of 739×949 points is defined by a *Lambert projection* with the center at $63.5°N$ and $15°E$ [**?**]
The model operates with a 3 hour update cycle, where atmospheric and land surface variables are updated. At every main cycle (00:00, 06:00, 12:00, and 18:00 UTC) a 67-hours forecast is produced and output. For these main cycles the cutoff time, the waiting time after the

official analysis for observations, is 1:15h. For the intermediate cycles (03:00, 09:00, 15:00, and 21:00 UTC), a short 3-hours cycle is produced and used as a feedback for the following main cycle. The cutoff time for the intermediate cycles is 3:40h.[43]
The stochastic character and fast development of convective cells emphasizes the need to use an ensemble prediction system rather than deterministic forecasts. Indeed, the latter is in use as a 10-members AROME-MetCoOp ensemble model known as MetCoOp Ensemble Prediction System (MEPS) [5, 56].

2.1.2 Post-processed weather features

MEPS outputs are available through the MEPS archive, a repository of historical data. MET shares not only the entire ensemble model it produces along with the above mentioned institutes, but also raw forecasts and a variety of post-processed forecasts.
For this project post processed forecasts retrievable in the meps_mbr0_pp_2_5km format have been used [38].

A single forecast file produced my MEPS consists of 21 weather variables. Out of them, in this study we consider a limited but comprehensive list of variables [38, 28]:

- **Air temperature**: air temperature at 2m height, measured in Kelvin (K)
- **Total cloud cover**: also known as cloud area fraction, measures portion of the sky covered by clouds. It is expressed in as a percentage (%)
- **Cloud cover of high clouds**: measures portion of the sky covered by high clouds and it is expressed as a percentage (%).
- **Cloud cover of medium clouds**: measures portion of the sky covered by medium clouds and it is expressed as a percentage (%).
- **Cloud cover of low clouds**: measures portion of the sky covered by low clouds and it is expressed as a percentage (%).
- **Fog**: also known as fog area fraction, it is expressed as a percentage (%), between 0 and 1, and represents percentage of the area affected by fog.
- **Precipitation amount**: amount of precipitation registered during an hour, measured in $\frac{kg}{m^2}$.

- **Accumulated precipitation amount**: total accumulated amount of precipitation registered at a time, measured in $\frac{kg}{m^2}$.
- **Relative humidity**: relative humidity registered at a 2 meter height, expressed as a percentage (%) and ranging from 0 to 1.
- **Surface air pressure**: Air pressure at surface level, measured in *Pa*
- **Thunderstorm index**: also thunderstorm index combined, expresses a probability for lighting, thus ranging from 0 to 1.
- **Speed of gust**: also know as wind speed of gust, measures in $\frac{m}{s}$ maximum speed of gust during a certain hour.
- **Wind direction**: direction toward which the wind is directed.
- **Wind speed**: mean speed of wind during an hour, expressed in $\frac{m}{s}$.

This list of features is a selection of the most important variables to consider when trying to predict power grid faults using weather data. According to studies mentioned in section 1.2, these are the variables mostly related to faults, easier to process and interpret, carrier of most of fault-related information.

In next sections of chapter 2 and in chapter 3 we will mainly focus on two of these variables: *air temperature* and *wind speed of gust*. While *air temperature* is the easiest variable to understand and process, *wind speed of gust* is probably the most important available feature. We proceed with the assumption that it is reasonable to extend evaluations we make for these two variables to the entirety of the forecast.

2.2 Best practices for forecast quality evaluation

As intuitive as it sounds, evaluating forecasts requires comparing them with real weather. That means, value after value, to take a forecast value A, for a certain time in a certain point in space, and compare it with an observed value B that must be noted, obviously, at the same time and at the same point in space.
Intuitive though doesn't stand for superficial. The Royal Meteorological Society, a long-established British institution that promotes academic and public engagement in weather and climate science, has drafted a substantial report where it describes all best practices to

use to properly evaluate weather forecasts [36]. The report distinguish between descriptive forecast and quantitative forecast. As we won't handle descriptive forecast, we rely on what's proposed by the Royal Meteorological Society to assess quantitative forecasts error for continuous variables:

Statistic	**Acronym**	**Formula**	**Range**	**Optimal score**
Mean Error	ME	$\frac{1}{n}\sum_{i=1}^{n}(f_i - o_i)$	$-\infty$ to ∞	0
Mean Absolute Error	MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert f_i - o_i\rvert$	0 to ∞	0
Standard Deviation of Error	SDE	$\left(\frac{1}{n}\sum_{i=1}^{n}(f_i - o_i - ME)^2\right)^{\frac{1}{2}}$	0 to ∞	0
Root Mean Square Error	RMSE	$\left(\frac{1}{n}\sum_{i=1}^{n}(f_i - o_i)^2\right)^{\frac{1}{2}}$	0 to ∞	0
Correlation	COR	$\frac{\frac{1}{n}\sum_{i=1}^{n}\left(f_i-\bar{f}\right))(o_i-\bar{o})}{SD(f)SD(o)}$	-1 to 1	1

Where $SD(f) = \left(\frac{1}{n}\sum_{i=1}^{n}\left(f_i - \bar{f}\right)^2\right)^{\frac{1}{2}}$ and $SD(o) = \left(\frac{1}{n}\sum_{i=1}^{n}(o_i - \bar{o})^2\right)^{\frac{1}{2}}$

Table 2.1 Error measures for weather forecast quality assessment

The statistics proposed above are not different from those generally used to evaluate regression tasks in machine learning. They are the standard for bias estimation.
There would also be two more measures that the Royal Meteorological Society proposes in its report, such as the Mean Average Percentage Error (MAPE) and Matthews Correlation. Even though these two measures have some interesting features we are not going to discuss them nor to use them. This is because we prefer to stick with measures that are in the same units as the quantity being forecast.
Throughout the course of this work we mainly use MAE and RMSE. RMSE has the benefit of penalizing large errors more so can be more appropriate in some cases. From an interpretation standpoint, MAE is the preferred measure. RMSE, in fact, does not describe average error alone. On the other hand, one distinct advantage of RMSE over MAE is that RMSE avoids

the use of taking the absolute value, which is undesirable in many mathematical calculations [36].

That said no single measure is adequate to describe the quality of a set of forecasts. Each and every one of these measures have benefits compared to the others, so we eventually use them all in order to draw the best possible interpretation out of this evaluation.

2.2.1 MET's evaluation

MET, along with its partners, already conduces an evaluation of their own forecasts. As already explained, in fact, MEPS is an ensemble model composed by 10 members. All these members are constantly evaluated in order to try to improve model outputs and to make sure quality of forecasts is always under control. To check the goodness of models, output forecast values are compared with observational data. The methodologies used by MET do not differ from those proposed by the Royal Meteorological Society and described at the beginning of this section [24].

Last MET's related publication is a report where the Institute goes through all its models. What is most interesting for this very study is the check-up they execute for MEPS. Here they analyze every single produced weather variable comparing forecast value with the corresponding observed value, doing something very similar to what we are going to do in this work. The evaluation supervised by MET though is only the starting point for the analysis performed for this thesis. At first we will try to replicate what MET has done in terms of evaluation, computing overall error measures, before moving on to unedited and custom analysis. This has a reason, and at the same time gives us a chance to explain in which way this work doesn't just replicate what is done by MET. MET, in fact, calculates errors with the idea of improving their own models from a meteorological standpoint. On the contrary, in this context we act like external users that just use forecasts as finished products. We are interested in forecast quality and error as a measure of uncertainty of the inputs we use for our predictive models. To feed back to meteorological ensemble models is not the aim of this project and would require a dedicated work.

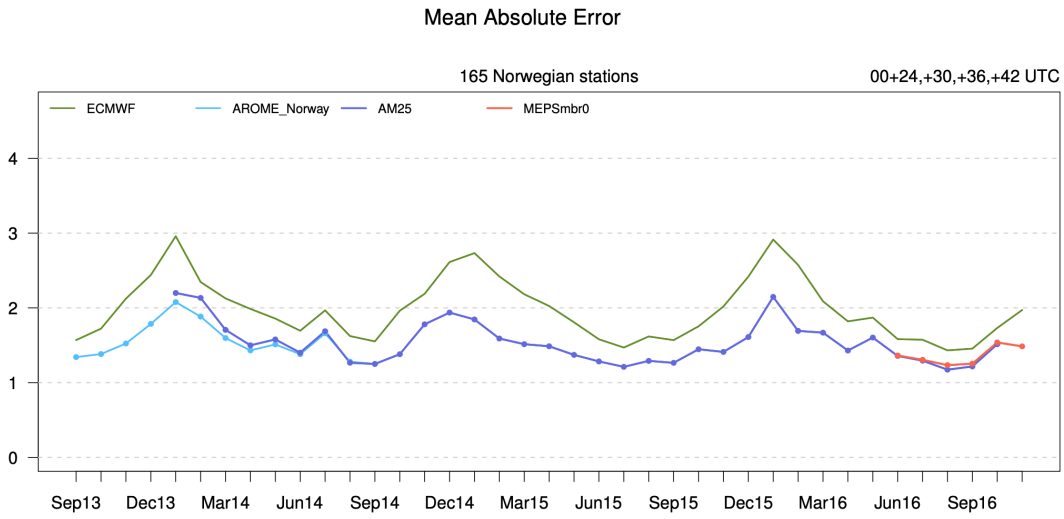

Figure 2.1 MET's evaluation thorugh 165 Norwegian stations

MET, variable after variable, computes the overall forecast error in a period ranging from September to November 2016 and divides weather stations used for observational controls into three categories: coastal stations, inner-land stations and Svalbard. These categories are then compared with the overall score to see whether they perform any different. MET doesn't provide any documentation though to label weather stations the same way.
In the next sections we focus on overall error measures too, but, still taking into account how MET has done its evaluation, we lay the ground to differentiate our analysis.

2.3 Observational data

Before seeing how to access and manipulate forecast files, it is relevant to dwell on observational data. An understanding of their nature is necessary and serves as a prerequisite to justify methodologies for access, manipulation and evaluation of forecast data that we are going to unwind in section 2.4 and 2.5.

2.3.1 Frost API

The source for observational data is the so called Frost API [45]. The Frost API provides free access to MET Norway's archive of historical observational weather and climate data. This data includes quality controlled daily, monthly, and yearly measurements of temperature,

precipitation, and wind data.
There are several different parts of the API, out of which it is possible to retrieve different types of data. The main pool of data consists of sources, available time series and observations. Querying for sources can be used to find available sources for a particular area or find meta information about a particular source. With sources we refer to weather stations, baloons, and any other tracking device that registers and provides weather data.
AvailableTimeSeries can be used to find out what types of weather elements, as types of observations, weather variables, are available for a particular station or time range. *Observations* can be used to retrieve, indeed, observations.
Frost API is very intuitive to use. It provides a graphic interface that helps to navigate API's endpoints and to simulate queries. In order to be able to perform queries, via graphic interface or making requests using Pythor or R, the user needs to sign-up and generate an id. Output data are in json format, always accompanied by metadata.

2.3.2 Observational data manipulation

To make a comparison of forecast data against observational data we need to compare these two types of data at the same exact location. While forecast data are available every 2.5 km^2, observations are only available where weather stations, et similia, are located. Therefore, we first extract observational data and then retrieve corresponding forecast data. All following operations are defined executed via Pyhton scripts.
For the sake of simplicity, we start to work around Air temperature. We send a request to Frost API asking for a list of weather stations that provide observational data from 1 January 2017 to 30 December 2017, for Air temperature at hourly resolution. Out of these request we receive a list of weather station ids. Using these ids we can send two new request to the API. The First one to retrieve location, expressed in latitude and longitude, and name of weather stations.

Weather station name	Latitude	Longitude	ID
FINSEVATN	60.5938	7.527	SN25830
GLOMFJORD	66.8102	13.9793	SN80700
HELLIGVÆR II	67.4048	13.8958	SN82410
KATTERAT	68.3995	17.966	SN84880
OSLO - BYGDØY	59.905	10.6828	SN18815
SPØRTEGGBU	61.5963	7.389	SN55425
...	...	...	...

Table 2.2 Weather stations location

Having spatial coordinates of stations we are going to use allows to filter those outside forecast grid boundaries and to plot them over the map of Norway and have a glimpse of their distribution all over the territory:

Boundaries

Latitude: *51.7390* *73.8591*

Longitude: *-14.3620* *48.6741*

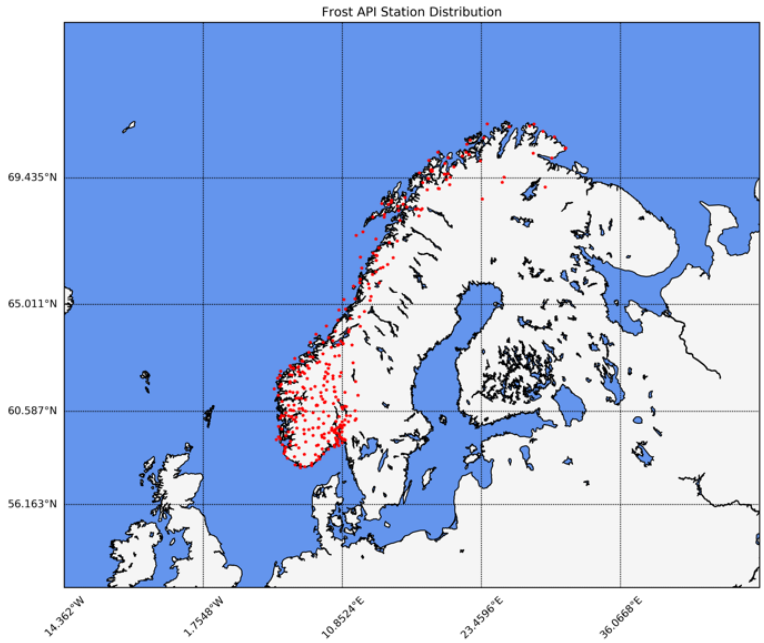

Figure 2.2 Distribution of weather stations across Norway

The second one is to get observational data for already specified parameters and for weather stations we have retrieved ids for. Frost API returns the required time series for every weather station. After some processing time series are concatenated into a Data Frame and joined with stations information. The obtained dataset is ready to be matched with forecast data.

A sample of air temperature observational data would look like the following:

Weather station name	**ID**	**Time**	**air_temperature_2m**
FINSEVATN	SN25830	2017-01-01 00:00:00	275,63
FINSEVATN	SN80700	2017-01-01 00:00:01	275.04
FINSEVATN	SN82410	2017-01-01 00:00:02	274,21
...	...	...	...
SPØRTEGGBU	SN55425	2017-01-01 00:00:00	273,44
SPØRTEGGBU	SN55425	2017-01-01 00:00:01	273,11
SPØRTEGGBU	SN55425	2017-01-01 00:00:02	272,85
...	...	...	...

Table 2.3 Sample of air temperature observational data

2.4 Forecast data

Forecast data are trickier to handle compared to observational data. They all need to be downloaded one by one from MEPS archive and stored. As mentioned before, out of all MEPS products, we work with post-processed multi-layer geo-referenced forecast files. Geo-referenced means that the internal coordinate system can be related to a ground system of geographic coordinates. The relevant coordinate transforms are typically stored within the image file (GeoPDF and GeoTIFF are examples), though there are many possible mechanisms for implementing geo-referencing. The most visible effect of georeferencing is that display software can show ground coordinates, such as latitude/longitude or UTM coordinates. In other words, georeferencing means to associate something, data in this case, with locations in physical space. The term is commonly used in the geographic information systems (GIS) field to describe the process of associating a physical map or raster image of a map with

spatial locations. Georeferencing can be applied to any kind of object or structure that can be related to a geographical location [64].
Geographic locations are most commonly represented using a coordinate reference system, which in turn can be related to a geodetic reference system, or projection, such as WGS-84.

In this section we discuss how to explore and gain information about, and from, geo-referenced data and how to plot them.
We proceed to access and manipulate such data through code and to retrieve the value closest to a single set of spatial coordinates in order to make the best possible comparison with observations.

2.4.1 Forecast data manipulation

MEPS post-processed files come in network Common Data Form (NetCDF). NetCDF is a file format for storing multidimensional scientific data, developed and maintained by the Unidata community. The very first need we have is to gain information about structure and content of such file. In GIS environment, a well known tool for gaining such insights is Panoply, developed and openly distributed by NASA. Not only Panoply comes handy to plot geo-referenced and other array files, but it also provides a graphic interface for reading meta data.
In our case, the latter feature is particularly helpful as Panoply allows to inspect weather variables independently and for each of them access information about array layers, projection and geodetic system, units. The main feature though remains plotting, as Panoply allows to display weather variables over all latitude/longitude points of forecast grid and over time.

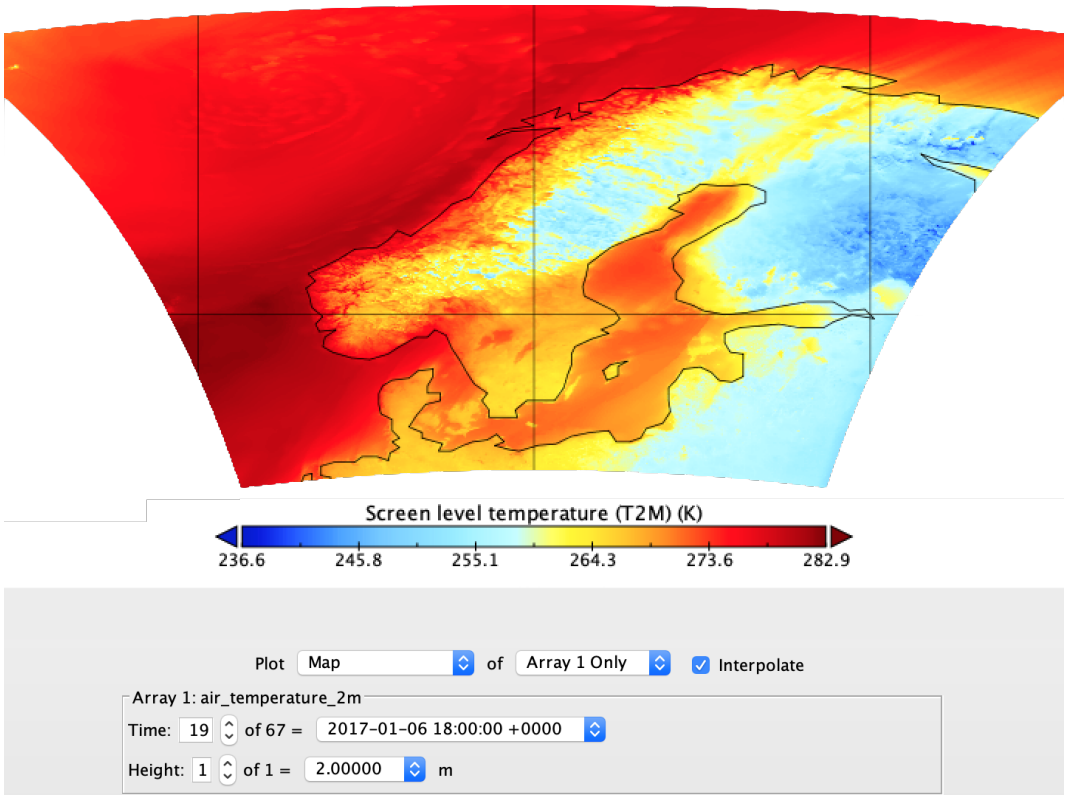

Figure 2.3 Typical Usage of Panoply

Panoply also gives the chance to view array in order to familiarize with how they are built over their dimensions and manipulate them as we shall see in the next subsection.

