Diese Arbeit wurde vorgelegt am
Lehr- und Forschungsgebiet Theorie der hybriden Systeme

Planung und Standorteignung von Windparks
Planning and Site Suitability of Wind Farms

Masterarbeit
Informatik

Dezember 2022

Vorgelegt von Presented by	Marcel Kröker Matrikelnummer: 343940 marcel.kroeker@rwth-aachen.de
Erstprüfer First examiner	Prof. Dr. rer. nat. Erika Ábrahám Lehr- und Forschungsgebiet: Theorie der hybriden Systeme RWTH Aachen University
Zweitprüfer Second examiner	Prof. Dr. rer. nat. Thomas Noll Lehr- und Forschungsgebiet: Software Modellierung und Verifikation RWTH Aachen University
Betreuer Supervisor	Dr. rer. nat. Pascal Richter Lehr- und Forschungsgebiet: Theorie der hybriden Systeme RWTH Aachen University

Contents

1 Introduction

With the drastic increase in worldwide electricity consumption since the industrial revolution, humanity had to find ways to meet these ever-increasing demands. For a long time, fossil fuels like coal and gas were the primary sources of electricity generation. However, their use has significant drawbacks. The burning of fossil fuels releases toxic pollutants and greenhouse gases into the atmosphere. The greenhouse gases prevent heat from being radiated away from earth into space, leading to increased temperatures on the planet [26]. The consequences of global warming have become very apparent in recent times, as we have seen many once-in-a-century natural disasters in the last few decades. Rising ocean levels due to the melting of ice on the earth's poles especially pose a threat to coastal regions [28]. An increasing number of floods, droughts, dropping air quality, and the outlook of damaging the planet beyond the point of no return have led to a worldwide re-evaluation of our energy consumption. The international consensus right now is that clean and renewable energy sources that do not emit greenhouse gases are an essential part of humanity's path to a sustainable future. This rethinking is reflected in the way we produce electricity. As seen in Figure 1, the worldwide installed wind energy capacity has seen unprecedented growth in the past two decades.

Besides the effects of climate change, political events have also necessitated a fast shift to green energy sources. For example, the war between Russia and Ukraine has resulted in severe sanctions against Russia, especially against their fossil fuel exports. Germany reacted quickly and decided to achieve 100 percent renewable energy sources for electricity by 2035, which is significantly faster than previously intended [10]. Especially wind power generation is thought to be doubled in the next eight years. New regulations for wind farms to speed up the planning and construction process can and should be expected due to the currently costly and slow process. For example, in Germany, the total duration from the initial planning phase until taking the park into operation takes five years on average [7].

As seen in Figure 2, fossil fuels are still the dominant source of electricity production. Luckily, renewable energy sources like hydropower, wind, solar, and bioenergy are on the rise. In this thesis, we will focus specifically on wind power, which comes with several advantages over other renewable energy sources. For example, it has the capacity to generate electricity at night, unlike solar energy sources. And it is more scalable than hydropower because it can be installed in a wider range of locations, especially in countries with a flat topography.

Despite the advantages, wind power also comes with its own unique set of problems. One concern is consistency, since the wind strength can vary unpredictably. Another difficulty is that onshore wind farms can cause conflict with nearby residents. For example, the appearance of wind farms can negatively impact the perception of high-quality and beautiful landscapes [25]. Additionally, wind turbine noise can potentially negatively impact nearby residents' sleep [27]. Moreover, laws exist that require

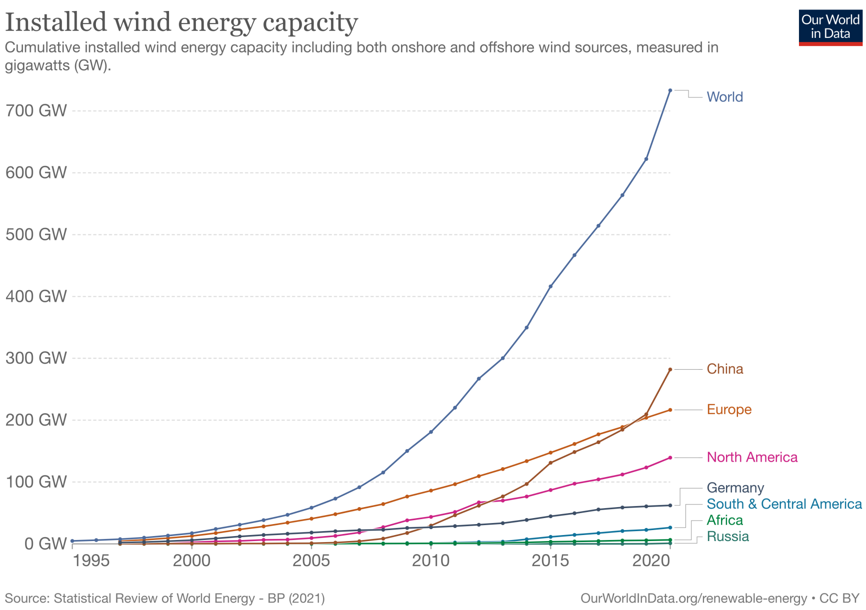

Figure 1: The installed wind energy capacity from 1995 to 2021 for the World, China, Europe, North America, Germany, South & Central America, Africa, and Russia. The world total has increased drastically, with only 4.86 gigawatts in 1995, up to 733.28 gigawatts in 2021. Especially China has made considerable advancements recently, while Germany somewhat stagnated.

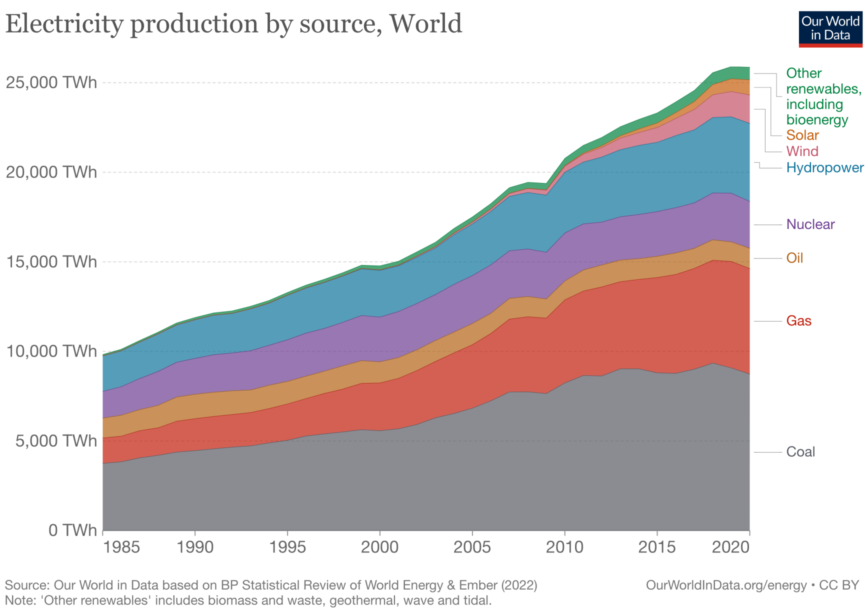

Figure 2: The global sources for electricity production [8], from 1985 to 2020. Hydropower generation has been around for longer, but other renewables like wind and solar only reached a noteworthy share in the past 20 years. Nevertheless, fossil fuels make up the vast majority.

minimum distances to residential areas to minimize the adverse health effects of the flickering shadows of wind turbine blades. In addition, land topography and local wind conditions play a role. Historic wind conditions, the wind zone of a planned site, and the surrounding terrain properties dictate which types of wind turbines can be used. The fact that the internal layout of a wind farm can prevent the construction permit also needs to be considered. Since internal turbulences might cause the turbulence intensity limit of individual turbines to be exceeded, a wind farm layout might be infeasible to construct. Even when considering all regulations and limits, the placement of wind turbines still requires more analysis. The turbulences within the wind farm by individual wind turbines or even nearby already constructed wind turbines reduce the total wind farm's efficiency.

Because of these considerations, wind farms often need to go through a lengthy process of iterated feedback with residents and validation reports of new layouts. Moreover, since the emergence of widespread wind farms is relatively recent, lawmakers change regulations frequently. For example, the minimum distance to buildings required by law can change quickly due to new political goals and can result in entirely different designs.

All of these factors make it challenging to plan a wind farm, resulting in a planning phase normally taking multiple years. Planners need to react quickly to changing laws and resident feedback. When a design is altered, they must re-evaluate all related factors, such as noise propagation, shadow cast, site suitability, and the total energy yield.

1.1 Contribution

This work aims to address the problems associated with the planning process of wind farms. We intend to contribute to improving the planning process by providing:

1. A model and implementation for evaluation of the site suitability for a wind turbine in relation to topography, wind, turbulence, and the local regulations in Germany.

2. An interactive tool for wind farm planners that they can use to plan and validate their designs quickly.

Our tool should allow wind farm planners to create projects, which can contain different possible wind turbine placement layouts. For each layout, they should be able to evaluate different metrics regarding the planned design, including the site suitability of each turbine, the optimal layout, the shadow cast generated by the wind farm, and the noise propagation. The algorithms for determining the shadow cast, noise propagation and optimal layout already exist in the broader context of this research project, and will be used by the planning tool. Moreover, we want to make the planning process as easy and intuitive as possible by examining user interface design principles and applying them accordingly.

In addition to the existing suite of algorithms, this work aims to provide the algorithm for determining a wind turbine's site suitability. It is known that the suitability of each wind turbine placement site in a wind farm is a crucial step of the planning process. It is useful to know if a wind farm contains wind turbines types that cannot be built there due to wind conditions and turbulences, so changes can be applied quickly. If we can determine the site suitability before doing a costly official assessment, the cost, and planning duration can be reduced.

We think that our contribution will have positive effects on the planning phase of a wind farm, potentially making the lengthy process a lot faster than before. Slow and costly iterations and multiple professional analysis reports can be reduced because the computations can be performed quickly within the tool. Thus, wind farms can be realized faster, which will ultimately help us reach our sustainability goals.

1.2 Outline

In Section 2, we will review the state of the art regarding the relevant literature for this work. Afterward, we are going into the details of determining the suitability for a wind turbine in Section 3. Using these insights, we will implement a site suitability model in Section 4, and compare it with an official site assessment. In Section 5, we will present laws and principles of user experience design. Those findings will be used as a guide to design the interface of the wind farm planning tool. Section 6 will discuss the planning tool's requirements, as well as the available technologies for building it, and we justify the choices that we made. Subsequently, Section 7 will depict the application's implementation and the usage workflows and present the interface. To conclude the work, Section 8 offers a final discussion and outlook on future work.