We already explained that georeferenced data internal coordinate system can be related to a ground system of geographic coordinates, generally expressed in couples of latitude/longitude values. In order to access data we need to slice over all array layers and pick value from the correct cell. This means, using as a direct example one of MEPS forecasts, that given the file *meps_mbr0_pp_2_5km_20171106T00Z.nc*, if we want to retrieve data for air temperature on 6 January 2017 at 18:00, in a pythonic way we need to do something like this:

forecast['variable_name][x, y, height, timestep]

We need to express name of the variable we want to access, height, even though the only permitted value is 2m, timestep which in our case is the hour of the day starting from midnight, and (X,Y) values. X and Y are coordinates point relative to the internal coordinate system and are related to geographic coordinates. Python libraries that handle georeferenced files,

such as netCDF4 and xarray, have built-in functions to provide access to multidimensional data by array indices, but we would often rather access data by geographic coordinates in order to get access to the closest available cell of the array. The developers at Unidata blog, a space where Unidata developers themselves use to keep the community updated, provides different valuable approaches to access netCDF, and similar, data based on coordinates instead of array indices [61]. The central problem is that we need to find indices X and Y such that the point (*Latitude[y, x]*, *Longitude[y, x]*) is close to the desired point (*lat0, lon0*). The naive way to do that would be to check squared distance for all grid pairs of Y and X values, one point at a time. This approach is both slow and wrong at times, because it suffers from flaws in the use of a flat measure of distance. In fact, this method treat the Earth as flat with a degree of longitude near the poles just as large as a degree of longitude on the equator. A more scientifically reliable method would be to use a better metric, such as the length of a tunnel through the Earth between two points as distance, which happens to be easy to compute by just using some trigonometry.

The last and best method is to use a KD-tree, a data structure specifically designed for quickly finding the closest point in a large set of points to a query point. Using a KD-tree is a two-step process: First you load the KD-tree with the set of points within which you want to search for a closest point, typically the grid points. How long this takes depends on the number of points as well as their dimensionality. The second step provides a query point and returns the closest point or closest n points in the KD-tree to the query point, where how "closest" is defined can be varied [61].

The KD-tree query is significantly faster than previous methods, this approach scales much better when we have one set, of points to search and lots of query points for which we want the closest point/s.

The KD-tree data structure can also be used more flexibly to provide fast queries for the N closest points to a specified query point, which is useful for interpolating values instead of just using the value of a variable at a single closest point, as we shall see in the next subsection.

2.4.2 Inverse distance weighting interpolation

Now we have a method to correctly extract, given a set of spatial coordinates, the right value from our georeferenced data. We know for a fact though that forecast grid is composed by 2.5 km^2 cells, each of them expressing a value at its center. Therefore, given coordinates of a weather station that produces observational data we can only retrieve the closest forecast cell, which is not necessarily at the same position of the weather station.

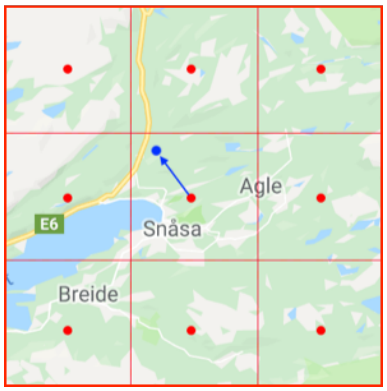

Figure 2.4 Forecast value retrieval by nearest neighbour

Figure 2.4 represent an hypothetical sample of the forecast grid, where red dots are forecast grid cell centers, the point from which we can extract weather variable values, while the blue dot is the possible position of a weather station. The arrow is just a graphic element that helps to link weather station position to the closest forecast grid cell. It is self evident that in this case we are comparing two values, with the aim of calculating an error, from two sources that do not originate data from the same position. This discrepancy can, at times, be minimal and possibly irrelevant, but for the sake of correctness and to avoid, as much as possible, miscomparisons we define a methodology in order to address this issue.
As hinted by Unidata developers in their blog post [61], data interpolation is a widely used technique within GIS community. Interpolation allows to, starting from values of known points, to estimate values of unknown points [3].

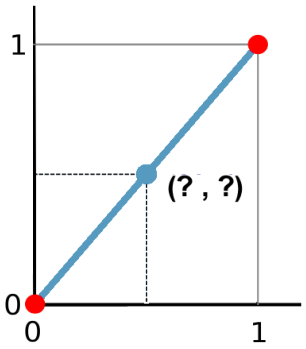

Figure 2.5 Example of linear interpolation

To review how interpolation works we use figure 2.5 as an example. To estimate the point between the two red dots, a dotted line is drawn to the x-axis and then to the y-axis. For linear interpolation the estimated of blue point is 0.5 and 0.5.
Interpolation in GIS works the same. Across the community, a common interpolation technique is Inverse Distance Weighting (IDW) interpolation. IDW interpolation includes multiple points, starting from the assumption that closer values are more related than further values with the point we want to estimate. Tobler's First Law of Geography, or spatial autocorrelation is the underlying assumption of Inverse Distance Weighting. IDW is a flexible technique, since it allows to set a radius of points to consider [16].

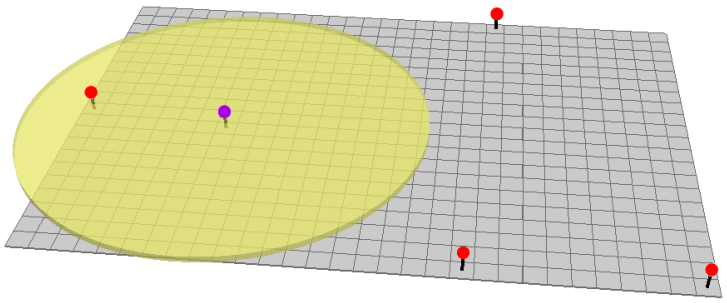

Figure 2.6 Example of interpolation radius of interest

IDW interpolation is defined as follow:

$$Z_p = \frac{\sum_{i=1}^{n} \left(\frac{z_i}{d_i^p} \right)}{\sum_{i=1}^{n} \left(\frac{1}{d_i^p} \right)}$$

Where d is the distance and p is power exponent usually taking 1 or 2 as value.

Starting from the latter, we assess the effects of such value in power variation.

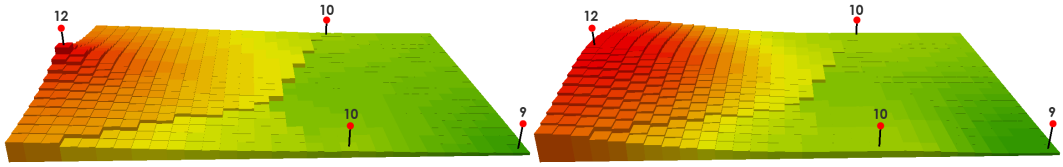

Figure 2.7 Effect of power in IDW formula

A power of 1 smooths out the interpolated surface while a power of 2 increases the overall influence it has from the known values. You can see how the peaks and values are more localized and are not averaged out as much as a power of 1 [16].

In our implementation of IDW, we do not consider d as linear distance, but we rather use a geodesic. The study of geodesics on an ellipsoid arose in connection with geodesy specifically with the solution of triangulation networks. A geodesic is the shortest path between two points on a curved surface, analogous to a straight line on a plane surface. As we have seen when calculating distances during the use previous use of KD-tree in the previous subsection, we should account for the fact that the Earth is not flat. For small distances Earth curvature might be ignored, but we prefer to run an implementation of IDW with the chance to return the most possible accurate value. Geodesic distances are brought into Python by Geopy, a popular library for geocoding web services. [17]

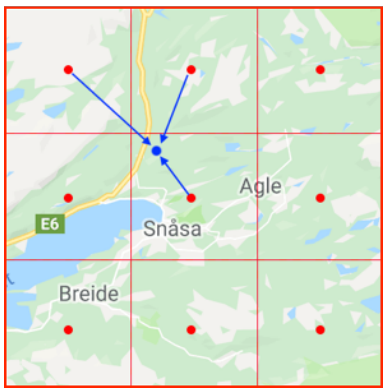

Figure 2.8 Forecast value retrieval by interpolating 3-nearest neighbours values

In this study, for the sake of computational scalability, instead of selection a wide radius of points we exploit KD-tree ability to return, given a set of geographic coordinates, not only the closest forecast grid cell but *N* cells, selecting the 3 nearest cells. We assume, in fact, that only closest cells can have a constructive impact when interpolating for weather data as we do in this case.

2.5 Computing overall error measures

In previous sections we've gone through our two main data sources and we've seen how to process observational data and forecast data. After querying for time series to Frost API, correctly slicing a interpolating forecast data, we are able to merge both time series in a thus looking dataset:

Weather station name	ID	Time	Observed_air_temperature_2m	Forecast_air_temperature_2m
FINSEVATN	SN25830	2017-01-01 00:00:00	275,63	275.12
FINSEVATN	SN80700	2017-01-01 00:00:01	275.04	274.78
FINSEVATN	SN82410	2017-01-01 00:00:02	274,21	274.56
...	...	...	...	...
SPØRTEGGBU	SN55425	2017-01-01 00:00:00	273,44	274.23
SPØRTEGGBU	SN55425	2017-01-01 00:00:01	273,11	273.98
SPØRTEGGBU	SN55425	2017-01-01 00:00:02	272,85	27.327
...	...	...	...	

Table 2.4 Sample of forecast and observational data merged

It is now possible to calculate quality scores and error measures, adhering to what enumerated in section 2.2.

Error measures are calculated, as anticipated, for forecast comprehended from 1 January 2017 to 30 December 2017, with hourly resolution, for *air temperature*. Out of 67 possible values available in a single forecast file we set a threshold deciding to utilize only first 24 hours of data from each file. This is based on the assumption that forecast quality degrades over time. From subsection 2.1.1 we know that forecast files are produced in blocks of 67 hours starting at 00:00, 6:00, 12:00 and 18:00 of every day of the year. Considering all 67 values from a file would expose forecast quality to useless reductions, while downloading and managing a file every 6 hours turns out to be very expensive in terms of storage and computational resources required.
The following table summarizes overall scores:

air_temperature_2m	
MAE	1.570
RMSE	2.173
STD	2.155
Corr	0.977

Table 2.5 Overall error measures scores for air temperature

wind_speed_of_gust	
MAE	2.102
RMSE	2.957
STD	2.957
Corr	0.807

Table 2.6 Overall error measures scores for wind speed of gust

For a task such ours there is no proper term of comparison. We are not measuring MEPS forecast quality to compare it with other producing system. The only assessment we can make is to declare whether we find forecast quality good enough as inputs for models we are to develop. Results shown here are coherent with those published by MET in their report [24], mentioned in subsection 2.2.1.
Other that good error scores results, what mostly makes our results encouraging in terms

of forecast reliability is the high score we obtain for correlation between observed data and forecast data for *air temperature*.
Latest information allow to prove at least two of the many assumptions we have been forced to make so far.

To say that accuracy of a forecast lowers over time starting from the moment it was generated is a common idea both among non-specialised audience and meteorologists [62, 18]. It can still be interesting to check for this concept in forecast data we are using. To achieve this we check whether, for every single file, forecast quality is higher during first 24 hours compared to hours from 24th to 48th, and again compared to last 19 hours.

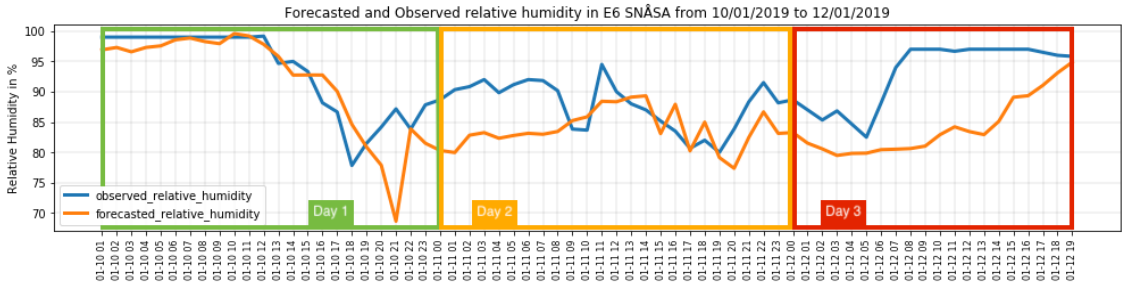

Figure 2.9 Example of forecast quality deterioration since time of its production

While figure 2.9 only provides an example, in 82% of forecast files appears that first 24 values, relative to the first 24 hours of prediction, when compared with observed value at that time score lower MAE than later hours. Obtained results shows how this disruption is consistent and confirms how using only 24 values for forecast is a reasonable compromise.

Another previously made assumption that is interesting to confirm was made in subsection 2.4.2 when implementing IDW interpolation. We decided to interpolate values from 3 cells in order to obtain a single estimated value to compare with observation made by a given weather station. We can obtained the dataset seen above in this section merging observations with forecast data retrieved both with and without interpolation and perform our evaluation.

	No interpolation	**3 cell interpolation**
MAE	1.850	1.803
RMSE	2.649	2.560
STD	2.629	2.542
Corr	0.886	0.892

Table 2.7 Measures of forecast quality after being retrieved with no interpolation or with 3-cell IDW interpolation

The table above reports error measures calculated for forecast with values obtained through the two different approaches.
There is no evidence in scientific literature of a proper range of values to interpolate for a given task. Forecast value estimated with a 3 cell IDW interpolation present best error scores against value selection per nearest neighbour, therefore we stick with our assumption.

Results discussed in this section only give us a general glimpse of forecast quality. We have only superficially dig to make sure the right assumption and decision were made.
In the next chapter we will go through a detailed study of what defines forecast error and what it is related to.

3

Understanding forecast error

In this chapter we are concerned with making sense of forecast error. We study how forecast error varies over time and space, how behaves when weather is harsher and how a physical description of the territory can help to discriminate where we can trust forecasts. Understanding what influences forecast error can be helpful not only to eventually improve ensemble forecast models themselves, but also to improve reliability of input data and model performances.

A source of inspiration certainly comes from the work of Linus Magnusson [35]. Magnusson has been working in the Diagnostics Team at the European Centre for Medium-Range Weather Forecasts (ECMWF), an independent intergovernmental organisation supported by 34 states, since 2011. One of his key tasks is to track down the causes of differences between the Centre's weather forecasts and observed outcomes. One of the approaches Magnusson follows during his researches is to start from mean forecast scores to establish in which areas of the globe or in which atmospheric conditions the errors are particularly large [34].

Following and expanding this approach, we provide an overview of what influences forecasts for the better or worse.

3.1 Forecast quality when forecasting harsh weather

For its own chaotic nature, weather predictability have inner limits. Besides, weather conditions themselves can influence level of predictability [48, 29].
What we can do to apply this theory to our work is precisely study the behaviour of forecast error at the varying of forecast value. Considering the high number of error data, the best option to correctly visualize this plausible trend is to use a plot respectful of data density, such as a hexagon plot. We conduct this analysis on the two elected reference weather variables, air temperature and speed of gust.

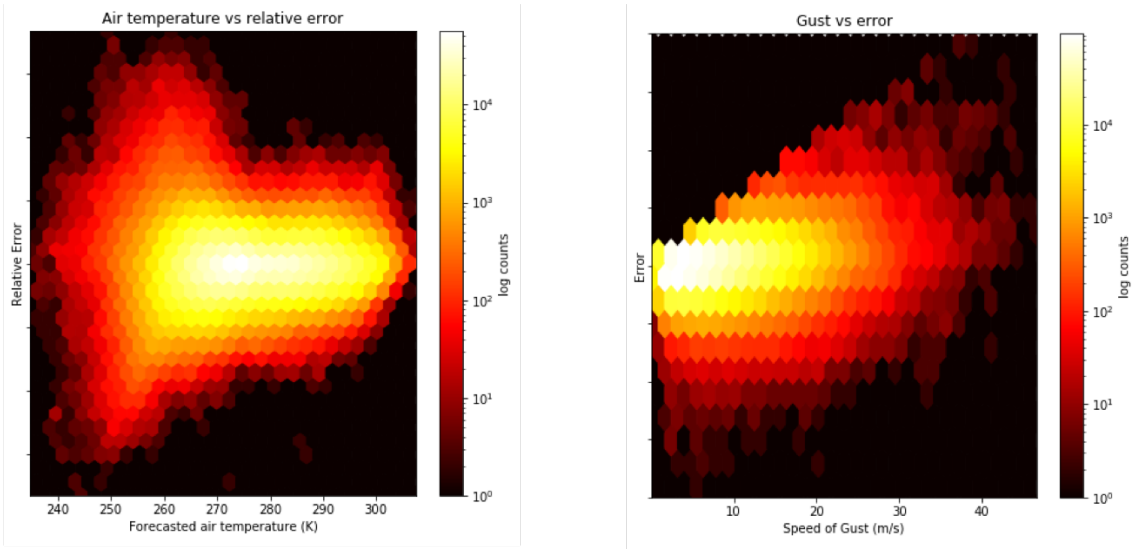

Figure 3.1 Air temperature and wind speed of gust forecast error against forecast value

Using log of counts we are able to zoom colored values and to see how the here plotted error distribution reflects the evaluation made in the previous chapter. On the left box, we can observe how variance of air temperature forecast error increases and registered error get higher as air temperature values decrease below 0°C (273.15 K). On the right box, we see how at higher forecast speed of gust correspond higher forecast errors. It is also interesting to note how when forecasting low speed of gust values the forecast tends to underestimate the actual weather. On the contrary, when forecasting high speed of gust forecasts bumps into higher errors. Regardless of under or over estimation of real weather, what we can infer out of these two plots is the awareness that forecast value matters when investigating what defines lower predictability in weather forecasts.

3.2 Forecast quality over time

The fact that weather variables are affected by seasonality is self-evident. Considering the theory that we validated in the previous section we would also expect forecast error to show the same trend. We have seen how harsh weather translates into higher forecast errors. We would therefore expect to observe higher variance and level of errors during winter months.

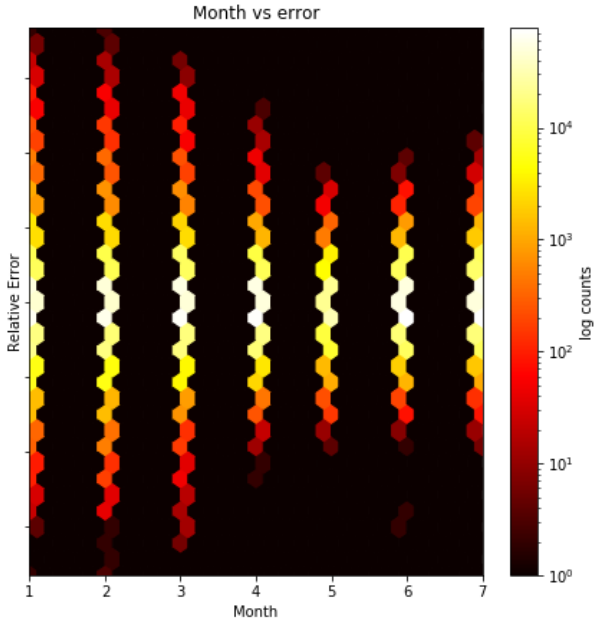

Figure 3.2 Air temperature forecast error against months of the year

Exploiting qualities of hexagon plot we can observe how during months such as May and June forecast error distribution tends to values closer to 0, while during the first three months of the year reaches highest values. Winter months are indeed expected to have harsher meteorological conditions, sparking higher levels of unpredictability.
It would also be interesting to study how forecast quality varies over hours of the day, but this notion would be conditioned by the period of the year and the resulting plot would be fairly unreadable.

3.3 Forecast quality over space

We need to remember that the forecast quality analysis we have done so far is based on the comparison between forecast data and observational data. These observations are retrieved from a limited number of working weather stations. Therefore, we are evaluating roughly 400 points belonging to the forecast grid out of more than 670000. Moreover, the evaluation performed in the previous chapter is based on averaged data coming from all considered points. As previous works suggest [35, 34], weather predictability varies in different areas of the globe.

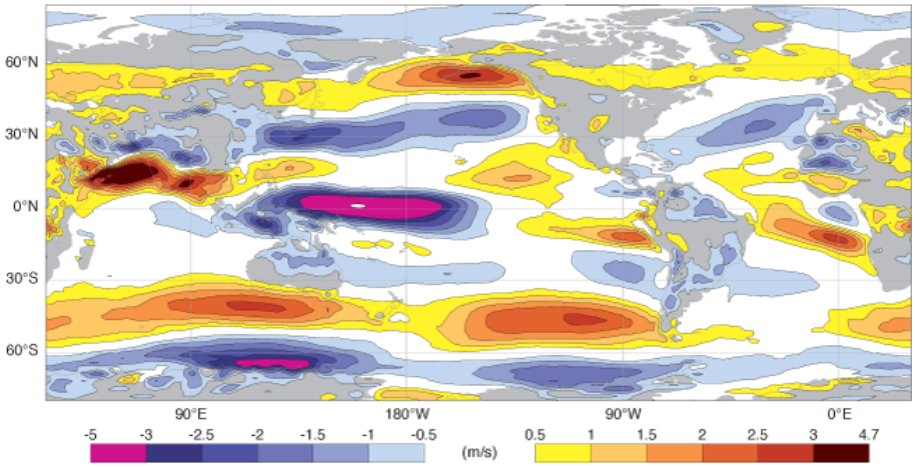

Figure 3.3 Forecast error at a larger scope

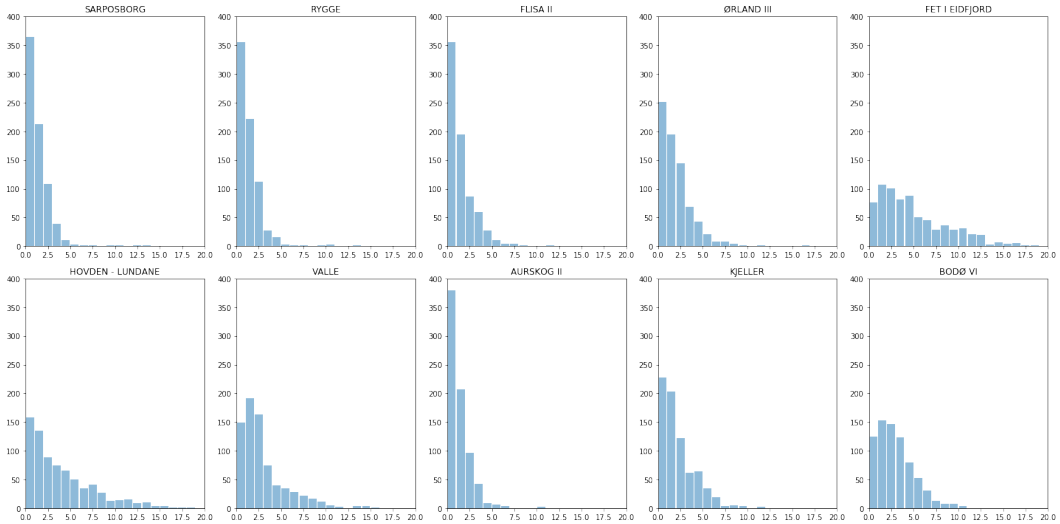

Figure 3.4 Forecast error distribution around 10 weather stations

This remains consistent even at different scopes.
In figure 3.4 we have histograms representing distribution of forecast MAE around ten randomly sampled weather stations. Error distribution clearly varies at different locations showing that there is high variability in weather predictability across Norway.

3.4 Terrain impact on weather forecasting

As well as harsh weather impacts forecast predictability, also terrain has an influence on it [46, 22]. This reflects what we have seen in the previous section when inspecting spatial variability of weather forecast error. Latitude, how far one is from the equator, greatly affects the climate and weather of an area. In geographic points close to the equator the climate is warmer, while moving north or south from the equator brings a cooler climate. Even altitude, the elevation of a point over the sea level, has a similar effect.
Proximity to water moderates the climate, while inland climates are harsher. This is why MET, in its forecast evaluation mentioned in subsection 2.2.1, divides weather stations into categories such as inland and coast. The further inland a spatial point is located, in fact, the drier the climate it experiences.

In this section we explore how the analysis of terrain can expand our understanding of forecast error.

3.4.1 Digital Elevation Models

In order to better discriminate how forecast quality diverges in different points in space and what is the impact of terrain in this extent, we need to describe how locations, pointed by power grid points, are geographically characterized.
A valuable source of information are Digital Elevation Models (DEMs). DEMs are an elevation model with a 10m resolution produced and released by the Norwegian Mapping Authority.

DEM need to be downloaded from the kartkatalogen of Geonorge[27]. They come in 254 *.tiff* files. These files are geo-referenced single layer raster. Every point, with the very high resolution of 10m, contains a value that represent the elevation, expressed in meters, of the point itself.

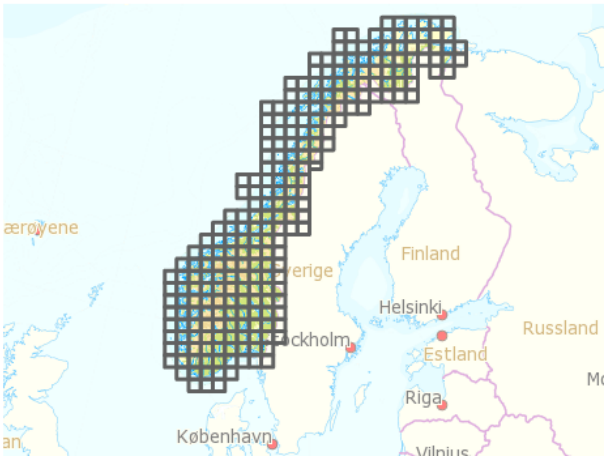

Figure 3.5 DEMs files segmentation

In order to start to visualize DEMs and obtain a single homogeneous file to work with we need to rely on tools such as QGIS. QGIS is a free and open-source cross-platform desktop GIS application that supports viewing, editing, and analysis of geospatial data. QGIS works as a wrapper for GDAL, a translator library for raster and vector geospatial data formats. Even though GDAL is available as a Python library, we exploit its functionalities through command shell for computational reasons. GDAL allows to merge all .tiff data and to convert the projection of the raster, returning a height map that we can utilize as a unique raster file in EPSG:4326, the same projection in use for forecast files.

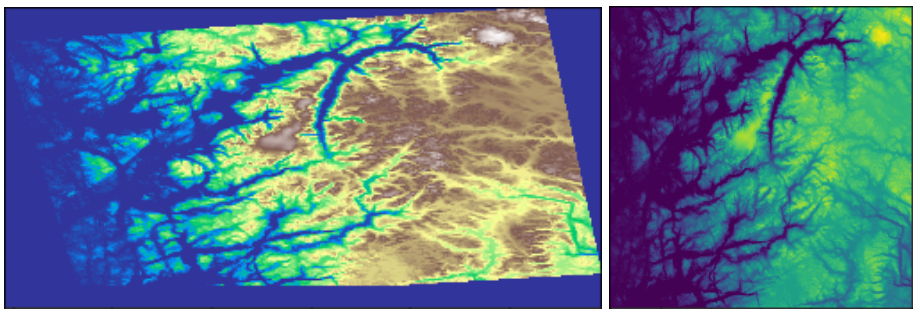

Figure 3.6 Single DEM file before and after projection convertion

The height map shows Norwegian territory coloured using a white to black gradient, where highest point of the country are represented by the darker areas of the image.

Figure 3.7 Height map of Norway

3.4.2 Terrain indexes

Even though DEMs only provide elevation data, altitude is not the only information we can extract out of them. Scientific literature commonly refers to terrain indexes as descriptors of terrain external structure. Main terrain indexes can be resumed as follow [67, 15]:

- **elevation**: altitude of a point in space, expressed in meters.
- **TPI**: Topographic Position Index, defined as the difference between a central pixel and the mean of its surrounding cells.
- **TRI** [37]: Terrain Ruggedness Index, defined as the mean difference between a central pixel and its surrounding cells
- **roughness** [37, 59]: the largest inter-cell difference of a central pixel and its surrounding cell.
- **slope**: maximum rate of change in value from a cell to its neighbors.
- **aspect**: azimuth that a slope is facing.

Ideally, it is also possible to think of a custom terrain indexes that possibly completes those above:

- **hm_rough**: the absolute value of the mean difference between a central value and its surrounding cells.

All these indexes provide a similar but complementary description of a portion of land. It is possible to calculate all above terrain indexes with the help of a GDAL extension, gdaldem, that brings in built-in functions respecting the indexes definition given above. Given a DEM file, *gdaldelm* gives back a raster where elevation data are substituted, for every point, by the desired index values [15].

As previously done with other sources of information, we plot error of air temperature forecast, for available points, against the so-calculated terrain indexes.

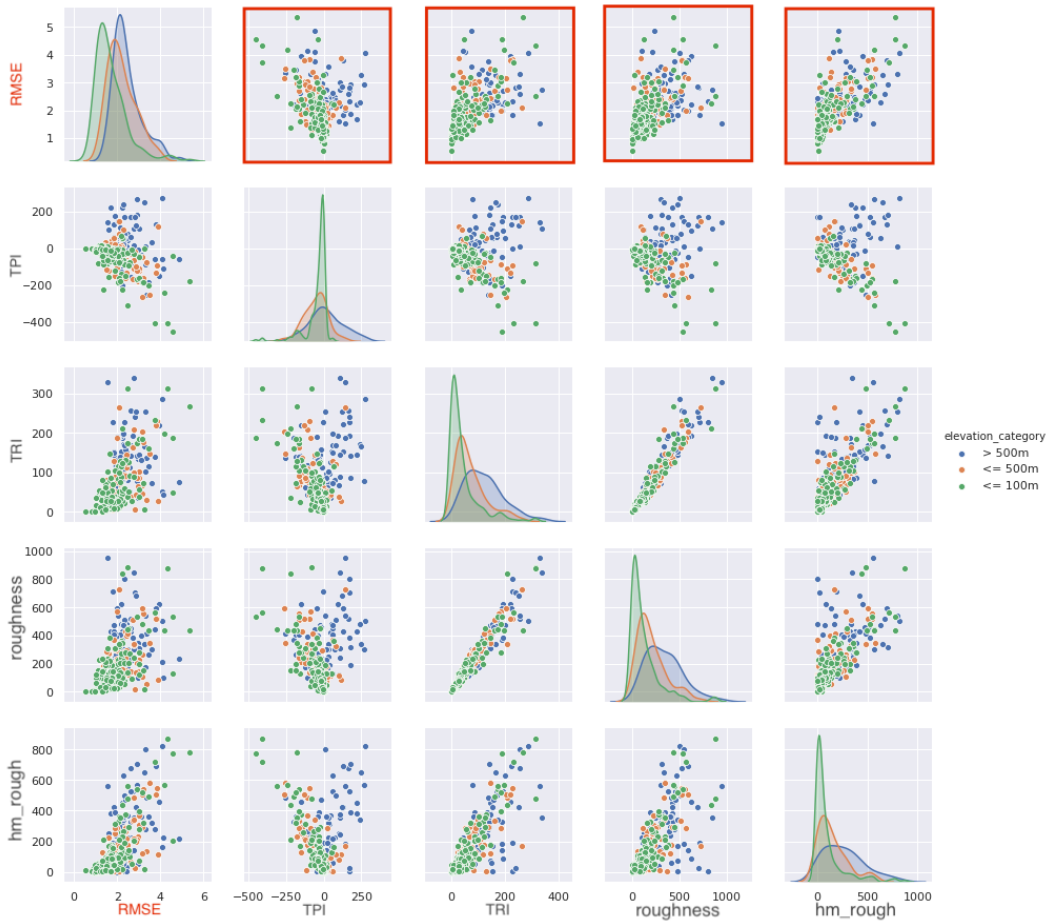

Figure 3.8 Terrain indexes and forecast error correlation plot

Figure 3.8 is a correlation plot for terrain indexes, except slope and aspect, against themselves and forecast error expressed as RMSE.

As foreseeable, terrain indexes are highly correlated among themselves, since they still provide similar information about a point in space. Coloring based on elevation categories slightly helps to make sense of correlation, but yet we do not have clear association between error and one of these indexes.

An explanation exist for the shortcomings of figure 3.8: terrain indexes suffer of an issue of considered area. As a consequence of DEMs high resolution, in this case, the area considered when calculating terrain indexes is pretty limited and possibly not extremely informative. Unfortunately, there is no scientific evidence of how large of a neighbourhood one should consider when computing terrain indexes for applications such as ours [66].

A possible solution that comes to mind when trying to define an appropriate region of interest for each of the above indexes is to model them altogether using a machine learning technique. Therefore, we develop a multivariate regression model having forecast error as a target and terrain indexes as features.

N.B. All machine learning models present in this study are introduced and described in chapter 4. Model details are not immediately relevant for what this section wants to convey. Therefore, we retain not to exceedingly interrupt the current content and to postpone description of mentioned models.

Still, the problem of the area to be considered to calculate terrain indexes remains. We decide to calculate terrain indexes utilizing a variety of ranges and areas of interest and to eventually evaluate which one to use. To compensate for limits of gdaldem, we implement two empirical solutions to address such issue and select the most informative area of calculation:

- **regridding**: gdaldem fournishes the possibility to regrid .tiff files and thus increase DEMs resolution, interpolating values to obtain cells that express averaged elevation for wider areas. By regridding it is possible, when calculating a terrain index for a given cell, to consider a wider area.
- **customize indexes range**: instead of taking only adjacent neighbors for every cell, we exploit more values through moving ranges of neighbourhood. Unlike regridding, with this technique DEMs resolution stays the same, while we choose to utilize a larger number of cells or different neighborhood shapes.

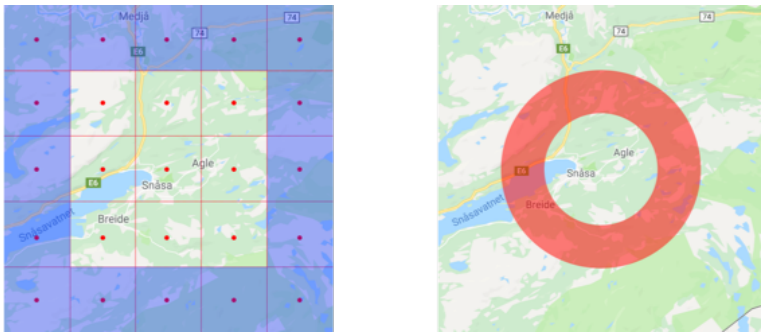

Figure 3.9 Possible neighbourhood shape and range

Figure 3.9 only provides an example of possible shape and range of windows to use. Again, there is no scientific evidence in literature to support the choice of this over that range, nor shape of neighbourhood. Since forecast grids cover an area of 2.5 km^2, we calculate terrain indexes covering areas starting from 100 m to 2.500 m using both techniques described above. Even though figure 3.8 hints that a linear association between forecast error and terrain indexes is not necessarily present, for all of them we compute Pearson's correlation against forecast error as a quick way to decide which index to include in the regression model. Eventually we decide to use TRI and TPI from 750 m cells generated by GDAL, a 250 m cell roughness index and what defined as hm_rough at the nearest neighbour.

Theoretically, we could build a model for every weather variable but, as for most of previous analysis, we rely on air temperature data. We prepare a dataset where for all available points of observation we provide TPI, TRI, elevation, roughness, slope, aspect.

avg_RMSE	weather_station	elevation	TPI	TRI	roughness	slope	aspect
1.986	SØLENDET	745.824	-20.219	30.839	87.018	3.225	183.620
2.197	MINNESUND	144.226	-45.653	15.990	45.172	2.515	64.795
1.746	ØSTRE TOTEN	261.622	14.858	38.770	114.326	1.871	30.191
1.979	ILSENG	182.511	-14.120	20.607	62.503	1.703	190.139
1.667	HAMAR II	140.205	-18.496	17.657	51.558	2.934	35.200
...	...	...	...	...	...	...	...

Table 3.1 Sample of dataset for regression with average RMSE as target variable

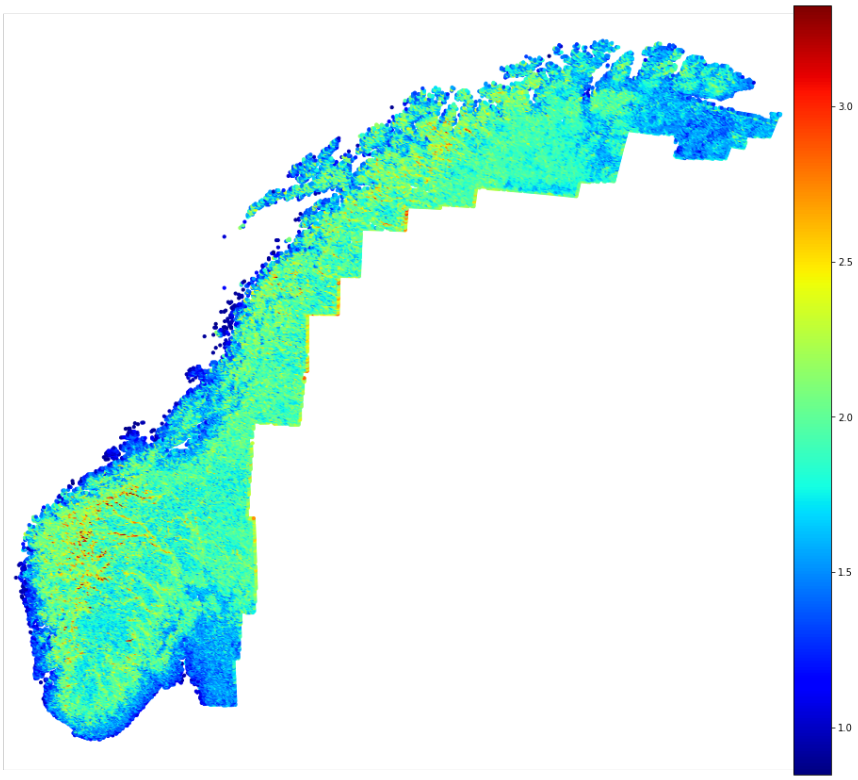

Figure 3.10 Estimate distribution of average forecast error over Norwegian territory

We set a regression task training on 336 geographic points corresponding to weather stations. We tune model parameters using a 10-fold cross validation build a model able to produce predictions for unseen points. We set an upper limit for unseen points terrain values at 120% of maximum observed training values and ask the model to return forecast error value for all points of the forecast grid. This model does not have the claim of actually predicting forecast error. A model bias of 0.32 RMSE must not hide that figure 3.10 shows the products of what is indeed a huge generalization. The model extends, in fact, what learned from roughly 336 points to more than 600000. Still, the output results in a gradient map of average forecast error for Norwegian territory. What is interesting to consider is how figure 3.10 resembles figure 3.7 and somehow recalls the height profile of Norway. It would have been possible to obtain a similar result just by interpolating error values, even utilizing IDW interpolation, of all power grid points starting from those available. However, with all its flaws, an estimation such this is much more informative, if non even more accurate, than just relying on distance proportional values.

3.4.3 Error by elevation category

The presence of mountains has, indeed, an influence on weather [8]. Consequently, as seen in previous cases, this can reflect on forecasts as well. Mountain areas are generally colder than surrounding land due to higher altitudes. Mountainous regions block the flow of air masses, which rise to pass over the higher terrain. The rising air is cooled, which causes condensation of water vapor, and precipitation[63].
We can use elevation data to divide weather stations, as well as all forecast grid points, into elevation categories.

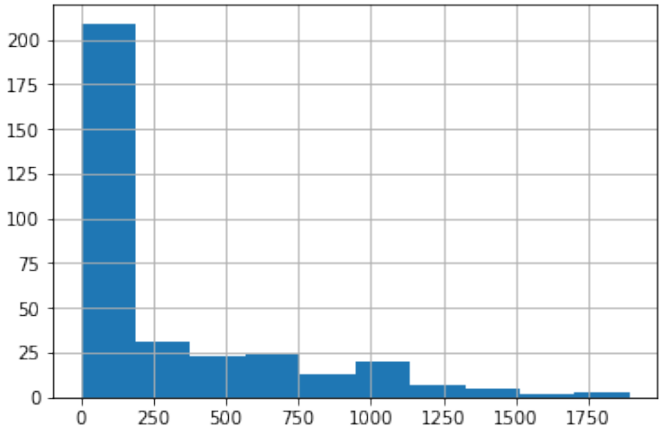

Figure 3.11 Elevation distribution in Norway

Elevation is the primary information that is possible to retrieve from DEMs. Building on elevation distribution histogram, figure x.x, we set three thresholds and build the following elevation categories:

- $<= 100m$
- $> 100m and <= 500m$
- $> 500m$

These threshold are arbitrary, but aim to represent plausible geographical areas that resemble how elevation peaks are distributed nationwide.
We replicate this categorization to again study are two reference variables. All working weather stations are divided into the categories above so that forecast quality can be plotted for all so-built categories.