2 State of the art

A wind farm is a set of wind turbines that are placed in proximity to each other at some shared site. Some wind farms only consist of a few turbines, while the biggest of them span tens of square kilometers, consisting of hundreds of turbines [9].

Wind turbines are structures that convert the wind's kinetic energy into electricity. Their main components are the tower, the nacelle, and the rotor, as seen in Figure 3. Wind pushes against the rotor blades, and the rotor converts the wind-induced rotational force into electricity. The rotational energy of the rotor is transferred to a gearbox by the central axle to translate it to the optimal speed for the electric generator. The generator is similar in design to an electric motor but does the opposite, which means that it takes the wind's kinetic energy and converts it into electricity. A yaw control motor underneath the nacelle can turn the nacelle so that the rotor directly points in the direction the wind is coming from. Furthermore, a disc brake is attached to the axle to stop the rotor if the wind speed is above the turbine's cut-out speed.

While some wind turbines are small, others, like the Siemens Gamesa SG 14-222 DD [2] offshore wind turbine, can produce as much as 14 megawatts of nominal power, with a rotor diameter of 222 meters. Onshore turbines are slightly less powerful, with up to 7.5 megawatts [1]. There are offshore wind farms in the sea and onshore wind farms on land. They share the same goal of extracting as much energy from wind as possible while minimizing area due to connection costs, land costs, and geographical limitations.

2.1 Considerations before constructing a wind turbine

Because even an average wind turbine in a wind park is a considerable machine, it influences its environment in a multitude of ways, both positive and negative:

- They produce clean and renewable energy, helping to mitigate climate change.
- They reduce dependence on energy imports.
- They create noise that stems from both the generator, and the wind's interaction with the rotor blades.
- As the rotor turns, the blades throw a moving shadow, which is called shadow flicker.
- They have a potentially negative impact on the local bird population.
- Some people perceive the appearance of wind turbines as detrimental to the landscape.
- For construction, the power grid potentially has to be expanded to accommodate the generated power. Furthermore, access roads for construction might have to be built, which can be a problem for nature.

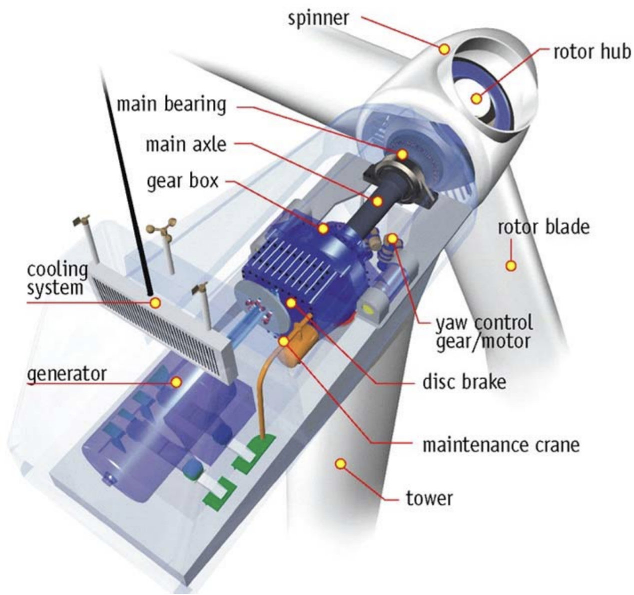

Figure 3: A common upwind turbine design, as depicted by Polinder et al. [33]. The nacelle houses the bulk of all systems. It sits on top of the wind turbine's tower. The rotor is attached to the nacelle, and drives the main axle.

On the other side, the environment also has impacts on the wind turbine:

- The wind conditions play an important role. At some sites, turbines cannot be constructed due to extreme wind conditions.
- The turbulence of nearby wind turbines can also influence the safety of a wind turbine.
- Extreme environmental conditions like the weather and earthquakes can damage it and lead to a shorter lifespan.

2.1.1 Shadow cast

Shadow flickers of rotating blades can be disturbing for nearby residents. Haac et al. [20] have modeled and predicted the annoyance of shadow flicker. They found out that with low exposures of less than eight hours per year, only about half of the residents even perceive shadow flicker in their homes. Their model was good at predicting the perceived shadow flicker. Interestingly, once perceived by a resident, higher flicker levels did not predict higher annoyance levels. The annoyance level was most correlated with the individuals' subjective response to the appearance of wind turbines, sensitivity to sound, level of education, and age.

2.1.2 Noise propagation and visual disturbance

Researchers like Haac et al. [21] and Hübner et al. [22] have observed similar results for the noise generated by a wind turbine. Visual perception of wind turbines, self-reported noise sensitivity, and the fact that someone moved in before or after the wind farm's construction were stronger annoyance predictors than the actual wind turbine audibility. They suggest that involving residents early on and improving the planning process might reduce perceived annoyance by making the perception of wind farms more positive.

Both shadow cast, and noise propagation have been studied in related theses. They implemented simulations of turbine noise propagation and shadow cast and used them to visualize the exposure for residents with a mobile application. By informing and positively involving residents, they aimed to improve the perception of planned wind farms so that the planning process is faced with less resistance. Their algorithms will be used by the wind farm planning application developed in our work.

2.1.3 Site suitability

There are multiple works which have analyzed how to determine the site suitability of a wind turbine or farm. Baseer et al. [15] used geographic information system (GIS) data and wind data to analyze the suitability over the entirety of Saudi Arabia, by considering the potential energy yield with statistical wind speed models. Their calculations also excluded or included regions by proximity to settlements, roads, airports, and the electrical gird.

Also measuring site suitability by energy yield, Petrov et al. [32] take a slightly different approach. By training and using a machine learning model, the evaluated the site suitability over the state of Iowa in the United States.

Many more researchers followed with similar reasoning about site suitability based on wind energy yield, like Bennui et al. [16] for provinces in southern Thailand and Rodman et al. [35] for northern California.

A different way of defining site suitability is to ask whether a turbine is allowed to be constructed at a specific site. We will face this question in the next subsection.

2.2 Site suitability based on wind and turbulence

The focus of this thesis is evaluating the suitability of a specific site for constructing a wind turbine. We will consider factors such as wind strength and direction, topography, and turbulence intensity to determine whether the site meets the design parameters and local regulations for wind farms. By analyzing these factors, we aim to answer whether a wind turbine can be successfully installed at the considered site.

2.2.1 Norms

There are multiple norms which regulate the site suitability of a wind turbine in Germany. The most important one is the ”Richtlinie für Windenergieanlagen” by DIBt [3], the German policy for wind turbines. It defines how the site suitability of a wind turbine must be determined in Germany, and which other regulations also apply. It frequently refers to models from the European norm DIN EN 1991-1-4 [4], so this norm is also essential in this context.

Another related norm is DIN EN IEC 61400-1 [6], which also provides site wind models. Additionally, it specifies the site suitability model for complex terrains, which the German policy for wind turbines [3] makes use of.

In Section 3, we will go into detail about these norms.

2.2.2 Existing tools and services

The open-source framework called OpenFAST[1] allows the simulation of wind turbines, as well as entire wind farms. It contains many possible simulations, from aerodynamic to structural dynamics. The aerodynamics module specifically can simulate rotor-wake effects, and the ”TurbSim” module can simulate turbulences. However, the tool is not related to any specific regulatory document, and only focuses on simulations.

Several commercial services exist which can assess the site suitability of a turbine. Wake2e[2] is a commercial tool which allows users to determine the site suitability of a planned wind turbine. The online application WakeGuard[3] allows users to determine effective turbulence intensities within a wind farm. They use the Frandsen model [18] for calculate added wake turbulence intensities. According to their website, I17 also sells official wind farm site suitability reports. Another commercial tool is the Wind Farm Assessment Tool (WAT) by WAsP[4].

Due to the commercial nature and high prices of all tools and services above, it is not possible to for us to use them. However, I17-Wind published a few site suitability reports online, which can be used for validation of our models.

2.2.3 Wind turbine wake effects

Wind turbines cause a wake effect of turbulence behind them. The wake is made from turbulent air, which lowers the efficiency of other turbines inside the wake, and increases the structural loads they are subjected to. Moskalenko et al. [29] have analyzed this effect and compared different mathematical models for it. They conclude that it plays a significant role, with an energy yield reduction of up to 12%.

The wake effect of the wind turbine tower structure has also been investigated by Powles et al. [34]. Tubular tower structures can have a zero wind speed contour up to five tube diameter widths long, and an area of reduced wind speeds up to ten tube

[1]OpenFAST: www.github.com/openfast
[2]Wake2e: www.wake2e.de
[3]WakeGuard: www.i17-wind.de/wakeguard
[4]WAT: www.wasp.dk/wat

widths long and four wide. At the edges of the tubular tower, increased wind speeds were also observed. If another wind turbine lies in the wake, its blades will periodically be subjected to varying considerable wind forces, causing fatigue through bending.

2.2.4 Other efficiency-reducing effects

There are even more effects at play that can influence the wind turbines in a wind farm. One of them is the torque oscillation induced by wind shear. As Dolan et al. [17] explain, the wind speed generally increases with height above the ground, and this variation is called wind shear. The wind shear follows the wind profile power law for structures that have a height of about 50 meters or more [31]. They observed the highest torque when a blade vertically points up, so there are three wind-shear-induced torque oscillations per rotor rotation, which results in an efficiency reduction of about 1%. The towers' wind shadow also generates zones of high and low air pressure. Even for upwind rotors, the torque falls considerably when a blade is in front of the tower due to the wind being redirected around the tower. The effect is most significant when the rotor is downwind from the tower. That is why today's wind turbines predominantly have upwind rotors.

2.2.5 Turbulence intensity models

Overall, the most important parameter regarding the wind farm's inner turbulences is the effective turbulence intensity. It is the ambient turbulence intensity - which is influenced by the local wind conditions and topographical complexity - combined with the added wake turbulence intensity, the turbulence generated by other nearby wind turbines. Since DIN EN IEC 61400-1:2019 [6] leaves open how the effective turbulence intensity should be computed, we will now take a look at several possible models.

Amplified wake model DIN EN IEC 61400-1:2019 defines the overall conditions which need to be fulfilled by the effective turbulence intensity model, and provides a reference model. Since the reference model assumes simplifications, e.g., that a wake exposure from a neighboring wind turbine occurs exactly 6%, it does not give the best estimation.