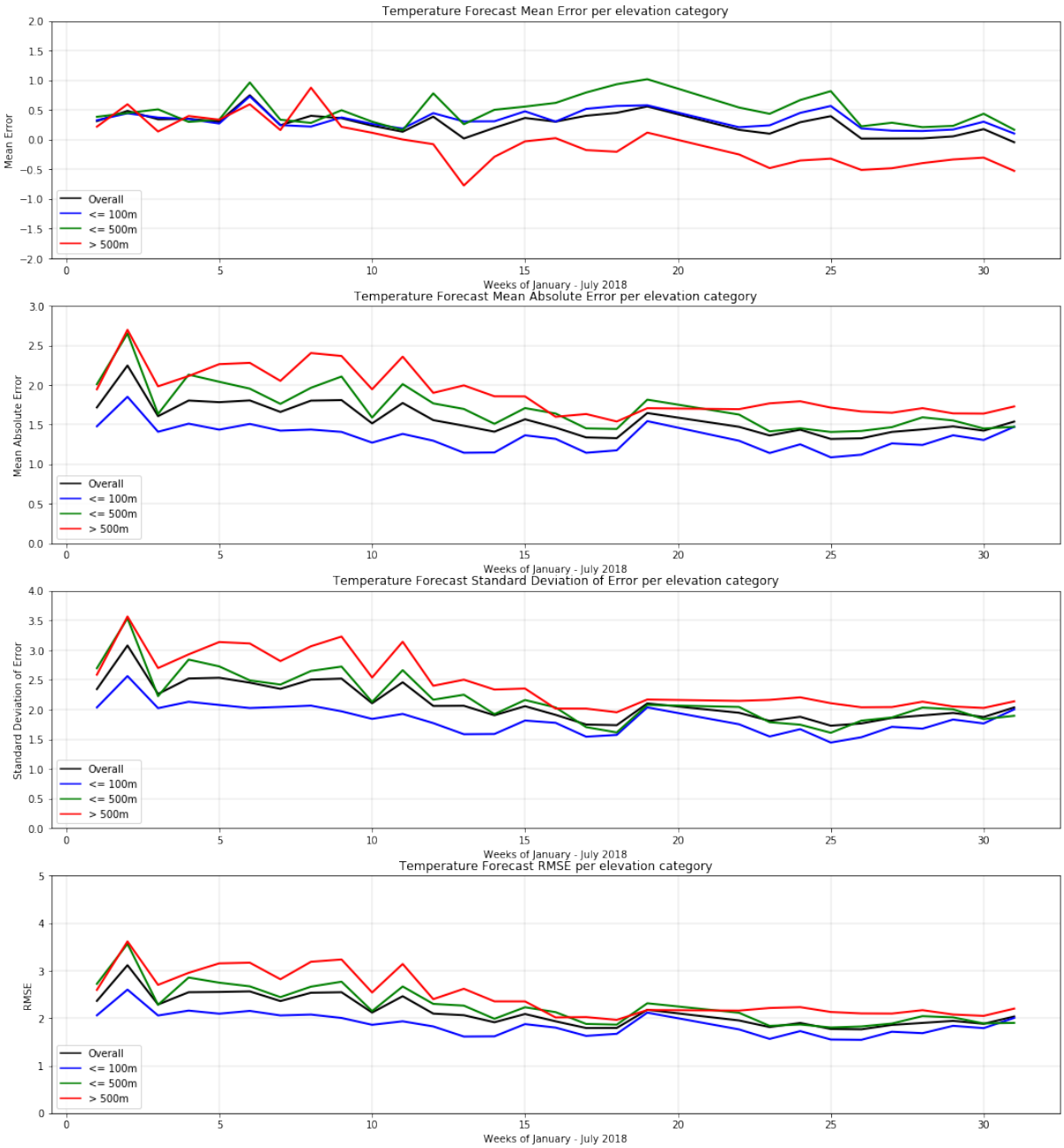

Figure 3.12 Error measures for air temperature divided into elevation category

Figure 3.12 shows clear differences in forecast error behaviour for points located at different elevation categories. The black line represents error measures for all considered points, while the other coloured lines show the trend of relative categories. Mean error of the three elevation categories stays very close to the overall trend during the first five weeks of the year. So far we have stressed that winter months bring harsher weather and higher errors with it. The first plot seems to contradict this idea, but it is to consider that plotted values are averaged and mean error can be both positive and negative, making the averaging of multiple points close to zero. Especially second and fourth plot confirm this interpretation, since highest error values are measured exactly during those firsts five weeks of the year.
Besides the apparent counterintuitivity of the first plot, we can observe a common trend: highest values of error are measured for points placed above 500 m from the level of the sea, while best results are observed for points of the forecast grid with corresponding ground lower than 100 m. Moreover, it can be interesting to observe how in points above 500 m, therefore with higher unpredictability, forecasts tend to underestimate the intensity of weather conditions.

A similar interpretation can be given for a specular analysis conducted on speed of gust. Quadratic or absolute error measure plots show how points in space corresponding to high ground present higher levels of unpredictability. Despite both figures having weekly resolution, for reasons do to readability of the plots, we can still appreciate how error behaves during time and how measurements computed for speed of gust return higher values of error. Speed of gust is the most important weather variable we have, but also one that is very hard to predict correctly.

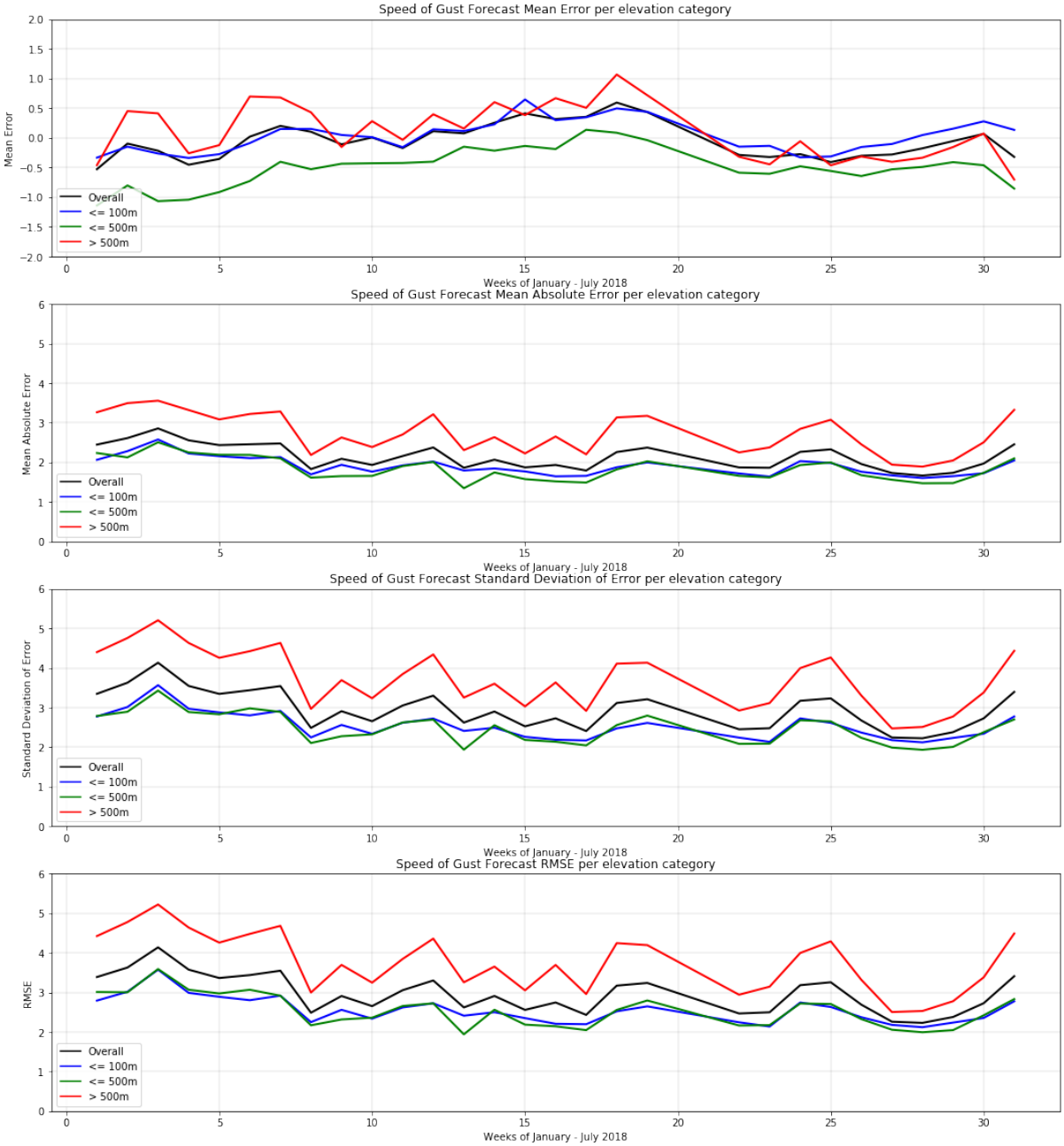

Figure 3.13 Error measures for wind speed of gust divided into elevation category

3.4.4 Error by slope category

Thresholding elevation is not the only significant way to build terrain categories. Also topography has a significant influence on weather: temperatures and precipitation are influenced by varied terrain. Temperatures generally decrease with height, thus the higher elevation regions tend to be cooler. However, in particular weather situations the temperatures can be cooler at the lower elevations. One way this can occur is when a cold front brings shallow cold dense air into the lower elevations. This is often called cold air damming. The cold air resists climbing the higher terrain since gravity holds and pushes the denser cold air toward lower elevations. Another way cooler temperatures can occur at lower elevations is when overnight cooling results in a pooling of cooler air into the lower elevation valleys [22].
Ground and landscapes can be described in more advanced ways. Andrew D. Weiss provides a landform analysis making use of the concept of topographic position [66]. Many physical and biological processes acting on the landscape are highly correlated with topographic position. There is good reason to think there might be associations between topographic position and weather predictability. Weiss thresholds TPI and slope values dividing spatial points into slope categories, providing a detailed description. The Topographic Position Index (TPI) is defined as in section 3.4.2. Along with TPI, slope is computed. For each cell, the Slope tool calculates the maximum rate of change in value from that cell to its neighbors. Basically, the maximum change in elevation over the distance between the cell and its eight neighbors identifies the steepest downhill descent from the cell. The output slope raster can be calculated in two types of units, degrees or percent rise. Here we stick with slope expressed in degrees. Both values can be produced utilizing GDAL. For slope calculation we stick with gdaldem functionalities. For TPI definition this time we use an annulus as range. The annulus has an inner cirle with 500m diameter and an outer cirle with 2500 diameter.
TPI, standardized, and slope are thresholded and divided into categories as below[66]:

- Ridge: TPI > 1
- Upper Slope: TPI <= 1 and TPI > 0.5
- Middle Slope: TPI > -0.5 and TPI < 0.5
- Flat Slope: TPI >= -0.5 and TPI <= 0.5 and slope <= 5
- Lower Slope: TPI >= -1 and TPI <0.5

We can divide forecast grid points into slope categories and see how are forecast error trends for each of them.

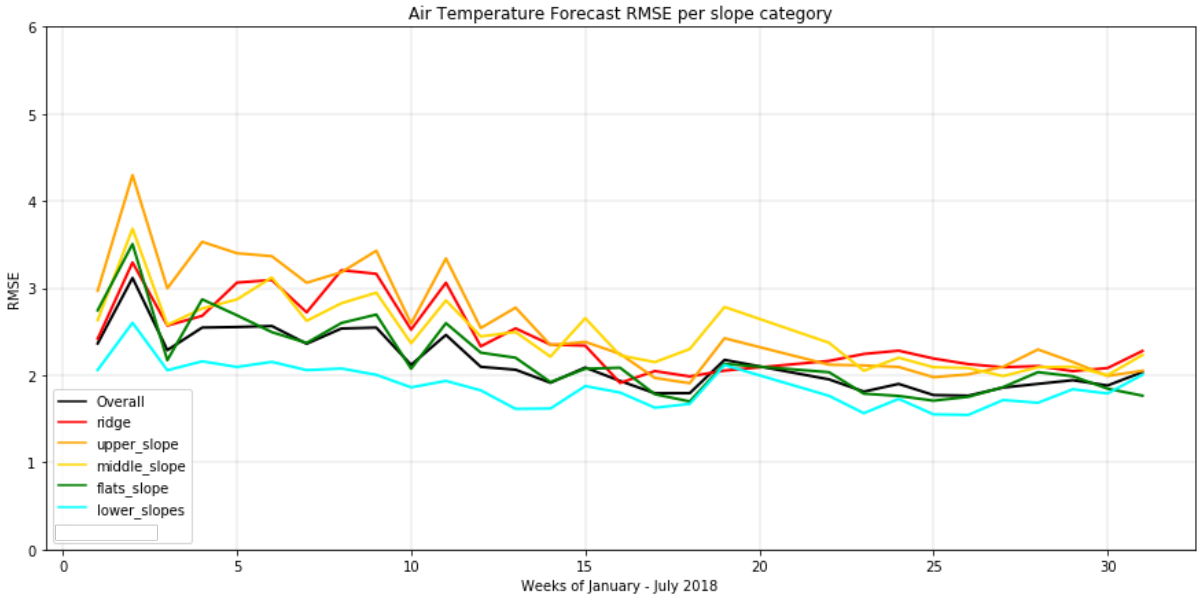

Figure 3.14 Air temperature forecast error over time divided into slope categories

As for plots of the previous section, the black line represents the overall error trend for points of the forecast grid we have been able to evaluate. Forecast error trend stays pretty similar among all slope categories. During first fifteen weeks of the year though it is possible to observe a divergence in error values range. As it was predictable, point placed on lower slopes register a better weather predictability, while upper slopes and ridges are categories associated with forecast grid points not so performative. Going toward spring, or the beginning of the summer, all error trends tend to converge toward the overall mean.
This analysis confirms what came out of the previous categorization based only on elevation, thus the idea of higher levels of weather unpredictability in mountainous or rough regions.

4

Reinterpreting fault prediction

After answering to research question 1 in chapter 2, in chapter 3 we have started to lay down the foundation to answer to research question 2. In this chapter we aim to improve what done from previous works in term of fault prediction and predictive models performance, possibly capitalising on what learned in chapter 3.
Before moving on we recall what is the status of previous works we want to improve and what are the major issues that need to be addressed: as mentioned in subsection 1.2.2, random forest was the machine learning model with best results in predicting faults in the power grid. However, multiple models need to be developed in order to give differentiated predictions for different seasons and fault types. All these models though have class imbalance issues, due to the sparseness of fault data, and relatively low performances, possibly due to the impossibility to know the exact position of a fault occurrence.
These are all problems we need to take into account when shaping are modeling approach during the course of this chapter.

We proceed analysing fault data available for this project. After an exploratory analysis we properly go through plausible machine learning techniques and evaluation metrics

in order to properly decide what to implement according to the state of the problem and to research necessities. Eventually, we prepare training data considering multiple possibilities and run a number of models to see how prediction performances fluctuates. Finally, we discuss obtained results.

4.1 Exploratory analysis of fault data

For this project, fault data come as a tabular file referring to a single node of the power grid in Norway. Details about the location can not be revealed because of non-disclosure policies. Available fault data has hourly resolution and refer to year 2017. For every timestep we have information about the registered fault. Faults are labeled as described in subsection 1.1.3, while within the current dataset are distributed as follow:

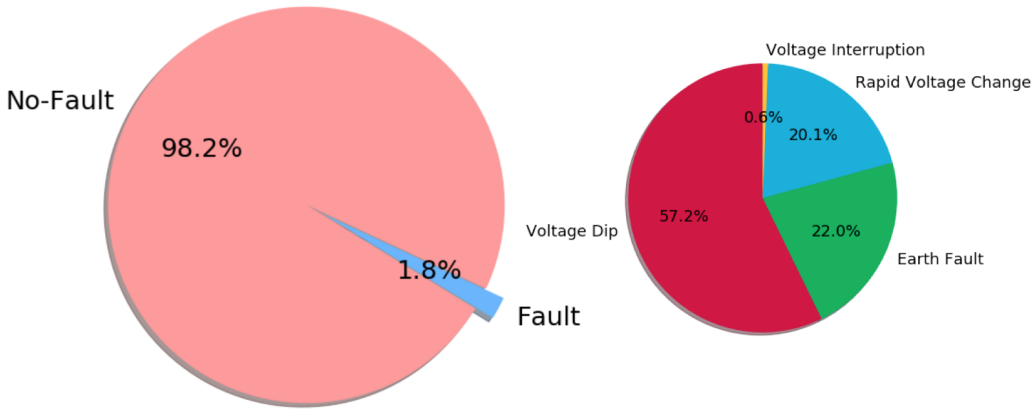

Figure 4.1 Class imbalance ratio and forecast causes

Voltage Dip: 91 *Rapid Voltage Change*: 32
Earth Fault: 35 *Voltage Interruption*: 1

As observable from figure 4.1, a class imbalance issue is not only present between non-fault class and fault classes, but also within the four fault types. It would be impossible to

correctly train a machine learning model only with one observation for voltage interruption, for example.

In subsection 1.1.4 we have seen how power grid faults may have common causes, commonly attributed to the impact of weather. In light of this and to address data sparseness, we turn to approach the problem as a binary classification task with the aim to recognize fault, the positive class, versus no-fault. Therefore, we convert all fault classes to a single positive class 'fault'. This does not eliminate the overall class imbalance problem. In fact, we still have 8591 observations for the negative class and only 159 for the negative class. At least though, we eliminate fault type sparsity and avoid the necessity to build separate models for different fault types.

This operation assesses one of the criticalities present in previous works, which is the need of developing separate models for different type of fault.

It is interesting to analyse how faults occurrences are widespread in different periods of the year.

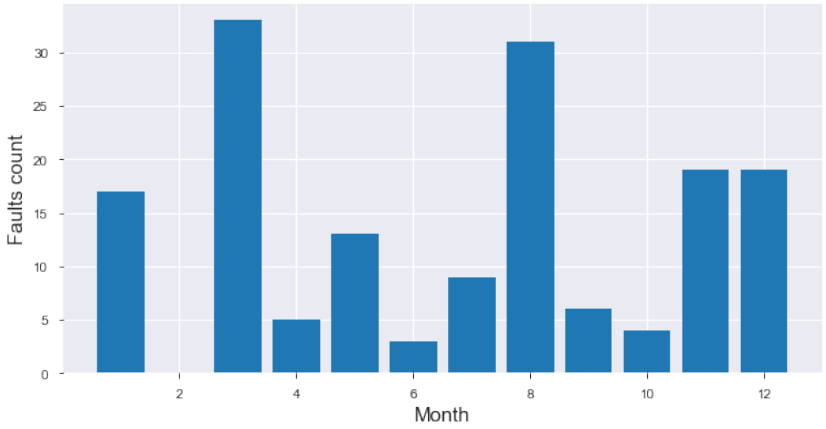

Figure 4.2 Fault count over months fo the year

That is another indication faults depend on weather conditions. For the sake of compactness and to reduce the level of dispersal brought by the usage of multiple models, instead to build a model for every season we prefer to incorporate season as a feature in order to account for seasonality.

This preliminary analysis serve as a start to shape our modelling approach.

4.2 Tree-based supervised machine learning algorithms

In this research we are not only concerned with model performances, but also with interpretability and in gaining an understanding of how different features impact predicting performances. For this and with the intention to maintain some continuity with previous work we favour the usage of instance-based models, picking from the family of tree-based learning algorithms. As we shall see, eXtreeme Gradient Boosting is the elected model for this project. Walking through the basics of tree-based algorithms helps us to justify our choice and to fully describe the algorithm before implementing it.

Tree-based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree-based methods are predictive models characterized by high accuracy, stability and ease interpretation. Unlike linear models, they map non-linear relationships quite well [52].

4.2.1 Decision tree

A decision tree is an acyclic graph that, in a classification task, can decide which class to attribute to an item. In each node of the graph the algorithm examines a specific feature j of the feature vector. If the value of the feature is below a specific threshold, the algorithms follows the left branch is followed, otherwise the right branch is followed. When the algorithm reaches the leaf node, finally attributes to the examined item the class where it belongs [6].
In a ID3 learning tree formulation the optimization criterion is the average log-likelihood:

$$\frac{1}{N}\sum_{i=1}^{N} y_i \ln f_{ID3}(\mathrm{X}_i) + (1-y_i)\ln\left(1-f_{ID3}(\mathrm{X}_i)\right)$$

where f_{ID3} is a decision tree [6].

The ID3 learning algorithm starts with a constant model that would give the same prediction for any input x f^S_{ID3}:

$$f^S_{ID3} = \frac{1}{S}\sum_{(x,y)\varepsilon S} y$$

The algorithm then searches through all features and all thresholds t, and split the set S into two subsets, $< t$ and $>= t$. Eventually, the algorithm pick the best values (j,t) and continue recursively on S_+ and S_-.
In ID3, the goodness of a split is estimated by using the criterion called entropy. Entropy is a measure of uncertainty about a random variable [6]. Entropy reaches its maximum when all values of the random variables are equally probable. The entropy of a set S is given by:

$$H(S) = -f_{ID3}^S \ln f_{ID3}^S - \left(1 - f_{ID3}^S\right) \ln \left(1 - f_{ID3}^S\right)$$

So, in ID3, at each step, at each leaf node, the algorithm finds a split that minimizes entropy or we stops at the current node [6, 52].
Such models can be improved by using techniques like backtracking during the search for the optimal decision tree at the cost of possibly taking longer to build a model.

Decision trees bring important advantages: they are easy to understand and have an high level of interpretability. They easily generate rules and have good qualities in reducing problem complexity. However, they can be relatively expensive in training time and may suffer from overfitting. Moreover, mistakes made at higher level of a tree are costly even at lower levels and overall they do not handle continuous variables very well [6].

4.2.2 Random Forest

A typical form under which we find decision trees is that of a random forest. Random forest are an ensemble implementation of multiple decision trees.
Ensemble learning is a learning paradigm that relies on a large number of weak learners and combines them to eventually obtain high-accuracy performances [31].
For its characteristics described above, decision trees are the most frequent type of used weak learner. To obtain the prediction for input x, the predictions of each weak model are combined using of a weighted voting that depends on the algorithm [6].
There are two ensemble learning paradigms: bagging and boosting. Bagging, the paradigm behind random forests, consists of creating many slightly different copies of the training data. Bagging techniques then apply the weak learner to each copy to obtain multiple weak models and then combine them [11]. After training, a random forest have produced B decision

trees. The prediction for a new example x, in classification, is obtained as the mode of B predictions:

$$y \leftarrow \hat{f}(x) \stackrel{\text{def}}{=} \frac{1}{B} \sum_{b=1}^{B} f_b(x)$$

Random forest uses a modified tree learning algorithm that inspects, at each split in the learning process, a random subset of the features. The reason for doing this is to avoid the correlation of the trees. Correlated predictors, indeed, cannot help in improving the accuracy of prediction [11, 6]. The most important hyperparameters to tune are the number of trees, B, and the size of the random subset of the features to consider at each split.
The reason why random forest is one of the most used algorithm is that by using multiple samples of the original dataset, we reduce the variance of the final model. By creating multiple random samples with replacement of our training set, we reduce the possible effect of overfitting [6].

4.2.3 Gradient Boosting

Another effective ensemble learning algorithm is gradient boosting. Gradient Boosting tries to create a strong learner from an ensemble of weak learners. The boosting problem can be visualized as an optimization problem by taking up a loss function and try to optimise it. The algorithm takes up a weak learner and at each step adds another weak learner to increase the performance and build a strong learner. This procedure reduces the loss of the loss function. The algorithm iteratively add new weak learners and computes the loss. The loss represents the error residuals, as the difference between actual value and predicted value. Using the loss value the predictions are updated to minimise the residuals.
During first iteration, the algorithm takes a weak model and tries to fit the complete dataset. The loss function tries to reduce these error residuals by adding more weak learners. The new weak learners are added to concentrate on the areas where the existing learners are performing poorly.
After multiple iterations the algorithm starts to better fit the data. This process is iteratively carried out until the residuals are zero. It can take a high number of iterations before the model properly fits the data [21].
Gradient boosting is one of the most powerful machines learning techniques. Not just because

it creates very accurate models, but also because it is capable of handling huge datasets with millions of examples and features. It usually outperforms random forest in accuracy but, because of its sequential nature, can be significantly slower in training [6].

4.2.4 eXtreeme Gradient Boosting

A popular machine learning algorithm, belonging to the learning trees family and exploiting boosting functionalities, is eXtreme Gradient Boosting. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XGBoost algorithm was developed as a research project by Tianqi Chen and Carlos Guestrin [41].
We have seen how gradient boosting is a special case of boosting where errors are minimized by gradient descent algorithm. XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods that apply the principle of boosting weak learners using the gradient descent architecture. However, XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements.

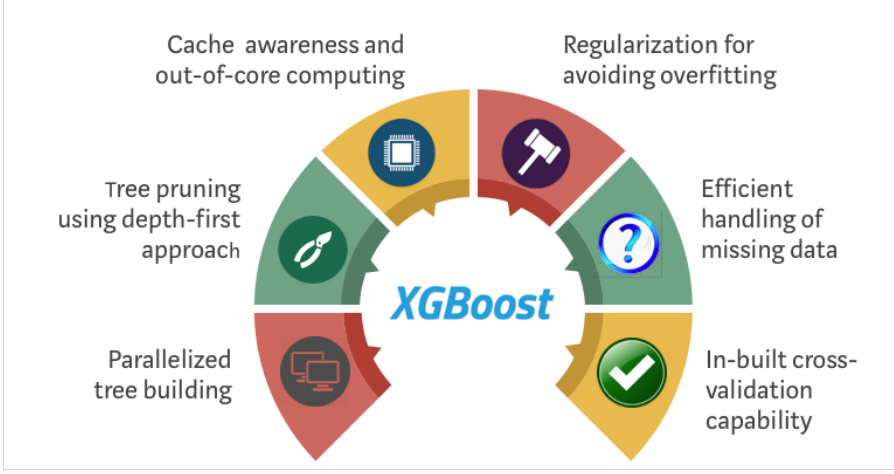

Figure 4.3 Main XGB features

System optimization in XGB is distributed in 3 points [41, 60]:

- **Parallelization**: XGBoost approaches the process of sequential tree building using parallelized implementation. This parallelization is possible thanks to the interchangeable nature of loops used for building base learners. Nesting of loops limits parallelization. Therefore, to improve run time, the order of loops is interchanged using initializa-

tion through a global scan of all instances and sorting using parallel threads. This switch improves algorithmic performance by offsetting any parallelization overheads in computation.

- **Tree Pruning**: The stopping criterion for tree splitting in GBM framework is greedy. It depends on the negative loss criterion at the point of split. XGBoost uses *max depth* parameter as specified instead of criterion first, and starts pruning trees backward. Such an approach improves computational performance significantly.
- Hardware Optimization: This algorithm has been designed to make efficient use of hardware resources. This is accomplished by cache awareness by allocating internal buffers in each thread to store gradient statistics. Further enhancements such as *out of core* computing optimize available disk space while handling big data-frames that do not fit into memory [41, 7].

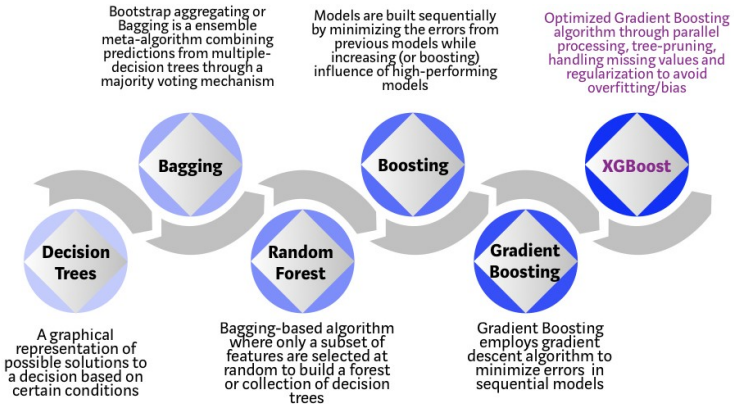

Figure 4.4 Tree-based learners evolution

In an experiment with the aim of testing XGB performances inits *scikit-learn*'s implementation, the *Make Classification* data package was used to create a random sample of 1 million data points with 20 features. When tested in the same way, several algorithms, such as Logistic Regression, Random Forest, standard Gradient Boosting, and XGBoost, offer different performances.

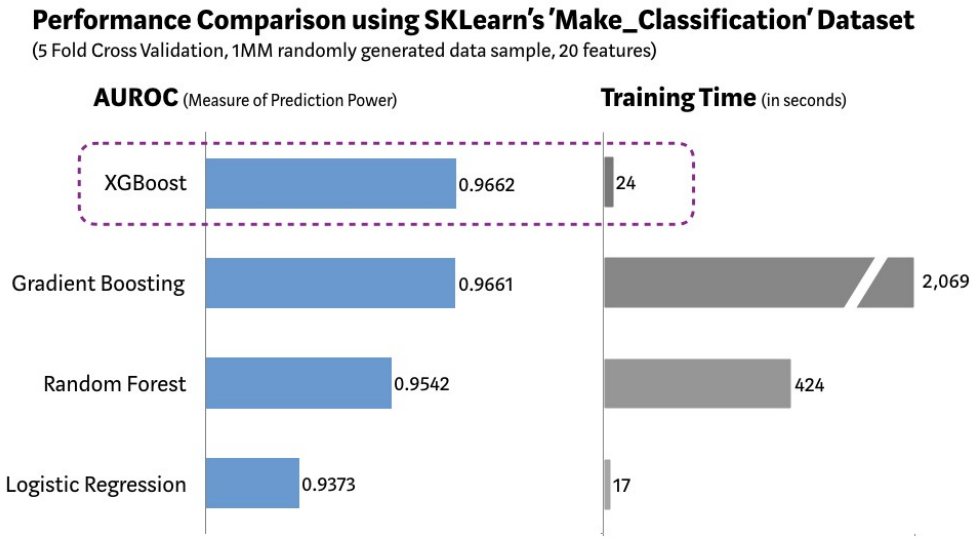

Figure 4.5 XGB performances compared to other algorithms

In its native Python implementation, XGB offer the possibility to tune seven main hyperparameters [68]:

- **eta**: or learning rate, is the step size shrinkage used in update to prevents overfitting. After each boosting step eta shrinks the feature weights to make the boosting process more conservative.
  range: [0,1] , default: 0.3
- **gamma**: the minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm.
  range: [0,∞] , default: 0
- **max depth**: the maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. XGBoost memory consume increases when training deep trees.
  range: [0,∞] , default: 6
- **min child weight**: the minimum sum of instance weight needed in a child. The larger min child weight is, the more conservative the algorithm.
  range: [0,∞] , default: 1
- **max delta step**: the maximum delta step the model allows each leaf output to be. If the value is set to 0, it means there is no constraint. If it is set to a positive value, it can

help making the update step more conservative.
range: [0,∞] , default: 0

- **subsample**: the subsample ratio of the training instances. Useful to prevent overfitting. Subsampling occurs once in every boosting iteration.
range: [0,1] , default: 1
- **colsample bytree**: the subsample ratio of columns when constructing each tree.
range: [0,1] , default: 1

4.3 Model performance evaluation

The most widely used metrics and tools to assess the classification model are:

- **confusion matrix**: The confusion matrix is a table that summarizes how successful the classification model is. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label. In a binary classification problem, such as ours, there are two columns per axis. [6, 25]

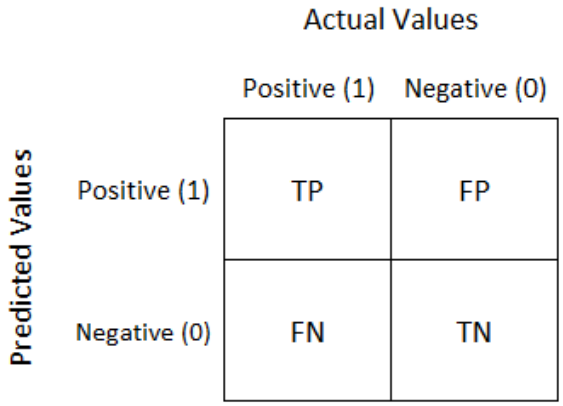

Figure 4.6 Binary confusion matrix

Positive class correctly predicted generates a True Positive, while negative class correctly predicted generates a True Negative. On the contrary, a positive class wrongly predicted generates a False Negative, while a negative class wrongly predicted generates a False Positive.

  - **True Positives (TP)** are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.
  - **True Negatives (TN)** are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.
  - **False Positives (FP)** occur when actual class is no and predicted class is yes.
  - **False Negatives (FN)** occur when actual class is yes but predicted class in no.

  Once understood these four parameters then we can calculate Accuracy, Precision, Recall and F1 score. [25]

- **accuracy**: Accuracy is the most intuitive performance measure. It simply is a ratio of correctly predicted observations divided by the total number of observations. Accuracy is indeed a great measure but only works well for symmetric datasets where values of false positive and false negatives balanced enough. When we have class-imbalance issues it is better to look at other measures to evaluate the performance of your model.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

- **precision**: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate.

$$\text{Precision} = \frac{TP}{TP+FP}$$

- **recall**: Recall, or Sensitivity, is the ratio of correctly predicted positive observations to the all observations in actual class.

$$\text{Recall} = \frac{TP}{TP+FN}$$

- **f1-score**: F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. It is not as intuitive and easy to understand as accuracy. However, F1 is generally more useful than accuracy, especially with unbalanced distributions. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall [30].

$$\text{F1-score} = \frac{2*(\text{Precision}*\text{Recall})}{\text{Precision}+\text{Recall}}$$

- **under the ROC curve**: the ROC curve is a commonly used method to assess the performance of classification models. ROC curves use a combination of the true positive rate and of the false positive rate to build a summary picture of the classification performance [30].

  The true positive rate and the false positive rate are respectively defined as follow:

  $$\text{TPR} = \frac{TP}{TP+FN} \text{ and } \text{FPR} = \frac{FP}{FP+TN}$$

  ROC curves can only be used to assess classifiers that return a probability, or confidence, score of a prediction. To draw a ROC curve, we first discretize the range of the confidence score [0, 1], to later discretize it like this: [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]. Then, we use each discrete value as the prediction threshold and predict the labels of examples in our dataset using our model and this threshold. The higher the area under the ROC curve (AUC), the better the classifier. A perfect classifier would have an AUC of 1.

  ROC curves are widely used because they are relatively simple to understand. They capture more aspects of a classification performance and allow to visualizeand compare performances of different models.

- **precision-recall curve**: we mentioned in section 1.3 that power grid fault prediction tasks generally have a class-imbalance problem. A particular, less common, evaluation metric able to assess such problem does exist. The Precision-Recall curve, in fact, is more informative than the area under the ROC curve when evaluating binary classifiers on imbalanced datasets.[53]

  The precision-recall curve shows the tradeoff between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results. In a system with high recall but low precision most of its predicted labels are incorrect when compared to the training labels. An ideal system with high precision and high recall would return many results, with all results labeled correctly [53].

  For the definition of precision given above, we note that precision may not decrease with recall. This shows that lowering the threshold of a classifier may increase the denominator, by increasing the number of results returned.

Recall again, as defined above, does not depend on the classifier threshold. This means that lowering the classifier threshold may increase recall, by increasing the number of true positive results. It is also possible that lowering the threshold may leave recall unchanged, while the precision fluctuates.
The relationship between recall and precision can be observed in the stair-step area of the plot, at the edges of these steps a small change in the threshold considerably reduces precision with only a minor gain in recall.

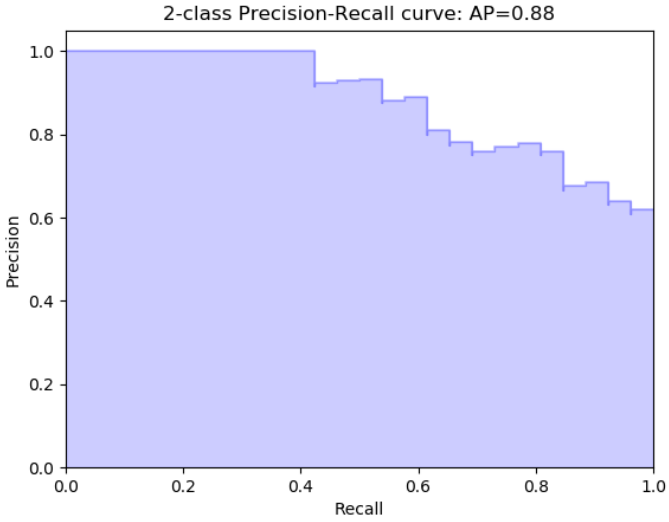

Figure 4.7 Example of precision-recall curve

Average precision, commonly *AP*, summarizes such a plot as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight:

$$\mathrm{AP} = \sum_n (\mathrm{Recall}_n - \mathrm{Recall}_{n-1})\mathrm{Precision}_n$$

where n indicates the nth threshold.

A pair $(\mathrm{Recall}_k, \mathrm{Precision}_k)$is referred to as an operating point.
AP and the trapezoidal area under the operating points are common ways to summarize a precision-recall curve that lead to different results [49].

Given the high class imbalance issue of our problem we decide to utilize Average precision and precision-recall area under the curve over area under the ROC curve as a way to evaluate our models.

4.4 SHapley Additive exPlanations

Tension between model performance and interpretability of predictions requires a method that help users interpret predictions. A largely used method to interpret a model is to look at built-in feature importance scores. However, in the case of XGB, as for many other models, there are three optional measures that, with different concepts behind, aim to reflect the importance a feature has in a model: weight, cover, gain. The weight, cover, and gain methods above are all global feature attribution methods [32].
Above mentioned measures are inconsistent and can cause troubles in determining feature importance. To check for consistency it is well advised to run a different feature attribution method such as SHAP: the SHapley Additive exPlanations (SHAP) is a unified approach to explain the output of any machine learning model. SHAP connects game theory with local explanations, uniting several previous methods, that we are not going to discuss here, and representing the only possible consistent and locally accurate additive feature attribution method based on expectations.[32]

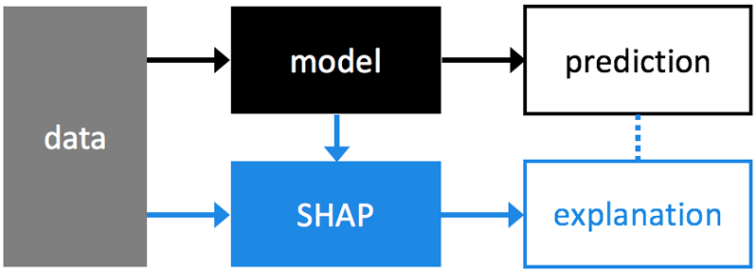

Figure 4.8 SHAP Python library pipeline

A prediction can be explained by assuming that each feature value of the instance is a player in a game where the prediction is the payout. Shapley values, a method from coalitional game theory, tells us how to fairly distribute the payout among the features. In general, the Shapley value is the average marginal contribution of a feature value across all possible coalitions. The goal of SHAP is to explain the impact of an instance x by computing

the contribution of each feature to the prediction.
For tree-based models TreeSHAP, a variant of SHAP for tree-based machine learning models such as decision trees, random forests and gradient boosted trees, exists. TreeSHAP is fast, computes exact Shapley values, and correctly estimates the Shapley values when features are dependent [32, 33].

To give an interpretation of the models we are going to present in the next section we rely on SHAP values to understand what's the impact of features on predicting power grid faults.

4.5 Results

In the previous sections of this chapter we came to the decision of utilizing eXtreme Gradient Boosting (XGB) as a machine learning model, of utilizing precision-recall curve in order to evaluate model performance addressing high class imbalance, and of inserting seasonal dummies as model features in order to account for seasonality. Moreover, we rely on SHAP to interpret models' predictions and to obtain an estimate of feature impact on the model.

The main source of information remain weather forecasts, which we evaluated in depth in the previous chapters. We also stated that, for research purposes, we are also concerned on features, and on the impact they have on predictions, and therefore on faults, and not only on model performance.

In this section we propose six possible models. From each of them we display the used dataset, the class-imbalance ratio, the number of features and observations, the full list of features, the precision-recall curve plot having in red, - -, a baseline = $\frac{P}{N}$, where P = positive observations (faults) and N = total observations.

We start were previous works left. As a first model we use daily averaged weather variables taken at the coordinates provided for the considered power station and seasonal dummies as features.

Model 1

month	day	fault	air_temperature_2m	air_pressure_at_sea_level	wind_speed_of_gust	thunderstorm_index_combined	...	surface_air_pressure
1	1	0	269.58	100176.95	8.56	0.00467	...	92848.3
	2	0	265.03	100756.41	5.89	0.00198	...	93392.67
	3	0	263.70	99581.39	9.16	0.00024	...	92134.84
	4	0	257.53	101720.65	4.78	0.0001	...	93958.83
	5	0	253.33	103209.134	2.54	0.000007	...	95410.24
	6	...	...	...	...	...	...	...