Frandsen Model The Frandsen model [37] is also mentioned in the norm. It yields a better estimation since it models the wake exposure by using wind data.

TurbOPark Model The TurbOPark model by Pedersen et al. [30] is the most recent and most sophisticated model to date. It assumes a Gaussian wake profile intensity distribution and a non-linear wake expansion.

3 Site suitability based on wind and turbulence

Wind turbines are highly complex machines, which are subject to strict regulations to ensure maximum safety. To assess the site suitability of a wind turbine, it is required to conduct an analysis of the site's wind conditions, physical characteristics, and other factors that may affect the turbine. Similarly, the properties of the wind turbines also have to be analyzed. The assessment typically includes:

- Collecting and analyzing wind speed and direction data at the site to determine the wind's average speed, turbulence, and other characteristics.
- Conducting a geo-environmental survey to assess the soil conditions, slope stability, weather, temperature, humidity, earthquake risk, and other factors that may affect the foundation design and stability of the wind turbine.
- Considering the local topography, vegetation, and other obstacles that may affect the wind flow and impact the wind turbine.
- Evaluating the availability of infrastructure such as roads and electrical grid capacity, as well as the potential impacts on the local community and environment.
- Considering all relevant regulations, zoning laws, and other legal issues that may affect the site suitability

Particularly interesting for the scope of this work are the wind conditions at a site. A turbine needs to withstand the wind conditions of a site, under sustained as well as extreme wind conditions. Extreme wind conditions can compromise the safety and integrity of wind turbines. Excessive turbulence generated by nearby wind turbines or complex topography can also pose a risk to the structural longevity and integrity of a wind turbine.

We will focus on determining if a specific turbine can be placed at a specific location according to local laws, considering that the turbine's structural integrity should not be compromised by the site's wind conditions and the turbulences generated by other nearby turbines and the terrain. The required steps to take this are:

1. Collect the required topography data, wind measurements and turbine specifications.
2. Model relevant site parameters using the collected data and applicable norms.
3. Model relevant turbine parameters using the design specifications and applicable norms.
4. Use modeled site and turbine parameters to assess the site suitability based on the site's wind and terrain data under consideration of the applicable norms' models.

Applicable norms In Germany, the "Richtlinie für Windenergieanlagen, DIBt 2015" [3] - the German policy for wind turbines - is the norm that provides the underlying regulations to determine the site suitability for a wind turbine. More accurately, it defines a simplified site suitability model for non-complex sites, and additionally provides several conditions under which simplified wind and turbulence conditions can be assumed.

The European norm DIN EN 1991-1-4 [4] and the national appendix [5] are also important, as it defines required concepts, estimations, and site parameter models.

DIN EN IEC 61400-1 [6] is also referenced frequently in the German policy for wind turbines, so we will also describe the applicable sections of it here. Most importantly, it contains the site suitability model which is used for complex sites, where the simplified model from the DIBt norm cannot be applied, the algorithm for determining a site's complexity, and the classification method for wind turbines.

Before going into the details, it is worth noting that we have considered building a globally applicable model using the international norm DIN EN IEC 61400-1 [6], but the norm by itself is not applicable by itself in almost every country, since they define their on adaptions on top of DIN EN IC 614000-1, similar to the German policy for wind turbines. That is why we focus specifically on wind farms that are planned on German soil. Another noteworthy fact is that the German policy for wind turbines also allows using DIN EN IEC 614000-2 for small wind turbines with a rotor radius of less than 7.98 meters. But again, the main issue with advancing renewable wind energy is the long planning process of classical large-scale wind farms, so our focus is on those.

3.1 Terms, definitions, and concepts

The most important terms and their definitions can be found in Table 1. The following subsections define concepts which are important for later use.

3.1.1 Notation

We use letters in the superscript of parameters to denote to which concept they relate. For example the turbine's design extreme wind speed function by height is denoted as $v^T_{m50}(z)$, while the turbine placement's site modeled extreme wind speed function by height is denoted as $v^S_{m50}(z)$.

3.1.2 Site

Generally speaking, a site is a precise geographic location where a wind turbine of a certain type is planned to be constructed. It is associated with the local wind data, and the surrounding terrain data points. The terrain data of the site is always dependent on the turbine type that should be built at the site, since we need data points for a disc with a radius of 20 rotor diameters around the site to be able to compute the site suitability.

Term	Definition
$v_{m50}(z)$	The function that defines the highest 10-minute average wind speed with a recurrence period of 50 years at height z above ground.
$v_{m1}(z)$	The function that defines the highest 10-minute average wind speed with a recurrence period of 1 year at height z above ground.
$v_{p50}(z)$	The function that defines the highest 3-second average wind speed with a recurrence period of 50 years at height z above ground.
$v_{ave}(z)$	The 1-year average wind speed determined with data of at least 1 year, at height z above ground.
v_r	The wind speed at hub height that is reached at the wind turbine's rated power.
$v_{ref} = v_{m50}^T(z_{hub})$	The reference wind speed of a wind turbine used to determine the turbine type and class.
$v_{b,0}$	The ground value for the basis wind speed. It is the highest 10-minute average wind speed at 10 meters above ground, with a recurrence period of 50 years.
v_b	The basis wind speed adjusted for the site's wind zone.
k^S	The shape parameter of the Weibull distribution of the site's wind data.
α^S	The profile exponent of the site. It is used to extrapolate wind speeds to all heights from a measurement at fixed height.
z_{hub}	The center hub height of a wind turbine above ground.
C_{CT}	The terrain complexity correction factor.
C_T	The rotor thrust coefficient of a wind turbine.
m_T	The Wöhler exponent of a wind turbine.
D_T	The rotor diameter of a wind turbine.
I_{ref}^T	The representative value of the design turbulence intensity at hub height.
$I_v(v)$	The turbulence intensity at wind speed v.
$I_{eff}^{S,T}$	The effective turbulence intensity at turbine placement site. It depends on the ambient turbulence intensity and the wake turbulence intensity generated by nearby wind turbines.
$\sigma(v)$	The standard deviation of the wind speed at a speed of v m/s. Broadly speaking, it is defined as $\sigma = I(v) \cdot v_{ave}$.

Table 1: The most important symbols with their definitions while defining the site suitability model.

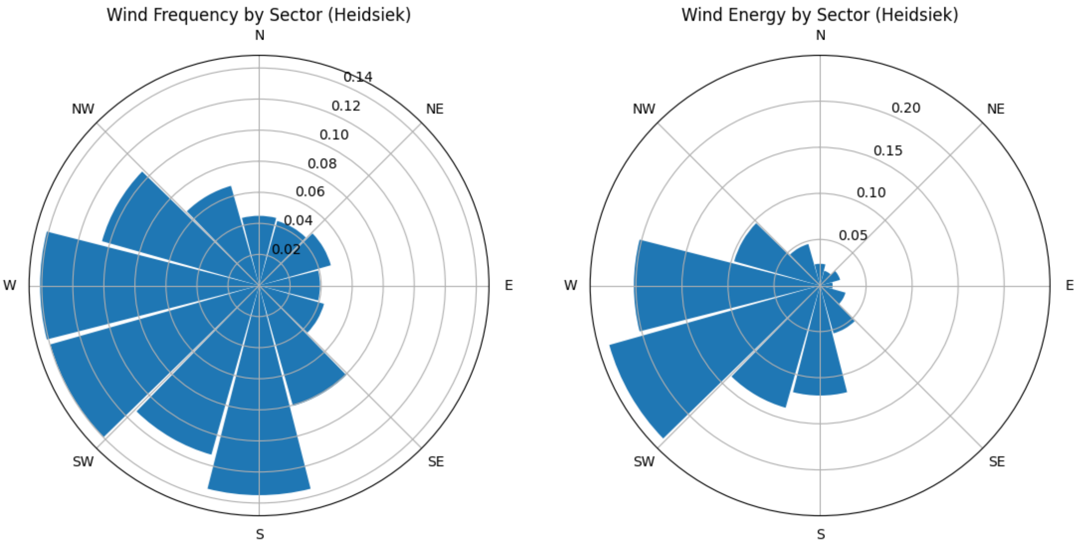

Figure 4: On the left is a wind rose based on frequency from the sectors, the right one is based on the wind energy percentage coming from the sectors. The wind data is from an example site in Heidsiek.

3.1.3 Turbulence intensity

Turbulence intensity describes the random fluctuations of wind speed. It is defined as the ratio of the standard deviation to the average wind speed calculated over averaged 10-minute intervals:

$$I_v(v) = \frac{\sigma(v)}{v_{ave}}$$

3.1.4 Wind Energy

Let's define how the energy of the wind can be calculated, which will be important to know later on. According to "The Physics of Wind Turbines" [19], the theoretical maximum harvested energy in Joules by a wind turbine at a given wind velocity of v meters per second is defined as follows:

$$E_{max}(v) = \frac{\pi \cdot R \cdot \rho \cdot v^3 \cdot t}{2}$$

where ρ is the air density in kg/m^3, and R is the radius of the wind turbine's rotor, and t is the time in seconds.

3.1.5 Wind rose

A wind rose is a graphical representation that divides the area around a point into N equally sized sectors. For example, if the sector count is 12, then the area is divided into 12 sectors with 30 degree width. The middle line of the first sector is the north line (0 degrees), and the sectors are enumerated clockwise. When wind speed and direction samples are given, a wind rose can be constructed in several ways.

Frequency wind rose One way to construct a wind rose is by the frequency of time the wind comes out of a given direction. Let S be the set of all wind samples, and for sector i, let S_i be the set of samples whose wind direction comes from sector i. The percentage of time the wind comes from sector i is then defined as:

$$f_{freq}(i) = \frac{|S_i|}{|S|}$$

Figure 4 (left) shows a wind rose based on frequency at a planned wind turbine site in Heidsiek, Germany.

Wind energy rose Another way to construct a wind rose is the wind energy rose, which shows the percentage of wind energy coming from each sector. As discussed in Section 3.1.4, the wind's energy is proportional to the wind speed cubed. So for this purpose, we can define the wind energy at a given wind speed v as v^3, since we are only interested in the fraction from the total. Let S be the set of all wind samples, and let S_i be the set of samples from sector i. Given a sample s, let $v(s)$ denote the measured wind speed of sample s in meters per second. The percentage of wind energy coming from sector i is then defined as follows:

$$f_{Energy}(i) = \frac{\sum_{s \in S_i} v(s)^3}{\sum_{s \in S} v(s)^3}$$

Figure 4 (right) shows a wind rose based on the percentage of wind energy at a planned wind turbine site in Heidsiek, Germany.

3.1.6 Weibull distribution

The Weibull distribution is a continuous probability distribution often used in engineering and data analysis. It is widely used to model wind speeds. The Weibull distribution is characterized by two parameters: the shape parameter k and the scale parameter λ. The shape parameter determines the distribution's shape, while the scale parameter determines the distribution's location. The probability density function is given by:

$$f(\lambda; k|x) = \begin{cases} \frac{k}{\lambda} \cdot (\frac{x}{\lambda})^{k-1} \cdot e^{-(\frac{k}{\lambda})^k}, & x \geq 0, k > 0, \lambda > 0 \\ 0 & \text{else} \end{cases}$$

and the cumulative distribution function is given by

$$F(x) = \begin{cases} 1 - e^{-(\frac{x}{\lambda})^k}, & x \geq 0 \\ 0 & \text{else} \end{cases}$$

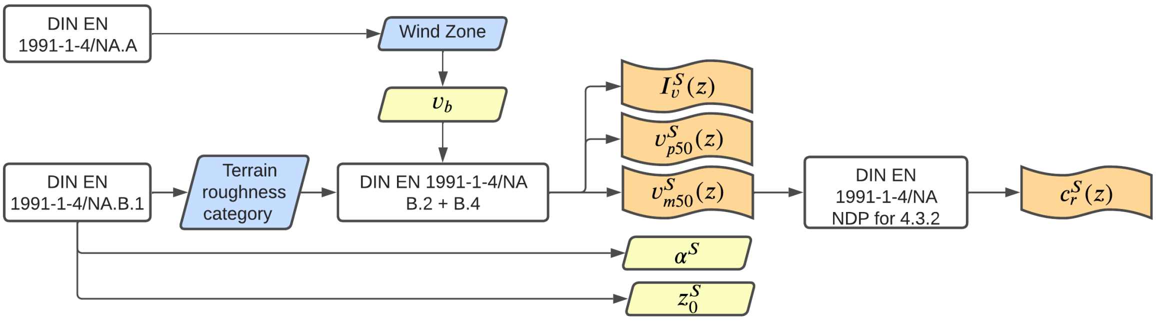

Figure 5: The site parameters which can be derived from DIN EN 1991-1-4 [4] and the national appendix [5]. Blue denotes values which have to be provided manually, yellow represents derived values, and orange represents derived functions.

where k is the shape parameter and λ is the scale parameter. Since we only need the shape parameter k later on, we will now provide a method to obtain the shape parameter. It only requires sufficient wind speed samples. The shape parameter k can be estimated iteratively with the iterative 4-maximum-likelihood method [36] using

$$k = \left(\frac{\sum_{i=1}^{N} x_i^k \cdot ln(x_i)}{\sum_{i=1}^{N} x_i^k} - \frac{\sum_{i=1}^{N} ln(x_i)}{N} \right)^{-1}$$

where N is the number of samples and x_i is the i-th sample's wind speed.

3.2 DIN EN 1991-1-4 and NA

DIN EN 1991-1-4 [4] is a European norm that provides guidance on how to analyze wind conditions for the structural design of civil engineering works that are up to 200 meters high. In conjunction, the national appendix [5] contains adjustments specific to Germany. This norm covers a wide range of topics related to wind engineering, but for determining the site suitability for wind turbines, we will focus on the parts of the norm that are relevant to this topic.

Figure 5 shows the parameters of a given site, which we will use. In Section 3.2.1, we will show how to derive the wind zone of a given site. Section 3.2.2 helps us to determine the sit's terrain roughness category. Using the wind zone and the terrain roughness category, we can obtain the site's turbulence intensity function $I_v^S(z)$ and the extreme 10-minute-averaged wind speed function with a recurrence period of 50 years, denoted as $v_{m50}^S(z)$, both depending on the height in meters z above ground. The terrain roughness factor $c_r^S(z)$ can then be directly derived from $v_{m50}^S(z)$ as shown in Section 3.2.3.

(a)

Wind zone	$v_{b,0}$
1	22.5 m/s
2	25.0 m/s
3	27.5 m/s
4	30.0 m/s

(b)

Figure 6: This figure shows the wind zones of Germany on a map (a), and allows us to get the value of $v_{b,0}$ for each wind zone (b).

3.2.1 Wind zone

The wind zone of a site can be determined according to DIN EN 1991-1-4 [4] and DIN EN 1991-1-4/NA A [5]. Figure 6 (a) shows the four different wind zones in Germany on a map. The norm does not specify a computational method for obtaining the wind zone, so it has to be determined manually. There is a website[5] which returns the wind zone for a given latitude and longitude, or another resource[6] where the wind zone can be obtained by postal code. If a wind turbine is placed closely to a border between wind zones, the higher wind zone should be used.

Once the wind zone is determined, the value of the basis wind speed $v_{b,0}$ can be obtained from Figure 6 (b). The parameter $v_{b,0}$ is a representative value for the highest 10-minute average wind speed at 10 meters above ground with a recurrence period of 50 years, which will be used later on.

3.2.2 Turbulence intensity and wind speed by terrain roughness categories

The site's turbulence intensity $I_v^S(z)$ and the highest 10-minute average wind speed with a recurrence of 50 years $v_{m50}^S(z)$ at a given height both are both defined in relation to the surrounding terrain roughness and wind zone. Four regular terrain roughness

[5] www.dlubal.com/en/load-zones-for-snow-wind-earthquake/wind-din-en-1991-1-4.html
[6] www.dehn.de/de/windzone#windzone-plz

categories and two additional ones for Germany (coastal and "binnenland") have been defined, as seen in Table 2.

Roughness category	Description	z_0	α
1	Open sea, lakes, or areas with small vegetation, without obstacles for at least 5 kilometers in wind direction.	$0.01m$	0.12
"1.5" coastal	A special category defined just for Germany, which sits in between category 1 and 2 and describes coastal regions	$0.03m^*$	0.14^*
2	Areas with small vegetation like grass and scattered obstacles like trees and building that are more than 20 times their height apart from each other, for example farming land.	$0.05m$	0.16
"2.5" binnenland	A special category defined just for Germany, which sits in between category 2 and 3 and describes the "Binnenland" found in large areas or northern Germany	$0.10m^*$	0.18^*
3	Areas with uniform vegetation or buildings with distances less than 20 times their height between them, like villages, forests and industrial parks.	$0.30m$	0.22
4	Areas where more than 15% of the surface is covered with buildings, which are taller than $15m$ on average	$1.05m$	0.30

Table 2: The terrain roughness categories defined in DIN EN 1991-1-1-4/NA [5]. Here, z_0 specifies the roughness length, and α is the profile exponent, which are both derived from the terrain roughness category. (*) Estimated since the norm does not provide these values.

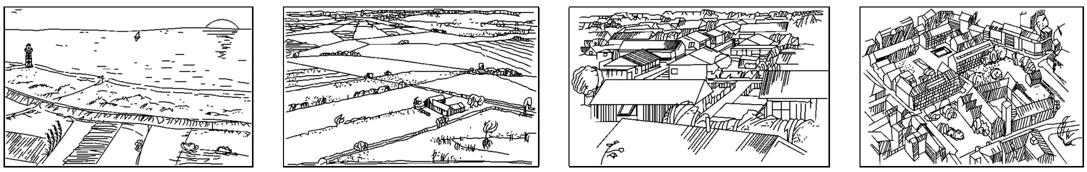

Figure 7: The terrain roughness categories 1, 2, 3 and 4 from left to right, as depicted in DIN EN 1991-1-4 [4].

There is no computational method specified to determine the terrain category, but the norm's national appendix [5] provides Table 2, so it has to be used to estimate the terrain roughness category. Additionally, Figure 7 can be consulted for selecting the appropriate terrain roughness category. In cases of uncertainty, the smoother terrain roughness category should always be chosen. Once the terrain roughness category has been determined, the functions for $I_v^S(z)$, $v_{m50}^S(z)$ and $v_{p50}^S(z)$ can be modeled as follows for each terrain roughness category:

For terrain roughness category 1:

$$v_{m50}(z) = \begin{cases} 1.18 v_b \cdot (\frac{z}{10})^{\alpha}, & \text{if } z > 2 \\ 0.97 v_b, & \text{otherwise} \end{cases}$$

$$v_{p50}(z) = \begin{cases} 1.61 v_b \cdot (\frac{z}{10})^{0.095}, & \text{if } z > 2 \\ 1.38 v_b, & \text{otherwise} \end{cases}$$

$$I_v(z) = \begin{cases} 0.14 \cdot (\frac{z}{10})^{-\alpha}, & \text{if } z > 2 \\ 0.17, & \text{otherwise} \end{cases}$$

For terrain roughness category "1.5" (coastal regions):

$$v_{m50}(z) = \begin{cases} 1.18 v_b \cdot (\frac{z}{10})^{\alpha}, \alpha = 0.12, & \text{if } 50 < z < 300 \\ 1.10 v_b \cdot (\frac{z}{10})^{\alpha}, \alpha = 0.165, & \text{if } 4 < z < 50 \\ 0.95 v_b, & \text{if } z \leq 4 \end{cases}$$

$$v_{p50}(z) = \begin{cases} 1.61 v_b \cdot (\frac{z}{10})^{0.095}, & \text{if } 50 < z < 300 \\ 1.51 v_b \cdot (\frac{z}{10})^{0.135}, & \text{if } 4 < z < 50 \\ 1.33 v_b, & \text{if } z \leq 4 \end{cases}$$

$$I_v(z) = \begin{cases} 0.14 \cdot (\frac{z}{10})^{-\alpha}, \alpha = 0.12, & \text{if } 50 < z < 300 \\ 0.15 \cdot (\frac{z}{10})^{-\alpha}, \alpha = 0.165, & \text{if } 4 < z < 50 \\ 0.17, & \text{if } z \leq 4 \end{cases}$$

For terrain roughness category 2:

$$v_{m50}(z) = \begin{cases} 1.00 v_b \cdot (\frac{z}{10})^{\alpha}, & \text{if } z > 4 \\ 0.86 v_b, & \text{otherwise} \end{cases}$$

$$v_{p50}(z) = \begin{cases} 1.45 v_b \cdot (\frac{z}{10})^{0.120}, & \text{if } z > 4 \\ 1.30 v_b, & \text{otherwise} \end{cases}$$

$$I_v(z) = \begin{cases} 0.19 \cdot (\frac{z}{10})^{-\alpha}, & \text{if } z > 4 \\ 0.22, & \text{otherwise} \end{cases}$$