Table 4.1 Sample of dataset used for Model 1

Full list of features:
air temperature, *air pressure at sea level*, *cloud area fraction*, *wind speed of gust*, *thunderstorm index combined*, *relative humidity*, *precipitation amount*, *surface air pressure*.
Class imbalance ratio 1:7
# of features: 8
# of observations: 361

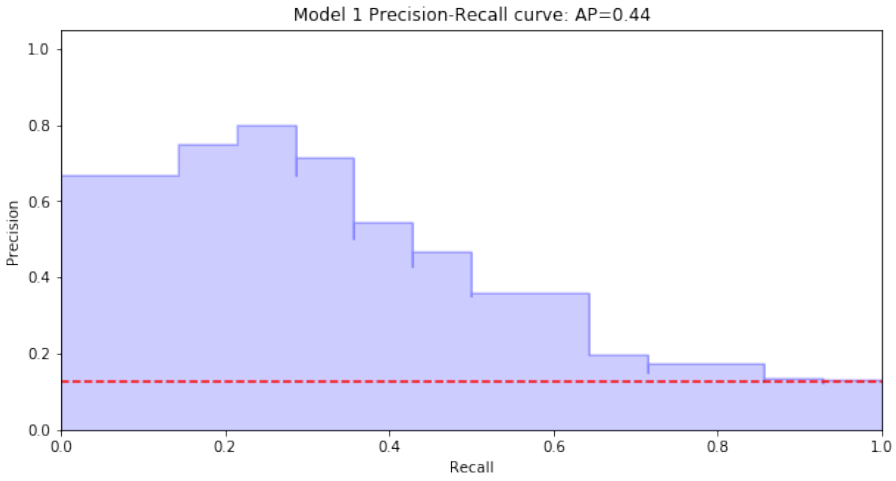

Figure 4.9 Precision-recall curve and average precision for Model 1

Average Precision score: 0.44

In the second model we move on from daily averaged weather and from daily labeled fault, increasing resolution to hourly.

Model 2

time	fault	air_temperature_2m	air_pressure_at_sea_level	wind_speed_of_gust	thunderstorm_index_combined	...	winter
1-1-17 00:00	0	269.58	100176.95	8.56	0.00467	...	1
1-1-17 01:00	0	265.03	100756.41	5.89	0.00198	...	1
1-1-17 02:00	0	263.70	99581.39	9.16	0.00024	...	1
1-1-17 03:00	0	257.53	101720.65	4.78	0.0001	...	1
1-1-17 04:00	0	253.33	103209.134	2.54	0.000007	...	1
1-1-17 05:00	...	...	...	...	...	...	...

Table 4.2 Sample of dataset used for Model 1

Full list of features:

air temperature, *air pressure at sea level*, *cloud area fraction*, *wind speed of gust*, *thunderstorm index combined*, *relative humidity*, *precipitation amount*, *surface air pressure*, *winter*, *spring*, *summer*, *autumn*.

Class imbalance ratio 1:54

# of features: 12

# of observations: 8750

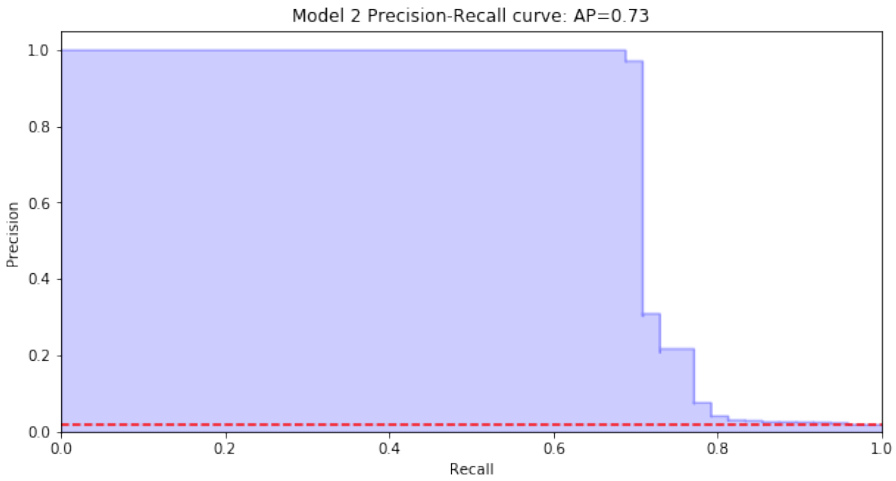

Figure 4.10 Precision-recall curve and average precision for Model 2

Average Precision score: 0.73

For Model 2 we compute the mean SHAP value for every feature, measuring its average impact on model output magnitude.

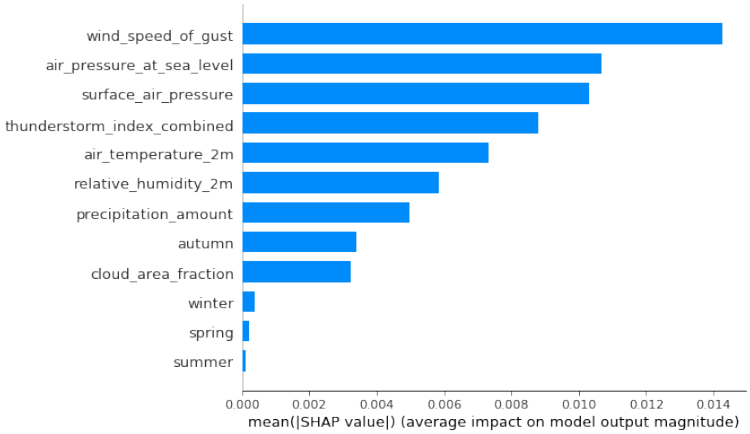

Figure 4.11 Average SHAP values for features in Model 2

Figure 4.11 hints that, among utilized features, we can identify features that are more relevant in determining the model output. Starting from the least important feature, we drop them all one by one and re-run the model with updated amount of features. Table 4.3 shows obtained results at varying of features:

n° feature	**average precision**
12	0.716
11	0.726
10	0.727
9	0.719
8	0.738
7	0.724
6	0.751
5	0.731
4	0.754
3	0.739
2	0.747
1	0.672

Table 4.3 Average precision score of Model 2 at varying of number of features

For Model 3 we maintain the same characteristics of Model 2, but we train it with only the four most important features, according to the results shown in table x.x.

Model 3

time	fault	surface_air_pressure	air_pressure_at_sea_level	wind_speed_of_gust	thunderstorm_index_combined
1-1-17 00:00	0	92329.6	100176.95	8.56	0.00467
1-1-17 01:00	0	92345.79	100756.41	5.89	0.00198
1-1-17 02:00	0	92325.38	99581.39	9.16	0.00024
1-1-17 03:00	0	92340.49	101720.65	4.78	0.0001
1-1-17 04:00	0	92343.12	103209.134	2.54	0.000007
1-1-17 05:00	...	...	...	...	...

Table 4.4 Sample of dataset used for Model 3

Full list of features:

air pressure at sea level, *wind speed of gust*, *thunderstorm index combined*, *surface air pressure*

Class imbalance ratio 1:54

# of features: 4

# of observations: 8750

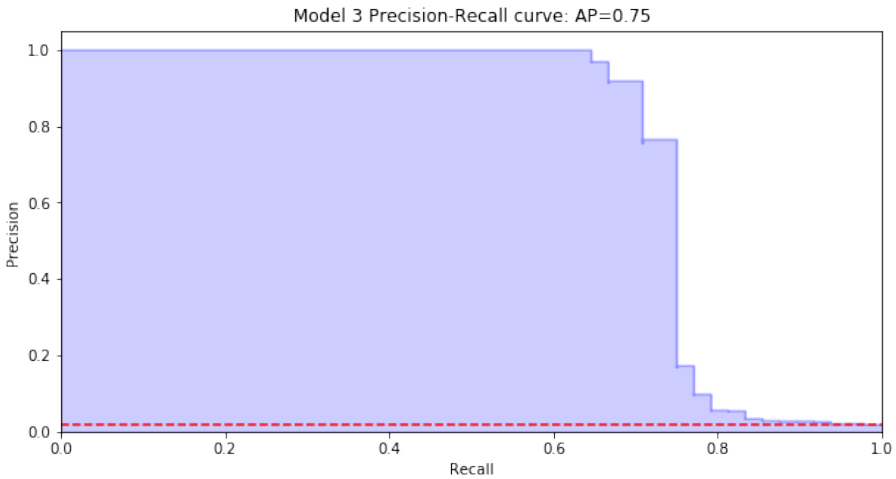

Figure 4.12 Precision-recall curve and average precision for Model 3

Average Precision score: 0.75

4.5.1 Accounting for missing fault position

As already mentioned in section 1.3, we do not know the exact position where a fault actually occurred. Faults data provided for this project only provide fault information at coordinates of the central point of a power station. However, a power station and its branches can cover large areas, an area where a fault could occur in any position. Still, previous works and our previous models only relied on weather data originated from the given coordinates of a fault: the coordinates of the central point of a power station.

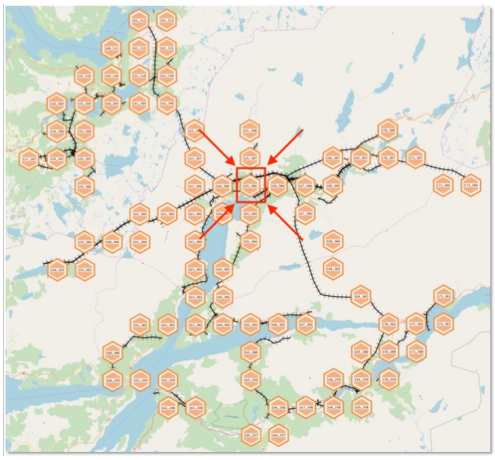

Figure 4.13 Hypothetical series of intersection between forecast grid and power grid

As an attempt to account for the impossibility to locate fault precise location, we cross the forecast grid with all power line branches, within a range with a diameter of 20 *km* from the central position, identifying all forecast cells that overlap the latter. In figure 4.13, black lines are power lines branches while all orange hexagons are the above mentioned forecast cell that cross them. The central cell highlighted in red is the cell that interests the central point of the power station and then the weather data source considered by previous works and our first three models. All cells differ not only in weather variables values, but also for terrain features.

We propose two approaches to integrate new weather information into the model, always without knowing the precise position of any fault occurrence: introducing multiple weather cells as new features or as new observations. Moreover, since we are already complicating the model we decide to increase the number of training feature from 8 to 14.

Model 4

time	fault	wind_speed_of_gust	...	wind_speed_of_gust_2	...	wind_speed_of_gust_n
1-1-17 00:00	0	8.56	...	5.6	...	4.2
1-1-17 01:00	0	5.89	...	3.44	...	2.94
1-1-17 02:00	0	9.16	...	3.56	...	2.99
1-1-17 03:00	0	4.78	...	4.41	...	3.47
1-1-17 04:00	0	2.54	...	4.87	...	8.31
1-1-17 05:00	...	...	...	...	...	...

Table 4.5 Sample of dataset used for Model 4

Full list of features for each cell:

air temperature, *cloud area fraction*, *fog area fraction*, *high type cloud area fraction*, *low type cloud area fraction*, *medium type cloud area fraction*, *precipitation amount*, *accumulated precipitation amount*, *relative humidity*, *surface air pressure*, *thunderstorm index combined*, *wind speed of gust*, *wind direction*, *wind speed* $*n$ of intersected cells.

Class imbalance ratio 1:54

# of features: 2394

# of observations: 8750

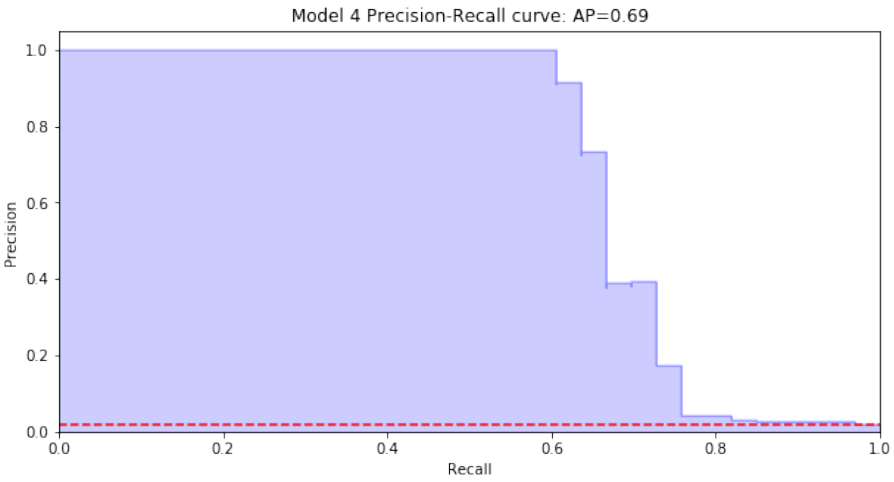

Figure 4.14 Precision-recall curve and average precision for Model 4

Average Precision score: 0.69

As for Model 2, we compute the mean SHAP value for every feature, measuring its averageimpact on model output magnitude.

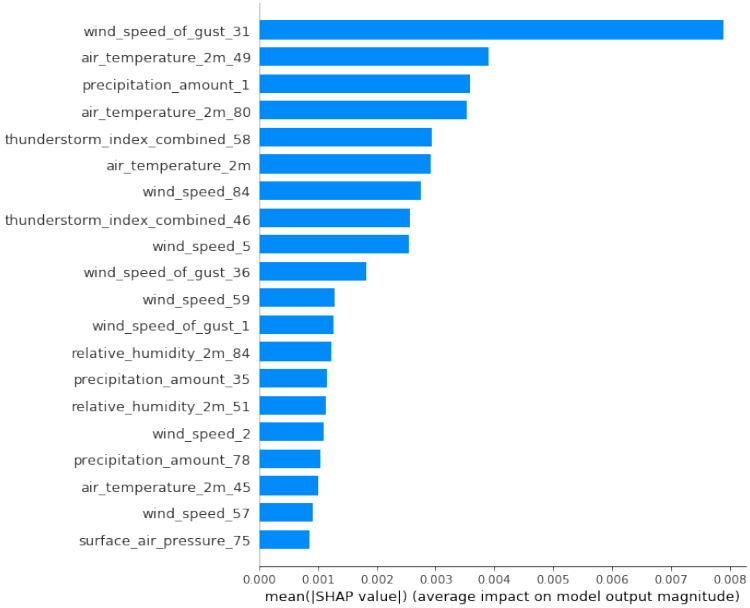

Figure 4.15 Average SHAP values for features in Model 4

SHAP values are calculated only for 20 features out of 2394. Differently from what happened for Model 2 though, re-running the model using only the most impact features do not show any improvement in terms of average precision, therefore we stick with Model 4.

n° feature	**average precision**
...	...
8	0.654
7	0.555
6	0.628
5	0.615
4	0.574
3	0.563
2	0.547
1	0.487

Table 4.6 Average precision score of Model 4 at varying of number of features

Model 5

time	cell	fault	wind_speed_of_gust	air_temperature_2m	fog_area_fraction	...	relative_humidity
1-1-17 00:00	0	0	8.56	-0.82	5.6	...	0.937
1-1-17 00:00	1	0	5.89	0.24	3.44	...	0.955
1-1-17 00:00	2	0	9.16	0.35	3.56	...	0.985
1-1-17 00:00	3	0	4.78	-1.29	4.41	...	0.986
1-1-17 00:00	4	0	2.54	1.55	4.87	...	0.975
1-1-17 00:00	5	...	...	...	...	...	...

Table 4.7 Sample of dataset used for Model 5

Full list of features:

air temperature, *cloud area fraction*, *fog area fraction*, *high type cloud area fraction*, *low type cloud area fraction*, *medium type cloud area fraction*, *precipitation amount*, *accumulated precipitation amount*, *relative humidity*, *surface air pressure*, *thunderstorm index combined*, *wind speed of gust*, *wind direction*, *wind speed*

Class imbalance ratio 1:54

# of features: 14

# of observations: 746513

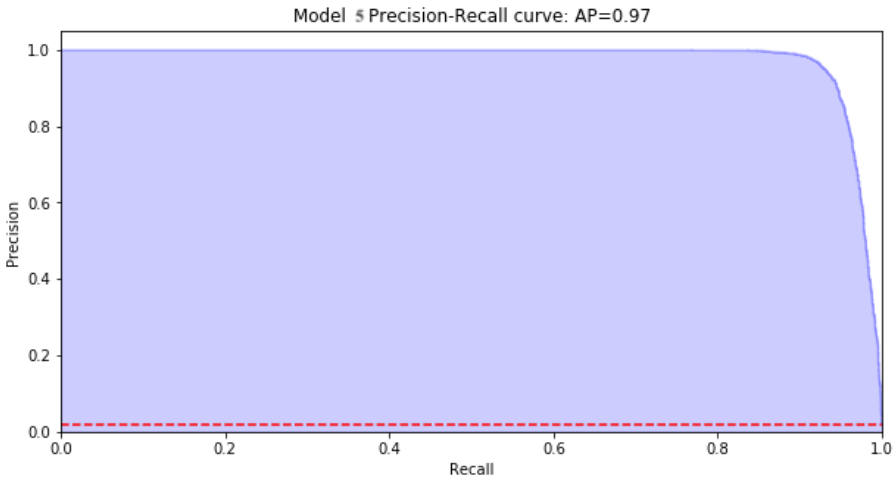

Figure 4.16 Precision-recall curve and average precision for Model 5

Average Precision score: 0.97

Since Model 5 registers the highest average precision score seen so far we integrate terrain indexes seen in section 4.5 to further differentiate weather cells.

Model 6

time	cell	fault	wind_speed_of_gust	...	elevation	TRI	TPI	roughness	slope	aspect
1-1-17 00:00	0	0	8.56	...	5.6	271.59	211.13	310.45	1.71	252.47
1-1-17 00:00	1	0	5.89	...	3.44	275.56	195.37	172.34	3.54	300.11
1-1-17 00:00	2	0	9.16	...	3.56	118.13	0.0	158.13	5.21	12.67
1-1-17 00:00	3	0	4.78	...	4.41	115.73	61.23	45.71	2.43	129.13
1-1-17 00:00	4	0	2.54	...	4.87	185.35	155.44	211.45	6.33	148.71
1-1-17 00:00	5	...	...	...	...	...	...	...	...	...

Table 4.8 Sample of dataset used for Model 6

Full list of features:

air temperature, *cloud area fraction*, *fog area fraction*, *high type cloud area fraction*, *low type cloud area fraction*, *medium type cloud area fraction*, *precipitation amount*, *accumulated precipitation amount*, *relative humidity*, *surface air pressure*, *thunderstorm index combined*, *wind speed of gust*, *wind direction*, *wind speed*, *elevation*, *TPI*, *TRI*, *roughness*, *slope*, *aspect*.

Class imbalance ratio 1:54

# of features: 14

# of observations: 746513

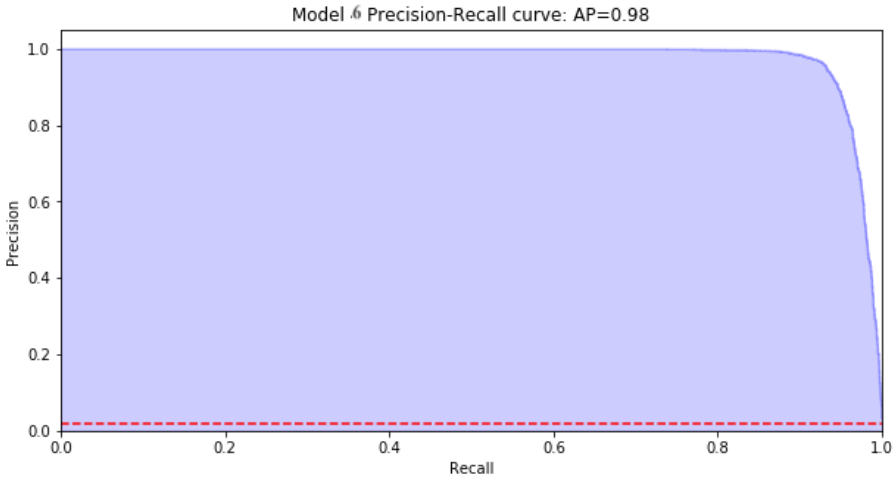

Figure 4.17 Precision-recall curve and average precision for Model 6

Average Precision score: 0.98

4.6 Discussion

In this section we retrace all models shown and discuss obtained results in details.