For terrain roughness category "2.5" (Binnenland):

$$v_{m50}(z) = \begin{cases} 1.00v_b \cdot (\frac{z}{10})^{\alpha}, \alpha = 0.16, & \text{if } 50 < z < 300 \\ 0.86v_b \cdot (\frac{z}{10})^{\alpha}, \alpha = 0.25, & \text{if } 7 < z < 50 \\ 0.79v_b, & \text{if } z \leq 7 \end{cases}$$

$$v_{p50}(z) = \begin{cases} 1.45v_b \cdot (\frac{z}{10})^{0.120}, & \text{if } 50 < z < 300 \\ 1.31v_b \cdot (\frac{z}{10})^{0.185}, & \text{if } 7 < z < 50 \\ 1.23v_b, & \text{if } z \leq 7 \end{cases}$$

$$I_v(z) = \begin{cases} 0.19 \cdot (\frac{z}{10})^{-\alpha}, \alpha = 0.16, & \text{if } 50 < z < 300 \\ 0.22 \cdot (\frac{z}{10})^{-\alpha}, \alpha = 0.25, & \text{if } 7 < z < 50 \\ 0.24, & \text{if } z \leq 7 \end{cases}$$

For terrain roughness category 3:

$$v_{m50}(z) = \begin{cases} 0.77v_b \cdot (\frac{z}{10})^{\alpha}, & \text{if } z > 8 \\ 0.73v_b, & \text{otherwise} \end{cases}$$

$$v_{p50}(z) = \begin{cases} 1.27v_b \cdot (\frac{z}{10})^{0.155}, & \text{if } z > 8 \\ 1.23v_b, & \text{otherwise} \end{cases}$$

$$I_v(z) = \begin{cases} 0.28 \cdot (\frac{z}{10})^{-\alpha}, & \text{if } z > 8 \\ 0.29, & \text{otherwise} \end{cases}$$

For terrain roughness category 4:

$$v_{m50}(z) = \begin{cases} 0.56v_b \cdot (\frac{z}{10})^{\alpha}, & \text{if } z > 16 \\ 0.64v_b, & \text{otherwise} \end{cases}$$

$$v_{p50}(z) = \begin{cases} 1.05v_b \cdot (\frac{z}{10})^{0.200}, & \text{if } z > 16 \\ 1.15v_b, & \text{otherwise} \end{cases}$$

$$I_v(z) = \begin{cases} 0.43 \cdot (\frac{z}{10})^{-\alpha}, & \text{if } z > 16 \\ 0.37, & \text{otherwise} \end{cases}$$

The value of α can be determined from Table 2, and v_b, the base wind speed, can be determined from the wind zone's $v_{b,0}$, which we already defined in Section 3.2.1. The calculation of v_b can follow custom rules, but the norm otherwise recommends setting it equal to $v_{b,0}$. Since the national appendix does not specify any custom rules for determining v_b, we can from now on assume that:

$$v_b = v_{b,0}$$

3.2.3 Terrain roughness coefficient

The terrain roughness coefficient accounts for the variability of the mean wind speed at the site due to the height above ground and the surrounding terrain's roughness category. The national appendix DIN EN 1991-1-4/NA [5] dictates the following function for Germany:

$$c_r^S(z) = \frac{v_{m50}^S(z)}{v_{m50}^S(10)}$$

3.3 DIN EN IEC 61400-1:2019

The German policy for wind turbines [3] is heavily based on DIN EN IEC 61400-1 [6], so we will present this norm's applicable parts in detail now.

It should be noted that the German norm permits to apply either DIN EN IEC 61400-1:2005-12 or DIN EN IEC 61400-1:2011-08, but mixing them is prohibited. However, since the release of the German norm in October 2015, a newer version, namely DIN EN IEC 61400-1:2019, has been released. In an official press release by DIBt[7], the application of DIN EN IEC 61400-1:2019 to the German policy for wind turbines [3] has been recommended without restrictions. To be up-to-date with the latest regulations, we will exclusively refer to the 2019 edition.

This international norm specifies a broad range of regulations and models, but to determine the site suitability of a wind farm, the following parts are the most important:

1. **Site parameters**: All important site parameters which can are derived from this norm can be seen in Figure 8. From the site's topographic data and the wind energy distribution $f_{energy}(i)$ across sectors, a complexity rating is computed as described in Section 3.3.6. The computation also yields a terrain variation indices and terrain slope indices. Using the complexity value, a turbulence structure correction factor called C_{CT}^S is introduced in Section 3.3.7. Furthermore, the site's air density ρ^S and the shape parameter k^S of a Weibull distribution fitted to the wind data are required, and the average wind speed function $v_{ave}^S(z)$ can be estimated using the wind profile power law [38].

2. **Turbine parameters**: Depending on the design turbulence category and design wind speed class, the turbine's design average wind speed at hub height $v_{ave}^T(z_{hub})$, the turbine's design 50-year extreme wind speed at hub height $v_{m50}^T(z_{hub})$, which is also sometimes called v_{ref}^T, and the turbines reference turbulence intensity I_{ref}^T can be determined, as seen in Figure 9, which will be described in Section 3.3.1. From those values, the design extreme wind speed function $v_{m50}^T(z)$ is modeled through the extreme wind model (EWM) in Section 3.3.2, the turbulence intensity standard deviations are modeled by both the normal and extreme turbulence

[7] www.dke.de/de/arbeitsfelder/energy/normenhinweise/verwendung-din-en-61400-1-mit-dibt-richtlinie-windenergieanlagen

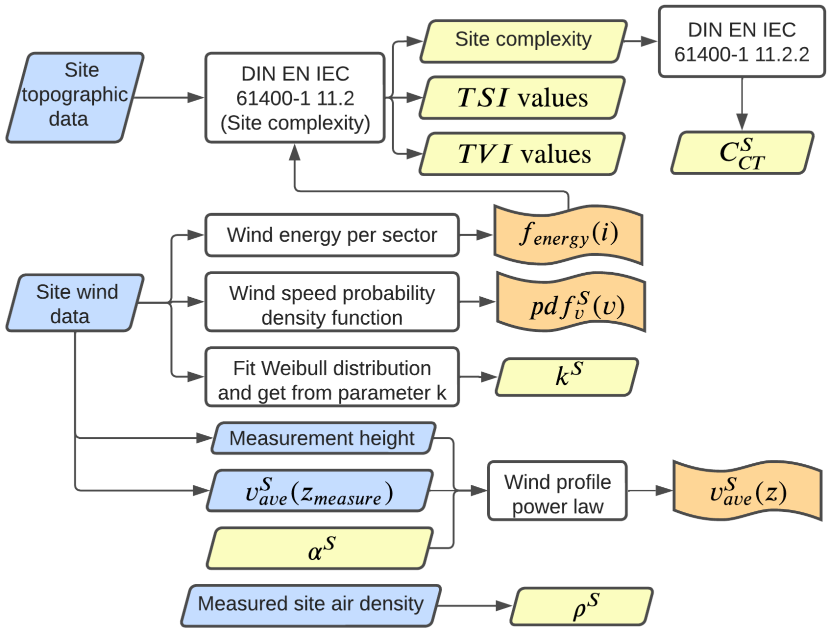

Figure 8: The data flowchart for determining all relevant site parameters in DIN EN IEC 61400-1. Blue boxes represent manual inputs, yellow ones show derived constants, and the orange ones are derived functions.

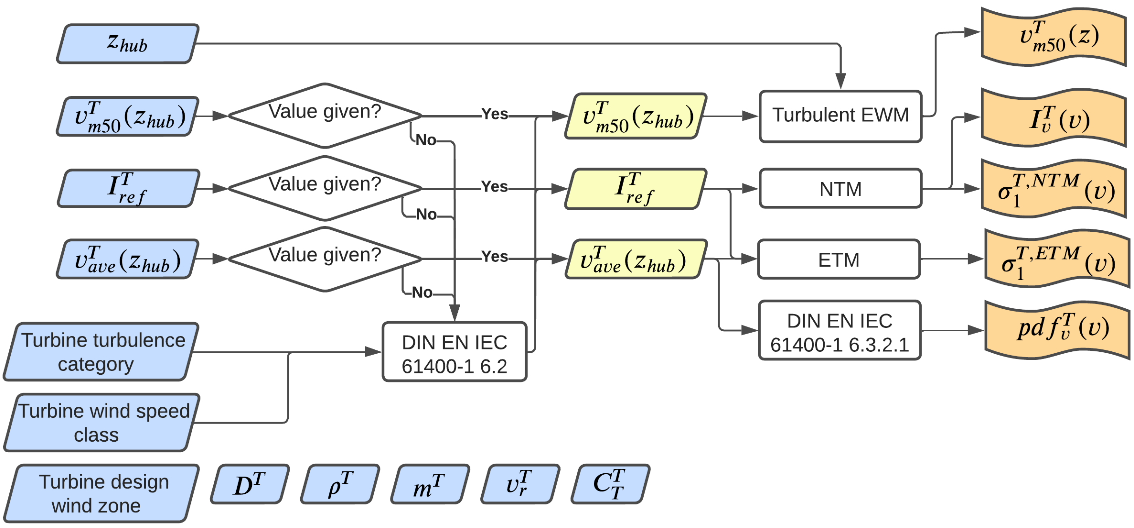

Figure 9: The data flowchart for determining all relevant turbine parameters.

Design wind speed class	$v_{ave}^T(z_{hub})$	$v_{m50}^T(z_{hub})$
I	10.0m/s	50.0m/s
II	8.5m/s	42.5m/s
III	7.5m/s	37.5m/s
S	custom	custom

Table 3: There are three common design wind speed classes for wind turbines. An extra class S exists, where the turbine designer can specify custom values for the design average and extreme wind speed at hub height, as well as the design reference turbulence intensity.

Design turbulence category	I_{ref}^T
A+	0.18
A	0.16
B	0.14
C	0.12
S	custom

Table 4: Wind turbines with design turbulence category A+ can withstand the highest turbulences, while category C turbines withstand the least turbulence.

model in Sections 3.3.3 and 3.3.4. Furthermore, the design probability density function can be obtained from Section 3.3.5. For the later site suitability model, we also need the turbine's rotor diameter D^T, the design air density ρ^T, the optimal wind speed for the rated power v_r^T, a and the optionally the rotor thrust coefficient C_T^T as well.

3. **Turbulence intensity**: In both the site suitability model for complex and non-complex sites, we need to calculate the effective turbulence intensity at a given site. Section 3.3.8 describes how it is done.

4. **Site suitability model for complex sites**: In Germany, the site suitability model from this norm needs to be applied if the site is rated as complex. Section 3.3.9 will introduce this model.

3.3.1 Wind turbine design class and turbulence category

We classify wind turbines by their design wind speed class, and their design turbulence category in accordance with DIN EN IEC 61400-1 6.2 [6]. Table 3 shows the design wind speed classes, and Table 4 lists the possible design wind turbine turbulence categories.

The overall class of a wind turbine is derived from both the turbulence category and wind class, so a turbine with wind class II and turbulence category C is denoted as II_C. It should be noted that the standard classes don't apply to offshore wind farms.

However, they can be adapted for tropical storm regions, but this adaption will not be necessary for Germany.

3.3.2 The extreme wind model (EWM)

The extreme wind speed model gives us the design extreme wind speed function by height above the ground in meters for a given wind turbine T. The model depends solely on the turbine's reference extreme wind speed at hub height $v_{m50}^{T}(z_{hub})$, which can either be provided manually or computed from the turbine's design wind speed class as discussed in Section 3.3.1. The model is further divided into two types: The stationary extreme wind model and the turbulent extreme wind model. Only one of them can be used, they are not allowed to be mixed.

Stationary Using the stationary extreme wind speed model, we can obtain the following functions:

$$v_{p50}^{T}(z) = 1.4 \cdot v_{m50}^{T}(z_{hub}) \cdot \left(\frac{z}{z_{hub}}\right)^{0.11}$$

and

$$v_{p1}^{T}(z) = 0.8 \cdot v_{p50}^{T}(z)$$

Turbulent Using the turbulent extreme wind speed model, these functions can be obtained:

$$v_{m50}^{T}(z) = v_{m50}^{T}(z_{hub}) \cdot \left(\frac{z}{z_{hub}}\right)^{0.11}$$

and

$$v_{m1}^{T}(z) = 0.8 \cdot v_{m50}^{T}(z)$$

Furthermore, the longitudinal standard deviation of the turbulence σ_1 must be fixed as:

$$\sigma_1^{T}(v) = 0.11v$$

The German policy for wind turbines [3] later recommends using the turbulent extreme wind model.

3.3.3 The normal turbulence model (NTM)

The normal turbulence model gives us the design turbulence intensity function by wind speed v in m/s for a given wind turbine T. The model depends solely on the turbine's reference turbulence intensity at hub height I_{ref}^{T}, which can either be provided manually or computed from the wind turbine's design turbulence category as discussed in Section 3.3.1.

The representative value of the turbulence intensity standard deviation is denoted as σ_1. It is fixed as the 90% quantile of the turbulence intensity at hub height, and depends on the wind speed v at hub height. For the standard wind turbine classes, it must be determined with the following formula:

$$\sigma_1^{T,NTM}(v) = I_{ref}^T \cdot \left(\frac{3}{4} \cdot v + 5.6\right)$$

The turbine's design turbulence intensity function is given by:

$$I_v^T(v) = \frac{\sigma_1^{T,NTM}(v)}{v}$$

3.3.4 The extreme turbulence model (ETM)

The extreme turbulence model gives us the turbulence standard deviation at hub height:

$$\sigma_1^{T,ETM}(v) = 2 \cdot I_{ref}^T \cdot \left(0.072 \cdot (\frac{v_{ave}^T(z_{hub})}{2} + 3) \cdot (\frac{v}{2} - 4) + 10\right)$$

where $v_{ave}^T(z_{hub})$ and I_{ref} are determined by the wind turbine class from Section 3.3.1.

3.3.5 Turbine design wind speed probability density function

The design wind speed probability density function is later relevant for determining if the turbine is suitable for the given site's wind speed probability distribution. It is defined as:

$$pdf_v^T(v) = 1 - e^{-\pi \cdot (v/2 \cdot v_{ave}^T(z_{hub}))^2}$$

where $v_{ave}^T(z_{hub})$ is determined by the wind turbine class from Section 3.3.1.

3.3.6 Site topographical complexity

Before we can demonstrate that the structural integrity of a wind turbine will not be compromised, DIN EN IEC 61400-1 [6] first requires an assessment of the site complexity. This is important, since the more complex the surrounding terrain, the more it distorts wind and creates turbulence.

To perform the evaluation of the site's topographical complexity, the following inputs are required:

1. A grid of coordinates with elevation values within at least $20z_{hub}$ meters around the wind turbine's base. The grid's resolution must not exceed 50 meters.

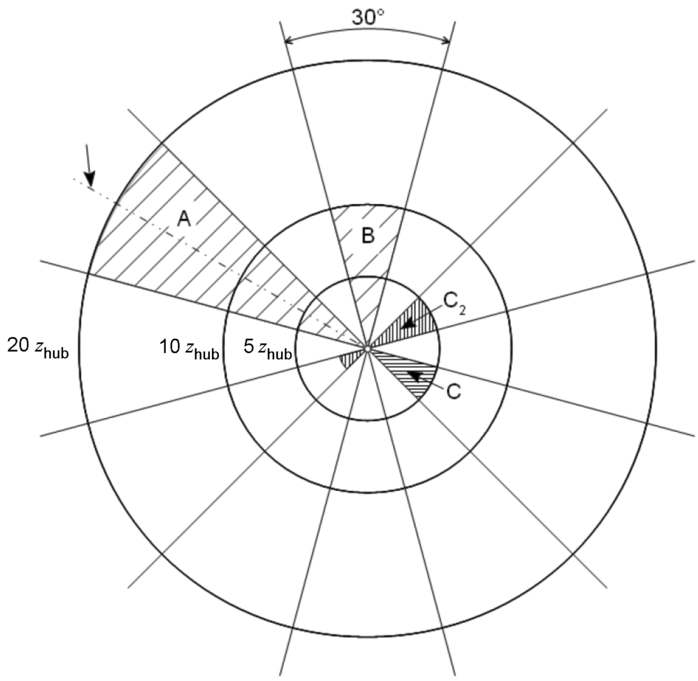

Figure 10: An example of how a site should be divided into radial sectors according to DIN EN IEC 61400-1:2019 [6]. For each of these sectors, a plane is fitted. The sectors with a radius of $5z_{hub}$ can be extended with points from the opposing sector by a range of $2z_{hub}$. For computing a plane's slope, the middle line is relevant.

2. Historic wind data for the site for at least one year. This is required to determine how much of the wind energy comes out of a specific direction.

The surface points around the turbine's base are divided into 12 sectors that are 30 degrees wide each, like seen in Figure 10. For each sector, and each of the horizontal distances $R = 5z_{hub}$, $R = 10z_{hub}$, and $R = 20z_{hub}$, a planed is now fitted to the surface points within that distance and sector. The plane is not required to pass through the turbine's base surface point. For the case of surface points that are within $R = 5z_{hub}$ of the turbine's base, the set of surface points can be extended by surface points in the opposite sector that are at most $2z_{hub}$ meters away from the turbine's base. This is visualized in Figure 10 for sector C_2. Additionally, a plane is fitted through the full disc of all points within $R = 5z_{hub}$ of the turbine's base.

For each fitted plane, we calculate the slope of the corresponding sector's middle line in degrees, or the angle of the plane's normal vector to the z axis in the case of the full disc with $R = 5z_{hub}$ radius. The slope is denoted as $\theta(i, R)$ and θ_{360} respectively. The standard deviation of the fitted plane to the terrain points is denoted as $D_{TV}(i, R)$ and D_{TV360} respectively, where the distance between terrain points and the fitted plane is measured through a vertical line. Furthermore, the percentage of wind energy from each sector i, denoted as $f_{energy}(i)$ can be computed from the site's wind data, as explained in Section 3.1.5.

The terrain slope indices $TSI_{30}(R)$ for all $R \in [5z_{hub}, 10z_{hub}, 20z_{hub}]$ and the full disc TSI_{360} are then calculated as follows:

$$TSI_{30}(R) = \sum_{i=1}^{12} f_{Energy}(i) \cdot |\theta(i,R)|$$

$$TSI_{360} = \frac{5}{3} \cdot \theta_{360}$$

Similarly, the terrain variation index for the 30 degree sectors, denoted as $TVI_{30}(R)$, and for the full disk, denoted as TVI_{360}, are defined as:

$$TVI_{30}(R) = \sum_{i=1}^{12} f_{Energy}(i) \cdot \frac{D_{TV}(i,R)}{R}$$

$$TVI_{360}(R) = \frac{D_{TV360}}{3 \cdot R}$$

Finally, a terrain complexity value is assigned for each disk radius and TSI as well as TVI value according to Table 5. The possible values are low (L), medium (M) and high (H).

Non-complex site A site is classified as non-complex if for all the four cases for TSI and all four cases of TVI in Table 5, the values are below the threshold for the low class L.

Complex site If a site does not fulfill the conditions of a non-complex site, then it is considered complex. The overall site complexity value is the highest of the eight cases from Table 5.

		TSI(R)			TVI(R)		
Radius R	Sector width	L	M	H	L	M	H
$5z_{hub}$	360°	≥ 10°	≥ 15°	≥ 20°	≥ 2%	≥ 4%	≥ 6%
$5z_{hub}$	30°	≥ 10°	≥ 15°	≥ 20°	≥ 2%	≥ 4%	≥ 6%
$10z_{hub}$	30°	≥ 10°	≥ 15°	≥ 20°	≥ 2%	≥ 4%	≥ 6%
$20z_{hub}$	30°	≥ 10°	≥ 15°	≥ 20°	≥ 2%	≥ 4%	≥ 6%

Table 5: Each row shows the conditions that need to be met for each radial fitted plane with radius r_f around the wind turbine to classify the terrain as non-complex.

	Terrain complexity			
	Not complex	L	M	H
C^S_{CT}	1.00	1.05	1.10	1.15

Table 6: The values for C^S_{CT} for each terrain complexity class.

3.3.7 Turbulence structure correction factor

The site's terrain affects the turbulence structure. That is why a turbulence structure correction factor for turbulence is introduced. Ideally, the factor should be derived from on site measurements of turbulence, however if there are no precise measurements from the site, the values from Table 6 can be used.