Model 1 is a slightly adjusted reproposal of how model of the work described in subsection 1.2.2 is set, where the author tried to predict faults at a daily resolution building different models for different seasons of the year and for different type of faults. The use of daily resolution results in a fairly low class-imbalance ratio, but also in a poor number of observation, 361.
For Model 2 we increase the number of observations by increasing fault data resolution, and therefore all other features resolution, to hourly. Performance-wise, having an higher number of observations helps the model, while an increased class-imbalance ratio, 1:54, does not affect XGB, which is able to handle it. The transition from a 0.44 average precision model to a 0.73 average precision model signs a clean difference in performance.

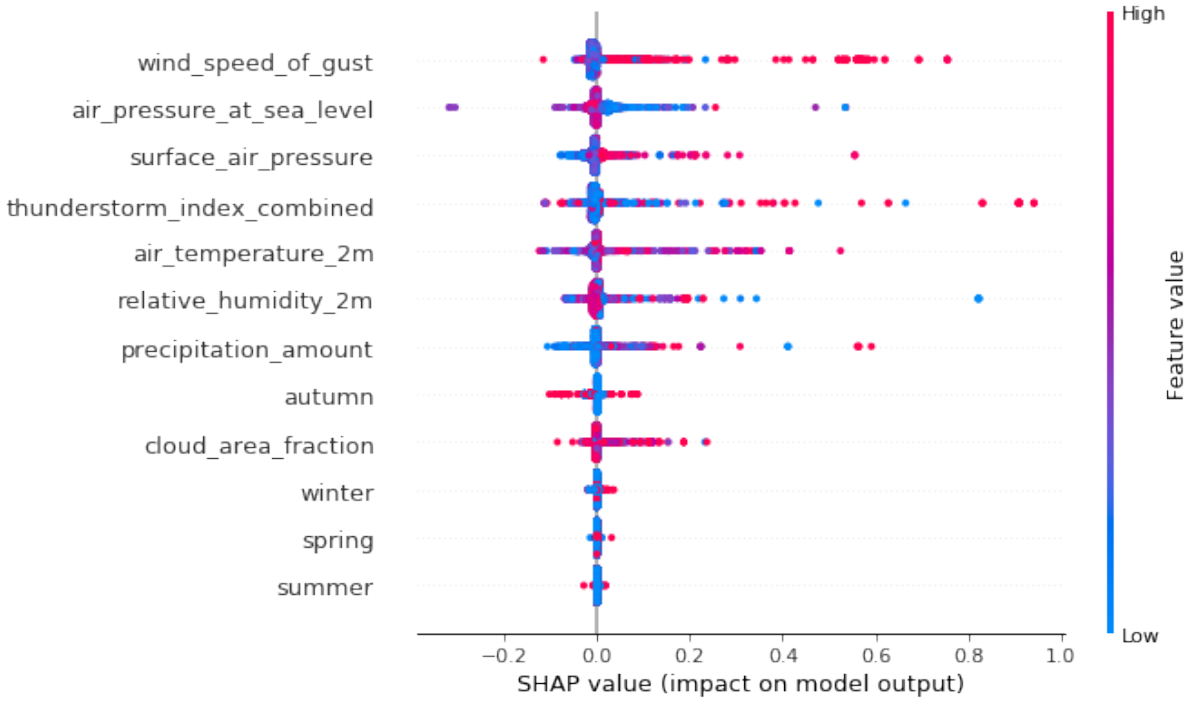

Figure 4.18 Feature impact on model output for Model 2

We can see in figure 4.18 how features contribute to the model prediction. The color gradient help us understand how every single value of every feature impacts the model. A

first assumption we can make out of figure 4.18 is that high values of *wind speed of gust*, *thunderstorm index combined* and *precipitation amount* stand out on the SHAP values scale as the observations with higher impact on the model output. Even though there is proof of weather values overlapping between positive and negative class, feature such as air pressure at sea level can have its relevance into the model.

Plotting for feature impact is also an attempt to simplify the model by running one with a lower dimensionality and to see how performance in terms of average precision change. Turns out using just the first four features, sorted for impact, gives better performances. In Model 3 we adjust by decreasing the number of features from twelve to four.

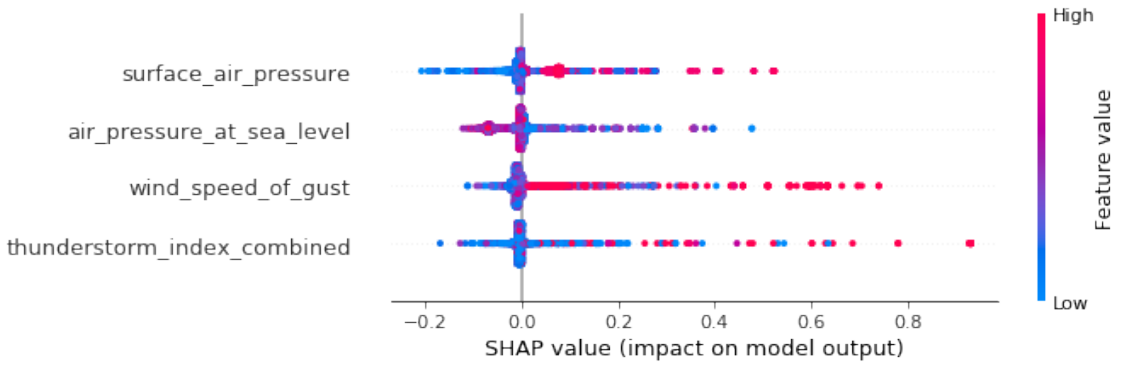

Figure 4.19 Feature impact on model output for Model 3

Model performance increase to 0.75 in average precision and features start to interact with the model in a slightly different way. In figure 4.19, the distribution of observations along the SHAP values axis for *surface air pressure* and *air pressure at sea level* changes, as they take over *wind speed of gust* and *thunderstorm index combined* in average impact on model output.

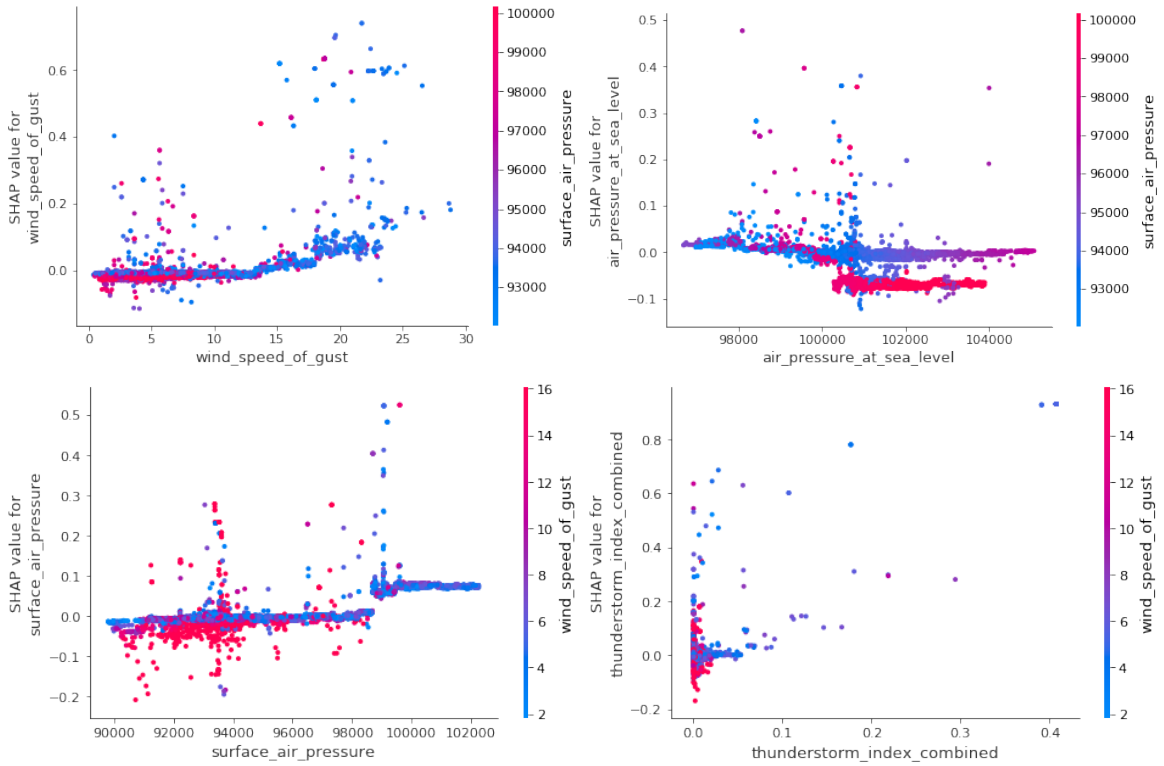

Figure 4.20 SHAP dependence plot for Model 3

While a SHAP summary plot gives a general overview of each feature, here in figure 4.20 present SHAP dependence plots. These plots show how the model output varies by feature value. Note again that every dot is an observation, and the vertical dispersion at a single feature value results from interaction effects in the model. The feature used for coloring is automatically chosen to highlight what might be driving these interactions. Weather variable values overlap makes it difficult to interpret such plots. However, it is interesting there is enough evidence to highlight some relations between variables and the model: *wind speed of gust* has higher chances to impact the model when its values are high. On the contrary, when *wind speed of gust* values are lower, *surface air pressure* has very little, if non negative, relation with the target variable. Again, when *surface air pressure* is at its highest, *air pressure at sea level* tends to have negative correlation with the model output.

Model 4, 5 and 6 represent an approach to introduce multiple weather cells in order to account for the missing fault position and terrain information as a measure of forecast reliability:

Model 4 introduces weather variables, in a higher number if compared to Model 2, for all forecast cells intersecting power grid lines as new features. This widely increase dimensionality of the model, while maintaining the same number of observations and the same class-imbalance ratio. Model performance decreases, comparing it to previous models. The addition of new features doesn't just increase dimensionality, a factor that XGB should be able to address, but also introduces a lot of noise due to the redundancy of adding multiple weather cells.

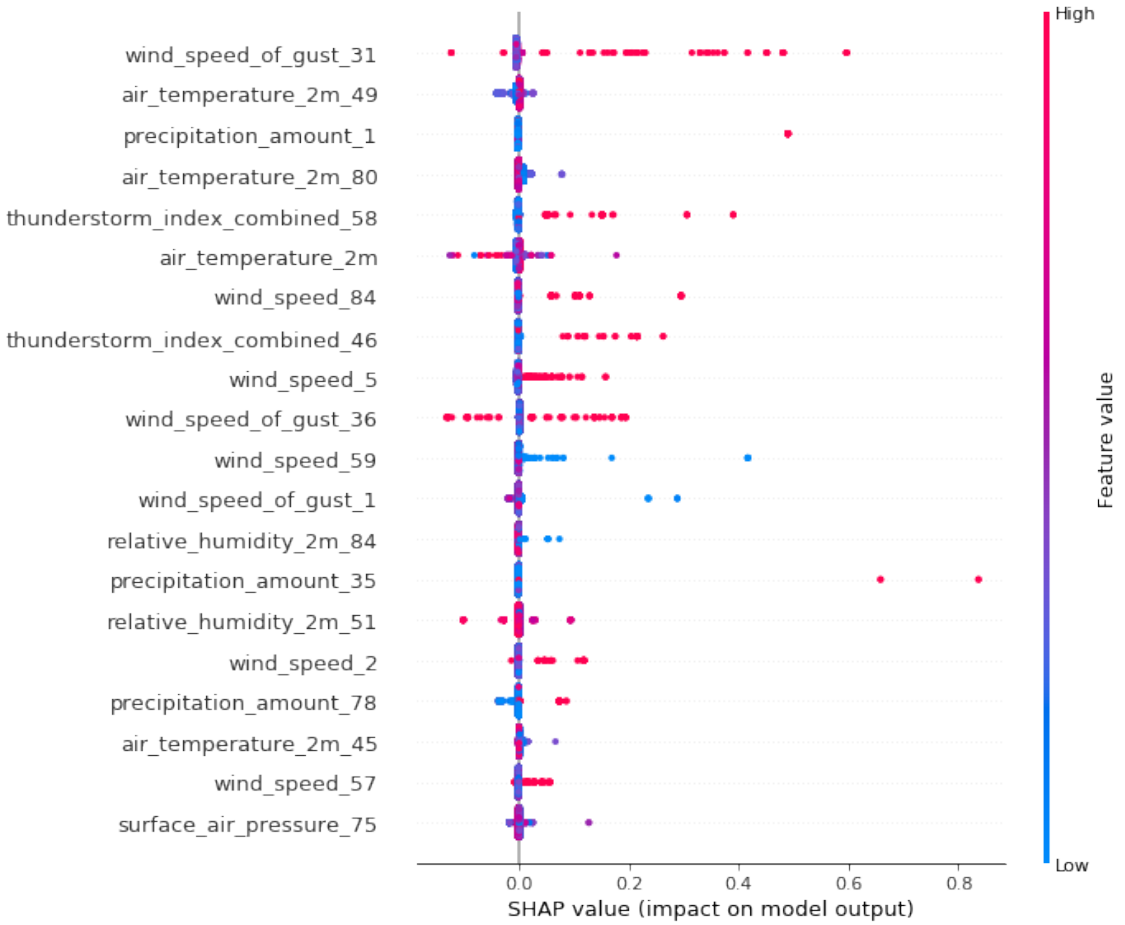

Figure 4.21 Feature impact on model output for Model 4

Figure 4.21 shows SHAP summary for most important features on Model 4. We do not push for further interpretation of the model, given its complexity and not satisfying performance. What can be interesting to note though, since features are reported with the identification number of their relative cell, is that same features may have different impact when originated from different cells, as for *wind speed of gust 31* and *wind speed of gust 1*.

In Model 5 we try a different approach to add more weather cells to the model. We go back to the original number of features, 14, and insert data about multiple weather cells as new observations. For the same *time* now we have one observation for every new weather cell we add. While maintaining class-imbalance ratio, this approach widely increases the number of available observations and, as expectable, model performance in terms of average precision score.

Model 6 is a reproposal of Model 5 with the addition of the terrain features seen in subsection 3.4.2 to further diversify newly added cells. Model performance reaches 0.98 of average precision score, slightly improving what in Model 5 was an already good result.

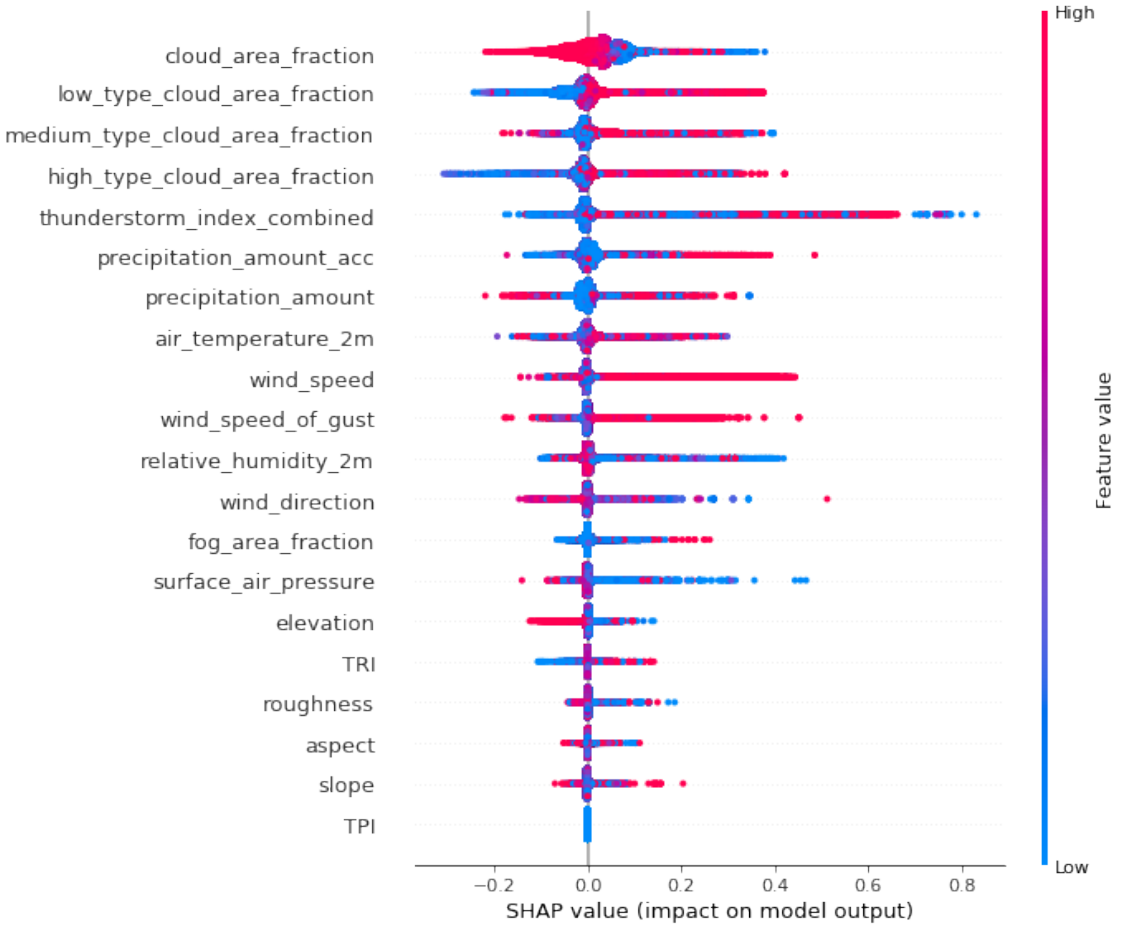

Figure 4.22 Feature impact on model output for Model 6

The SHAP summary displayed in figure 4.22 shows how in this last model features interact differently with the model output. All four cloud area fraction related features tend to have a net impact on the target variable, both with clear negative or positive relation, while speed of gust falls down a few positions in relevance. Terrain related features are of poor impact for the model output, but still somehow relevant.

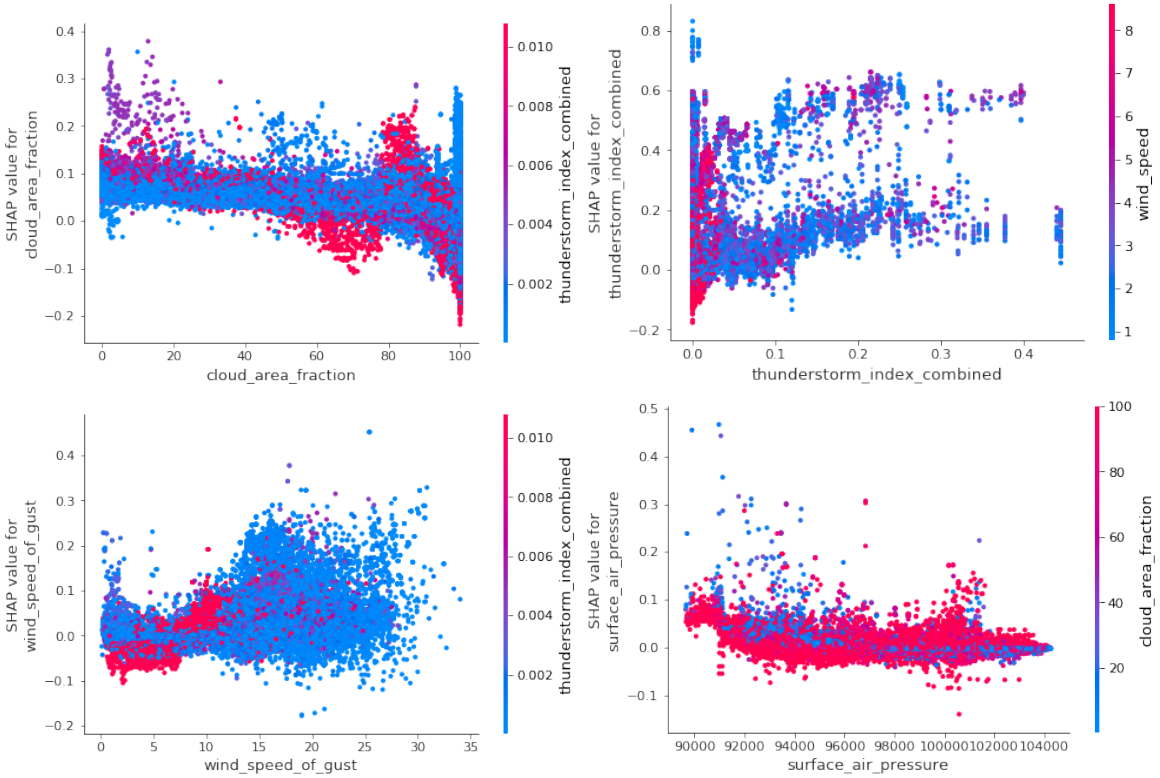

Figure 4.23 SHAP dependence plot for Model 6

Again, there is no clear and absolute interpretation of the model. A value from a given feature can result in having none to high impact on the model. High values in *wind speed of gust*, for example, may lead to a fault but at the same time be present at many non-fault times. This overlapping makes our SHAP dependence plots noisy. Still, we can find interesting relationship to add to what learned from the SHAP summary plot in figure 4.23: *thunderstorm index combined* has its highest impact when *wind speed* remains with low values. It may look like the information is carried by the combination of too few observation. However, also *wind speed of gust* increases in impact on the model output when lower values of *thunderstorm index combined* are observed.