3.3.8 Effective turbulence intensity

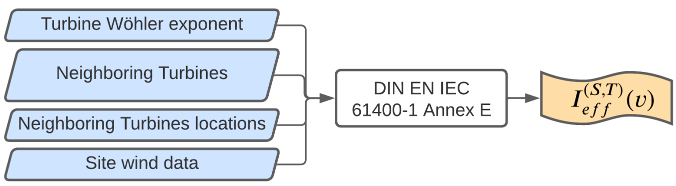

Figure 11: The input and output parameters of the effective turbulence intensity model

Both the site suitability model for complex and non-complex sites require the turbine's design turbulence intensity function $I_v^T(v)$ and the site's effective turbulence intensity function $I^S_{eff}(v)$ for a specific wind speed range. Here, we will show how to obtain it from turbine and site parameters as seen in Figure 11. For some cases, as defined later in Section 3.5.3, we can assume simplified criteria. However, when the terrain roughness is too high or the distance to neighboring turbines is too low, then the effective turbulence intensity has to be computed. According to DIN EN IEC 614000-1:2019, the effective turbulence intensity at hub height is given by:

$$I^{S,T}_{eff}(v) = \left(\int_0^{2\pi} pdf_v^S(\theta|v) \cdot I_{combined}(\theta|v)^{m^T} d\theta \right)^{\frac{1}{m^T}}$$

where $pdf_v^S(\theta|v)$ is the probability density function for wind speed v from direction θ, and $I_{combined}(\theta|v)$ is the combined turbulence intensity at wind speed v in direction θ. the Wöhler exponent of the wind turbine in denoted as m^T, and is determined by the weakest material used in the wind turbine. Typical values range from 6 for steel, 10 for glass and 15 for carbon-fiber [11].

Since the norms allow us to inspect the wind directions as bins with 30° width, we can discretize the formula as follows:

$$I_{eff}^{S,T}(v) = \left(\sum_{i=1}^{12} pdf_v^S(i|v) \cdot I_{combined}(i|v)^{m^T} \right)^{\frac{1}{m^T}}$$

where $pdf_v^S(i|v)$ and $I_{combined}(i|v)$ now depend on the sector id i.

The combined turbulence intensity is the value of the site's ambient turbulence intensity and the turbulence intensity generated by other nearby wind turbines. The norm does not specify how exactly this has to be done, but demands that it has to follow the formulas from above.

Method of amplified wake One method defined in Annex E of DIN EN IEC 61400-1 [6] is the method of amplified wake. It considers turbines within a distance of at most $10D$ where D is the rotor diameter of the currently considered turbine T. If no other turbine is within $10D$, then is simply

$$I_{eff}^{S,T}(v) = C_{CT}^S \cdot I_v^S(v)$$

which is the ambient turbulence intensity times the turbulence structure correction factor.

Otherwise, let $S_{<10D}$ be the set of turbines within that radius. The method of amplified wake defines the effective turbulence intensity in this case as:

$$\hat{\sigma}_{eff}(v) = \left((1 - 0.06 \cdot |S_{<10D}|) \cdot \hat{\sigma}_{rep}(v) + 0.06 \cdot \sum_{t \in S_{<10D}} \hat{\sigma}_t(d(T_j, T), v)^m \right)^{\frac{1}{m^T}}$$

$$\hat{\sigma}_t(d, v) = \sqrt{\frac{v^2}{(1.5 + \frac{0.8 \cdot d}{\sqrt{C_T^T(v)}})^2} + \hat{\sigma}_{rep}(v)^2}$$

$$\hat{\sigma}_{rep}(v) = v \cdot \left(\frac{3v + 15}{3v} + 1.28 \cdot \frac{5.76}{3v} \right) \cdot I_{ambient}(v)$$

$$I_{ambient}(v) = \frac{\sigma_1^S(v)}{v}$$

$$I_{eff}^{S,T}(v) = C_{CT}^S \cdot \frac{\hat{\sigma}_{eff}(v)}{v}$$

where $C_T^T(v)$ is the rotor thrust coefficient of the wake-producing neighboring turbine. If it is unknown, it can be substituted with $7/v$.

Note how this method assumes that a wake exposure generally happens exactly 6% or the time. Furthermore, a uniform probability distribution of wind speeds over the sectors is assumed.

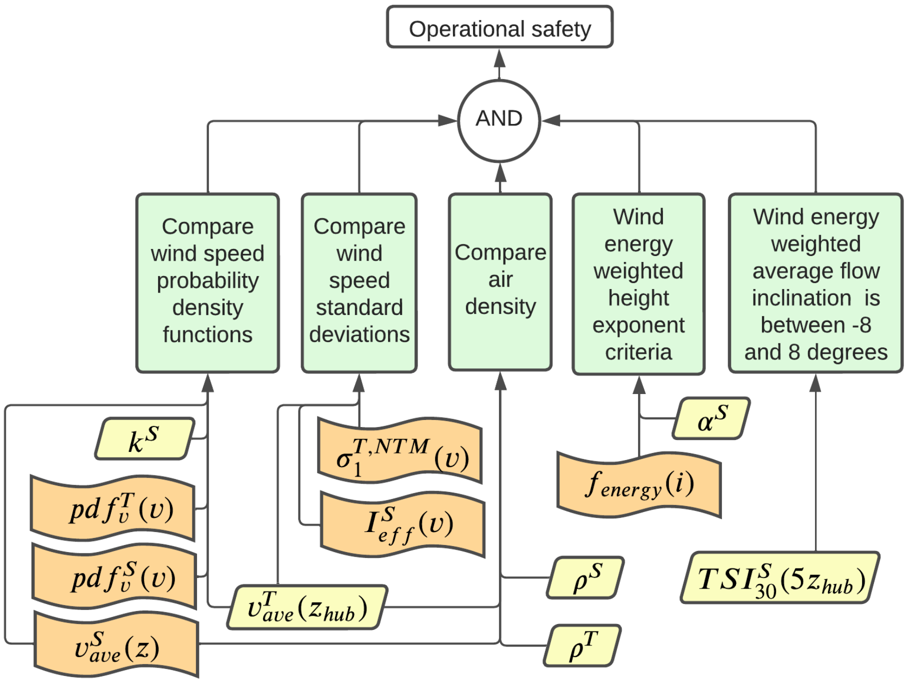

Figure 12: The flowchart for determining the operational safety of a turbine placed at a complex site.

3.3.9 Site suitability for complex sites

If the site is deemed to be complex according to Section 3.3.6, the full site suitability model for complex sites has to be applied. It consists of two parts: The operational safety, and the structural integrity.

Operational safety To ensure the operational safety of a wind turbine at a complex site, the following requirements as seen in Figure 12 must be met:

1. **Compare wind speed probability densities**: For the site's wind speed probability density function and the turbine's design probability density function, it holds that:
$$pdf_v^S(v) \leq pdf_v^T(v) \ , \forall v \in [v_{ave}^S(z_{hub}), 2v_{ave}^S(z_{hub})]$$
If the Weibull distribution shape parameter k^S is greater or equal to 1.4, then additionally, it must hold that:
$$\frac{6.5 \cdot v_{ave}^S(z_{hub})}{v_{ave}^T(z_{hub})} - 4.5 \leq k \leq \frac{-6 \cdot v_{ave}^S(z_{hub})}{v_{ave}^T(z_{hub})} + 8$$

2. **Compare wind speed standard deviations:** Using the effective turbulence intensity $I_{eff}^S(v)$, we can obtain the effective standard deviation of the wind speed as
$$\hat{\sigma}_{eff}(v) = I_{eff}^S(v) \cdot v$$

Together with the standard deviation of the turbine's design wind speed obtained through the NTM model denoted as $\sigma_1^{T,NTM}(v)$, we must prove that:

$$\sigma_1^{T,NTM}(v) > \hat{\sigma}_{eff}(v) \ , \forall v \in [v_{ave}^S(z_{hub}), 2v_{ave}^S(z_{hub})]$$

3. **Airflow inclination:** The airflow inclination of a specific 30° sector can be assumed to be equal to the slope of the fitted plane to the given sector with $5z_{hub}$ radius (extended by $2z_{hub}$ into the opposing sector) as defined in Section 3.3.6. It must hold that:
$$TSI_{30}^S(5z_{hub}) \leq 8$$

4. **Height exponent:** For all wind direction 30° sectors, the site's wind energy weighted height exponent must fulfill:

$$0.05 \leq \alpha_w^S \leq 0.25$$

For sites that are modeled after their terrain roughness category as described in Section 3.2.2, the height exponent is the same from every direction, so $\alpha_w^S = \alpha^S$.

5. **Air density:** For the air density, the following inequality must be true:

$$\rho^T \cdot (v_{ave}^T(z_{hub}))^2 \geq \rho^S \cdot (v_{ave}^S(z_{hub}))^2$$

Structural integrity Additional to the operational safety, we must also determine the structural integrity as seen in Figure 13 with the following criteria:

1. **Compare wind speed standard deviation:** The standard deviation of the turbine's design wind speed obtained through the NTM model denoted as $\sigma_1(v)$ must fulfill:
$$\sigma_1(v) \geq \hat{\sigma} + 1.28\hat{\sigma}_\sigma \forall v \in [0.6v_r^T, 1.6v_r^T]$$
where v_r^T is the wind turbine's rated wind speed.

2. **Extreme wind speed:** It must hold that:

$$v_{m50}^S(z_{hub}) \leq v_{m50}^T(z_{hub})$$

Additionally, it must hold that:

$$\rho^T \cdot (v_{m50}^T(z_{hub}))^2 \geq \rho^S \cdot (v_{m50}^S(z_{hub}))^2$$

3. **Extreme turbulence and ETM comparison:** It must be proven that the maximum value of the wind speed standard deviation at the site does not exceed the value of the turbine's ETM model, which is denoted as $\sigma_1^{T,ETM}(z_{hub})$.

4. **Wake turbulence and ETM comparison:** In a wake situation, the maximum value of the standard deviation of the wind speed in the middle of the wake does not exceed the ETM model in the most affected wind sector.

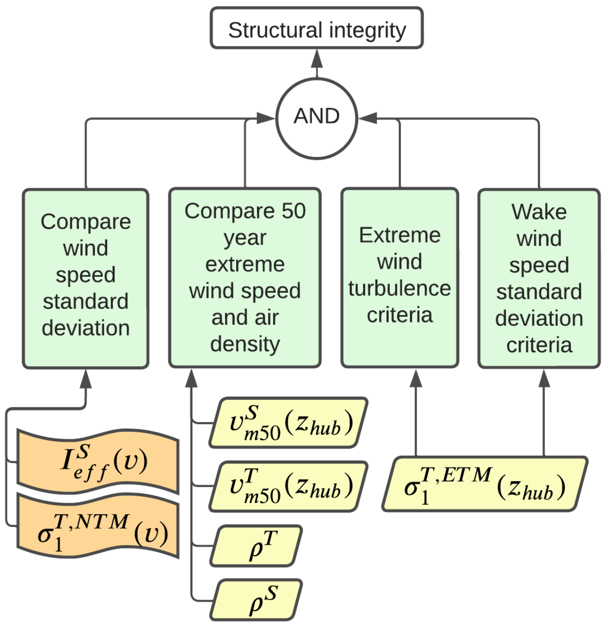

Figure 13: The flowchart for determining the structural integrity of a turbine placed at a complex site.