4.6.1 Benchmarking and comparison with previous work

Even though we are concerned with feature relevance on fault prediction, we need to properly benchmark our results and compare them with those obtained from previous projects.
Being developed from different concepts and having different characteristics, all our models can be valuable tools when trying to predict faults on the power grid and to understand how features interact between themselves and with the model output.

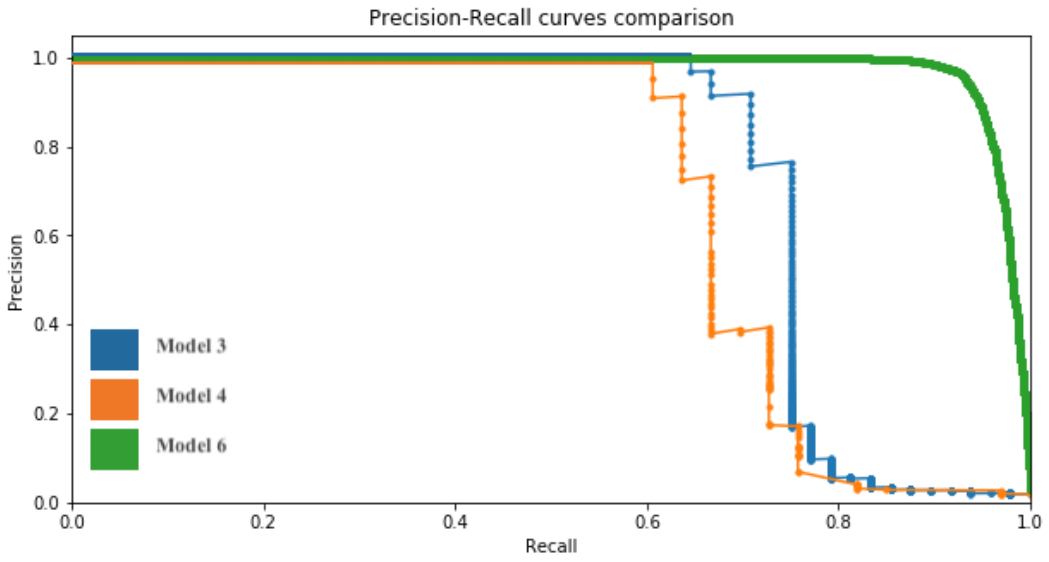

Figure 4.24 Precision-recall curve comparison for Model 3, 4, 6

Figure 4.24 shows how our more significant models' precision-recall curves stand when compared to each other. All three curves can be considered to be satisfying. All three models have flaws though.
Model 3 offers the best interpretation possibilities, counting only 4 features.
In model 4 average precision drops, as does the area under the precision-recall curve. The model has high complexity, and other than offering lower performance is more difficult to interpret. Still, the model can offer plausible insight about the different impact of weather features in different points of the power grid.
Model 6 has the point of interest of including terrain features. As hinted by a 0.98 average

precision, its precision-recall curve is close to the perfect one. However, this model can be prone to overfitting, since we used a dataset filled with observations that may be redundant.

The best model presented in the work mentioned in subsection 1.2.2 obtains an F1-score of 0.38. Figure 4.25 shows the F1 score of all our models, as a function of classification threshold, compared to the F1 score obtained by the previous work. All our models consistently obtain higher F1-score values, an improvement from the previous state of the problem.

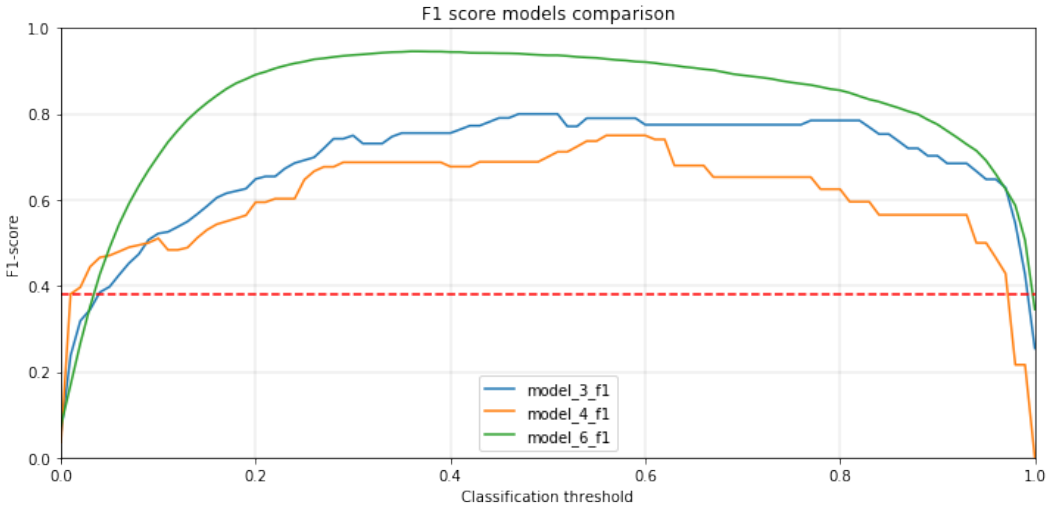

Figure 4.25 Comparison of F1-score, as a function of classification threshold, for Model 3, 4, 6 and work at subsection 1.2.2

Given the definition of precision-recall curve, average precision and F1, recalled in section 4.3, picking different points on the curves showed in figure 4.24 and in figure 4.25, being them precision-recall curves or F1 in function of classification threshold, favors different type of errors over others. Opting for higher Precision means favoring False Negatives while opting for higher Recall means privileging False Positives. We could argue that in a task such as predicting power grid faults a False Negative might cost more than a False Positive, but we better leave business choices about such a delicate balance to the interested parts.

5

Conclusions

As a conclusion to this work, we give our answer to the research questions we posed ourselves in section 1.3. Then we address what could be a possible follow-on and what would take to further improve the state of the problem.

5.1 Answer to research questions

Research question 1:

How good weather forecasts are? Through chapter 2 and 3 we have performed a detailed evaluation of weather forecast quality.

In chapter 2 we have kept to what MET regularly does in terms of evaluation of MetCoOp Ensemble Prediction System outputs. We have computed error measures listed in section 2.2, comparing weather forecast against real weather observations, obtaining encouraging results, coherent with those scored for MEPS outputs in year 2016.

In chapter 3 we have dug more in depth to understand how forecast error behaves in time and space. We found out that forecast error is related to the forecast value itself, underlining a relation between high error and harsh weather. Forecast error increases in value and vari-

ability during winter months, where weather is generally harsher and, indeed, more difficult to predict. Forecast quality also differs in space. We have observed, through the use of how multiple weather stations, how forecast quality differs in different places, as seen in section 3.3. To better study what are the physical characteristics of space, we have introduced and integrated Digital Elevation Models into our work. Through DEMs we have been able to calculate a variety of terrain indexes such as TRI, TPI, roughness, slope, aspect, and to see how all of them combined can relate to forecast error. We have also been able to build elevation and slope categories, thus dividing our analysis into categories that allow us to expand our understanding of how forecast error trends at different latitudes and longitudes. An elevation categorization confirms how mountainous environments make weather harder to predict, while a slope categorization allows to describe space with more details.
Overall, we have gained valuable insights about weather forecast that can be not only helpful for the rest of our work, but also interesting knowledge for MET.

Research question 2:
How can our understanding of weather data translate into new methodologies and help improve model performances?
Chapter 4 has been dedicated to the improving of the state of the art described in section 1.2. An exploratory analysis of fault data, along with previous knowledge of the problem and with information gained while answering to research question 1, have led to the decision to implement eXtreeme Gradient Boosting as our predicitve algorithm and to evaluate it using precison-recall curve and average precision, while giving interpretation exploiting SHAP values.
After reproposing models similar to what done in previous works, model 1-3, we have proposed two approaches to address the impossibility to know the exact position of a fault occurrence.
Even though all models can be improved and have flaws determined by the approach chosen for their development, they all outscore best model produced in the previous work marking a major improvement in terms of model performance.

5.2 Future work

The state of the problem we have faced is strongly determined by data we have available. Class imbalance is not going to disappear, but more fault observations can be of use. In particular, it would be highly beneficial to dispose of faults registered for more power stations. Staying on faults, it would be fundamental to know the precise position of a fault. Having fault location only related to the central point of a power station is too generic. Having the precise position of a fault would also make terrain information more relevant than what they are now in terms of modelling.
It would also be interesting to accompany weather data with power station functioning features, to provide internal and not only external or environmental information the model.

Most predictive problems can benefit of better, more numerous and more accurate data. However, it would still be valuable to try different algorithms or different approaches such as Deep Learning, and therefore Neural Networks.

List of Figures

List of Tables

Bibliography

[1] Karin Alvehag and Lennart Soder. A reliability model for distribution systems incorporating seasonal variations in severe weather. *IEEE Transactions on Power Delivery*, 26 (2):910–919, 2010.

[2] Nagaraj Balijepalli, Subrahmanyam S Venkata, Charles W Richter, Richard D Christie, and Vito J Longo. Distribution system reliability assessment due to lightning storms. *IEEE Transactions on Power Delivery*, 20(3):2153–2159, 2005.

[3] Patrick M Bartier and C Peter Keller. Multivariate interpolation to incorporate thematic surface data using inverse distance weighting (idw). *Computers & Geosciences*, 22(7): 795–799, 1996.

[4] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

[5] Lisa Bengtsson, Ulf Andrae, Trygve Aspelien, Yurii Batrak, Javier Calvo, Wim de Rooy, Emily Gleeson, Bent Hansen-Sass, Mariken Homleid, Mariano Hortal, et al. The harmonie–arome model configuration in the aladin–hirlam nwp system. *Monthly Weather Review*, 145(5):1919–1935, 2017.

[6] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.

[7] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. URL http://arxiv.org/abs/1603.02754.

[8] Fotini Katopodes Chow, Stephan FJ De Wekker, and Bradley J Snyder. *Mountain weather research and forecasting: recent progress and current challenges*. Springer, 2013.

[9] G.H. Coldevin. Smart grid in norway: Status and outlook, 2017. URL https://goo.gl/MyYYtn.

[10] International Electrotechnical Commision. Electropedia: The world's online electrotechnical vocabulary, 2018. URL https://goo.gl/b6mwSz.

[11] Roger Dev. Learning trees – a guide to decision tree based machine learning, 2018. URL https://bit.ly/35c3J3n.

[12] Hassan Farhangi. The path of the smart grid. *IEEE power and energy magazine*, 8(1): 18–28, 2009.

[13] SINTEF Referansegruppe for feil og avbrudd. Definitions related to errors and interruptions in the electric power system (definisjoner knyttet til feil og avbrudd i det elektriske kraftsystemet), 2001. URL https://goo.gl/weofWo.

[14] Opplysningsrådet for Veitrafikken AS. Bilsalget i september 2018 i norge (car sales in september 2018 in norway, 2018.

[15] Even Rouault Frank Warmerdam et al. Gdal documentation, 2019. URL https://gdal.org/programs/gdaldem.html.

[16] GIS Geography. Inverse distance weighting (idw) interpolation, 2018. URL https://bit.ly/2Kz7bNT.

[17] Geopy. Geopy's documentation, 2019. URL https://geopy.readthedocs.io/.

[18] Michael Glantz. The value of a long-range weather forecast for the west african sahel. *Bulletin of the American Meteorological Society*, 58(2):150–158, 1977.

[19] Norwegian Government. White paper on norway's energy policy: Power for change, 2016. URL https://goo.gl/pMQnxs.

[20] Norwegian Government. Renewable energy production in norway, 2018. URL https://goo.gl/CDhyXD.

[21] P Grover. Gradient boosting from scratch. *Retrieved from Medium*, 2017.

[22] Jeff Haby. Influence of topography on weather, 2019. URL https://bit.ly/33YR2ZK.

[23] Seung-Ryong Han, Seth D Guikema, and Steven M Quiring. Improving the predictive accuracy of hurricane power outage forecasts using generalized additive models. *Risk Analysis: An International Journal*, 29(10):1443–1453, 2009.

[24] Mariken Homleid and Frank Thomas Tveter. Verification of operational weather prediction models september to november 2016. *METInfo Rep*, 16:2016, 2016.

[25] Renuka Joshi. Accuracy precision recall f1 score: Interpretation of performance measures. *Retrieved April*, 1:2018, 2016.

[26] Padmavathy Kankanala, Anil Pahwa, and Sanjoy Das. Estimation of overhead distribution system outages caused by wind and lightning using an artificial neural network. In *Proc. Int. Conf. Power Syst. Oper. Plan.*, 2012.

[27] kartverket.no. Open and free geospatial data from norway, 2018. URL https://bit.ly/2NVfCFb.

[28] Jørn Kristiansen, Ulf Andae, Heiner Körnich, Sami Niemelä, Mikko Partio, and Ole Vignes. The metcoop ensemble prediction system for nordic weather conditions. In *99th American Meteorological Society Annual Meeting*. AMS, 2019.

[29] Carolyn LaRoche. What are the similarities between weather climate?, 2017. URL https://bit.ly/2qYFCqd.

[30] Jake Lever, Martin Krzywinski, and Naomi Altman. Points of significance: classification evaluation, 2016.

[31] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[32] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874, 2017. URL http://arxiv.org/abs/1705.07874.

[33] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. Explainable ai for trees: From local explanations to global understanding. *arXiv preprint arXiv:1905.04610*, 2019.

[34] Linus Magnusson. How to pinpoint the sources of forecast errors, 2015. URL https://bit.ly/2OmWDm7.

[35] Linus Magnusson. Diagnostic methods for understanding the origin of forecast errors. *Quarterly Journal of the Royal Meteorological Society*, 143(706):2129–2142, 2017.

[36] Pascal J Mailier, Ian T Jolliffe, and David B Stephenson. Quality of weather forecasts. *Review and recommendations Royal Meteorological Society*, pages 1–89, 2006.

[37] Brock Adam McCarty. Working in gdal (wig) – creating a terrain roughness index map, 2012. URL https://bit.ly/35hcnhl.

[38] MET. met.no threddscatalog, 2019. URL http://thredds.met.no/.

[39] KE Michalowska. Predicting faults in the norwegian power distribution grid. Master's thesis, 2018.

[40] Franco Molteni, Roberto Buizza, Tim N Palmer, and Thomas Petroliagis. The ecmwf ensemble prediction system: Methodology and validation. *Quarterly journal of the royal meteorological society*, 122(529):73–119, 1996.

[41] Vishal Morde. Xgboost algorithm: Long may she reign, 2019.

[42] A Muir and J Lopatto. Final report on the august 14, 2003 blackout in the united states and canada: causes and recommendations, 2004.

[43] Malte Müller, Mariken Homleid, Karl-Ivar Ivarsson, Morten AØ Køltzow, Magnus Lindskog, Knut Helge Midtbø, Ulf Andrae, Trygve Aspelien, Lars Berggren, Dag Bjørge, et al. Arome-metcoop: A nordic convective-scale operational weather prediction model. *Weather and Forecasting*, 32(2):609–627, 2017.

[44] Roshanak Nateghi, Seth D Guikema, and Steven M Quiring. Forecasting hurricane-induced power outage durations. *Natural hazards*, 74(3):1795–1811, 2014.

[45] MET Norway. Met norway's archive of historical weather and climate data, 2019. URL https://frost.met.no/.

[46] Didier Ntwali, Bob Alex Ogwang, and Victor Ongoma. The impacts of topography on spatial and temporal rainfall distribution over rwanda based on wrf model. *Atmospheric and Climate Sciences*, 6(02):145, 2016.

[47] Ministry of Petroleum and Energy (Olje og energidepartementet). Regulations on the quality of delivery in the power system (forskrift om leveringskvalitet i kraftsystemet), 2004. URL https://goo.gl/F3meZH.

[48] Tim Palmer and Renate Hagedorn. *Predictability of weather and climate*. Cambridge University Press, 2006.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[50] Rahmat Poudineh and Tooraj Jamasb. Electricity supply interruptions–sectoral interdependencies and the cost of energy not served for the scottish economy. 2015.

[51] SINTEF Energy Research. Earlywarn, 2018. URL https://goo.gl/PYgV7f.

[52] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[53] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10 (3):1–21, 03 2015. doi: 10.1371/journal.pone.0118432. URL https://doi.org/10.1371/journal.pone.0118432.

[54] Abdullahi M Salman and Yue Li. Multihazard risk assessment of electric power systems. *Journal of Structural Engineering*, 143(3):04016198, 2016.

[55] Abdullahi M Salman and Yue Li. A probabilistic framework for seismic risk assessment of electric power systems. *Procedia engineering*, 199:1187–1192, 2017.

[56] Y Seity, P Brousseau, Sl Malardel, G Hello, P Bénard, F Bouttier, C Lac, and V Masson. The arome-france convective-scale operational model. *Monthly Weather Review*, 139 (3):976–991, 2011.

[57] Statistics Norway SSB, Statistisk Sentralbyrå. Statistical data, electricity production in norway in 2017, 2017. URL https://goo.gl/CDhyXD.

[58] avdeling Feilanalyse Statnett SF. Annual statistics 2016. operational disturbances, faults and scheduled disconnections in the 1-22 kv network (arsstatistikk 2016. driftsforstyrrelser, feil og planlagte utkoplinger i 1-22 kv-nettet), 2017. URL https://goo.gl/tZQ4if.

[59] Hind Taud and Jean-François Parrot. Measurement of dem roughness using the local fractal dimension. *Géomorphologie: relief, processus, environnement*, 11(4):327–338, 2005.

[60] Gabriel Tseng. Gradient boosting and xgboost, 2018.

[61] UniData. Accessing netcdf data by coordinates, 2013. URL https://bit.ly/2KyKNnD.

[62] Willem A Wagenaar and JENNY G VISSER. The weather forecast under the weather. *Ergonomics*, 22(8):909–917, 1979.

[63] A.J. Walkley. Factors affecting weather climate, 2018. URL https://bit.ly/2OlC7lP.

[64] Leah Wasser, Jenny Palomino, and Chris Holdgraf. earthlab/earth-analytics-python-course: earthlab /earth-analytics-python-course: Version-1.0.1, 2019. URL https://doi.org/10.5281/zenodo.3523193.

[65] Weather.us. Global models with worldwide weather forecasts, 2019. URL https://weather.us/model-charts.

[66] Andrew Weiss. Topographic position and landforms analysis. In *Poster presentation, ESRI user conference, San Diego, CA*, volume 200, 2001.

[67] Margaret FJ Wilson, Brian O'Connell, Colin Brown, Janine C Guinan, and Anthony J Grehan. Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope. *Marine Geodesy*, 30(1-2):3–35, 2007.

[68] XGBoost. Xgboost documentation, 2019. URL https://xgboost.readthedocs.io/.

[69] Dan Zhu, Danling Cheng, Robert P Broadwater, and Charlie Scirbona. Storm modeling for prediction of power distribution system outages. *Electric power systems research*, 77(8):973–979, 2007.