3.4 Other effective turbulence models

As we have seen in Section 3.3.8 the effective turbulence intensity is the most important factor for determining if a turbine can withstand the site's local turbulences. In general, the effective turbulence intensity at a site depends on the turbine's Wöhler coefficient, the ambient turbulence intensity, the added wake turbulence intensity of upwind turbines and the wind distribution. We have seen the method of amplified wake in DIN EN IEC 61400-1 [6]. But it has clear limitations, since it assumes a uniform wind distribution over all sectors, and a fixed wake exposure frequency of 6%. So in this section we will consider more sophisticated approaches.

3.4.1 The Frandsen model

The turbulence model by Frandsen [18] uses the probability distribution of wind data, and uses it to more accurately estimate wake exposure effects.

For a wind turbine T, let $S^T_{<10D}(i)$ be the set of turbines in upwind sector i of turbine T within a range of 10 rotor diameters. Then, we can compute the combined turbulence intensity as:

$$I^T_{combined}(i|v) = \sqrt{I^S_{rep}(v)^2 + \sum_{T_j \in S^T_{<10D}(i)} I^T_{add}(T_j|v)^2}$$

where

$$I_{rep}(v) = I_{mean}(v) + 1.28 \cdot I_{stddev}(v)$$

$$I_{mean}(v) = \frac{3v + 15}{3v} \cdot I_v^S(z_{hub}^T)$$

$$I_{stddev}(v) = \frac{1.92}{v} \cdot I_v^S(z_{hub}^T)$$

and

$$I_{add}^T(T_j|v) = \sqrt{\frac{0.9}{(1.5 + 0.3 \cdot d_{T,T_i} \cdot \sqrt{v})^2}}$$

Here, d_{T,T_i} is the distance of turbine T and T_j divided by the rotor diameter of turbine T, denoted as D^T.

With the Frandsen model, a wake exposure is considered if the upwind turbine is inside the currently analyzed sector within the given maximum distance, and the percentage of exposure time depends on the wind data. The overall $I_{eff}^{(S,T)}(v)$ can finally be derived from the discretized formula from Section 3.3.8.

3.5 German guideline for wind turbines by DIBt

The "Richtline für Windenergieanlagen" by DIBt [3] contains the main requirements regarding the construction of wind turbines, which apply if the planned site is on German soil. It spans from criteria for earthquake-zones, construction procedures and details, design criteria, criteria for the electrical grid, and site wind and turbulence criteria. All those criteria together are used to assess the site suitability of a wind turbine. We will go into detail for chapter 7 of the German policy for wind turbines [3], which describes simplified formulas that can be applied under certain conditions, and chapter 16.2 which describes a simplified procedure of determining the site suitability of wind turbines.

Figure 14 shows the assumptions we can make for the simplified site suitability model. In Section 3.5.1, we will show which functions can be used for $I_v^S(z)$, $v_{m50}^S(z)$ and $v_{m1}^S(z)$ for lower terrain roughness categories. Section 3.5.2 will define the average wind speed function, and Section 3.5.4 gives a standard value for the air density.

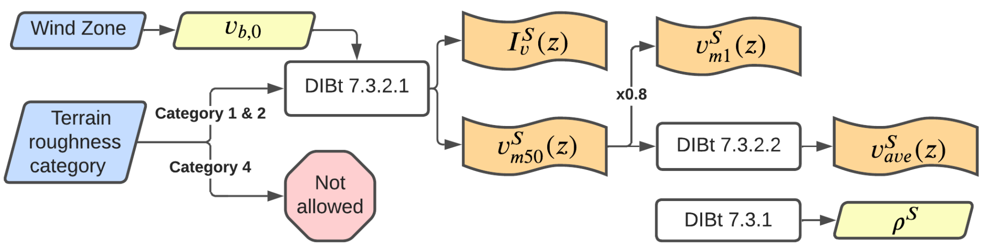

Figure 14: The site parameter which can be assumed for the simplified site suitability model. Blue denotes values which have to be provided manually, yellow represents derived values, and orange represents derived functions.

3.5.1 Simplified turbulence intensity and extreme wind

The German policy for wind turbines [3] defines conditions under which the turbulence intensity function $I_v^S(z)$ and the function for the highest 10-minute average wind speed with a recurrence of 50 years $v_{m50}^S(z)$ can be estimated by the provided formulas. To obtain the estimated functions, we first have to determine the site's terrain roughness category according to Section 3.2.2, as well as the wind zone according to Section 3.2.1.

For terrains of category 1 and 2, the norm leaves us the option to use the following functions $I_v^S(z)$ and $v_{m50}^S(z)$:

$$I_v^S(z) = 0.128 \cdot (\frac{z}{10})^{-0.05}$$
$$v_{m50}(z) = 1.15 \cdot v_{b,0} \cdot (\frac{z}{10})^{0.121}$$
$$v_{m1}(z) = 0.8 \cdot v_{m50}(z)$$

where z is the height above ground, and $v_{b,0}$ is the basis wind speed.

If the terrain roughness category is 4, then the whole German policy for wind turbines [3] is not applicable anymore, and the site suitability model for complex sites from DIN EN IEC 61400-1 [6] has to be used instead.

Figure 15 compares the simplified $v_{m50}^S(z)$ and $I_v^S(z)$ with those of DIN EN 1991-1. Since the simplified $v_{m50}(z)$ gives a lower estimate only in case of the terrain roughness category of I or "coastal", we will only use it in these cases, and fallback to the estimations from DIN EN 1991-1 in all other cases. The same applies for $v_{m1}(z)$. The turbulence intensity is lower in the simplifications for all height for terrain roughness category II, but only up to above 30 meters above ground for I and "coastal". So, we are only going to use the simplified function for terrain roughness category II when applying the simplified DIBt site suitability model. If the wind farm planner provides their own functions, those will be used instead if they are lower.

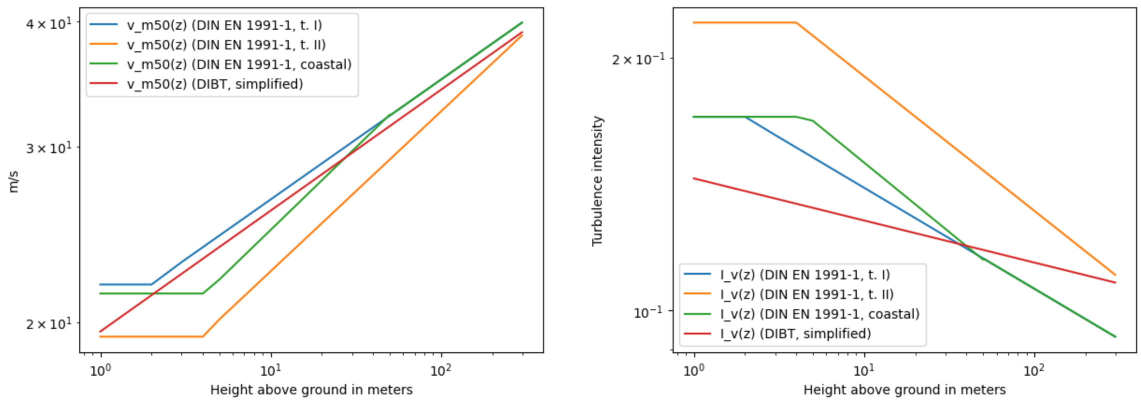

Figure 15: The simplified functions for $v_{m50}^S(z)$ (left) and $I_v^S(z)$ (right) compared to the functions from DIN EN 1991-1-4 [4] in wind zone 1, plotted in a range from 1 to 300 meters above ground.

3.5.2 Simplified operating wind speed

The 1-year average wind speed determined with data of at least 1 year at hub height z_{hub}, denoted as $v^S_{ave}(z)$, can also be simplified. If there are no lower measurements available for the site, then It can be estimated as follows:

$$v^S_{ave}(z) = 0.18 \cdot v^S_{m50}(z)$$

for normal mainland sites, and

$$v^S_{ave}(z) = 0.2 \cdot v^S_{m50}(z)$$

if the site is on a North Sea island.

3.5.3 Simplified wake turbulence criteria

If many wind turbines are placed inside a wind park, then the turbulence wake generated by each wind turbine can potentially increase the turbulence intensity for another wind turbine. Thus, it has to be insured that turbulences generated by nearby wind turbines does not compromise the structural integrity of a wind turbine. For that, the additional wake turbulence intensities of nearby turbines need to be added to the currently analyzed turbine's site effective turbulence intensity. But for wind turbines that are designed for the turbulence category A or higher, the wake effect can be neglected under certain conditions. Let T be the currently analyzed turbine and T_j a nearby turbine. The added wake turbulence intensity effect of T_j on T can be neglected if:

$$\hat{d}(T, T_j) \geq 8, \text{ for } v^S_{m50}(z^T_{hub}) \leq 40\text{m/s}$$

$$\hat{d}(T, T_j) \geq 5, \text{ for } v^S_{m50}(z^T_{hub}) \leq 45\text{m/s}$$

where

$$\hat{d}(T, T_j) = \frac{d(T, T_j)}{max(D^T, D^{T_j})}$$

So $\hat{d}(T, T_j)$ is the distance between both turbines, normed by the maximum rotor diameter. $d(t_i, t_j)$ represents the Euclidean distance between turbines t_i, t_j, and $v^S_{m50}(z^T_{hub})$ is the hub height of turbine T at site S. So because turbulence is less at higher wind speeds, the required minimum distance is less at higher wind speeds. For values of $v^S_{m50}(z^T_{hub})$ between 40 and 45 m/s, the minimum required distance has to be interpolated linearly, which means that the following condition has to be fulfilled to allow neglecting the added wake turbulence intensity effect of T_j on T:

$$\hat{d}(T, T_j) \geq -\frac{3}{5} v^S_{m50}(z^T_{hub}) + 32, \text{ for } 40 < v^S_{m50}(z^T_{hub}) < 45$$

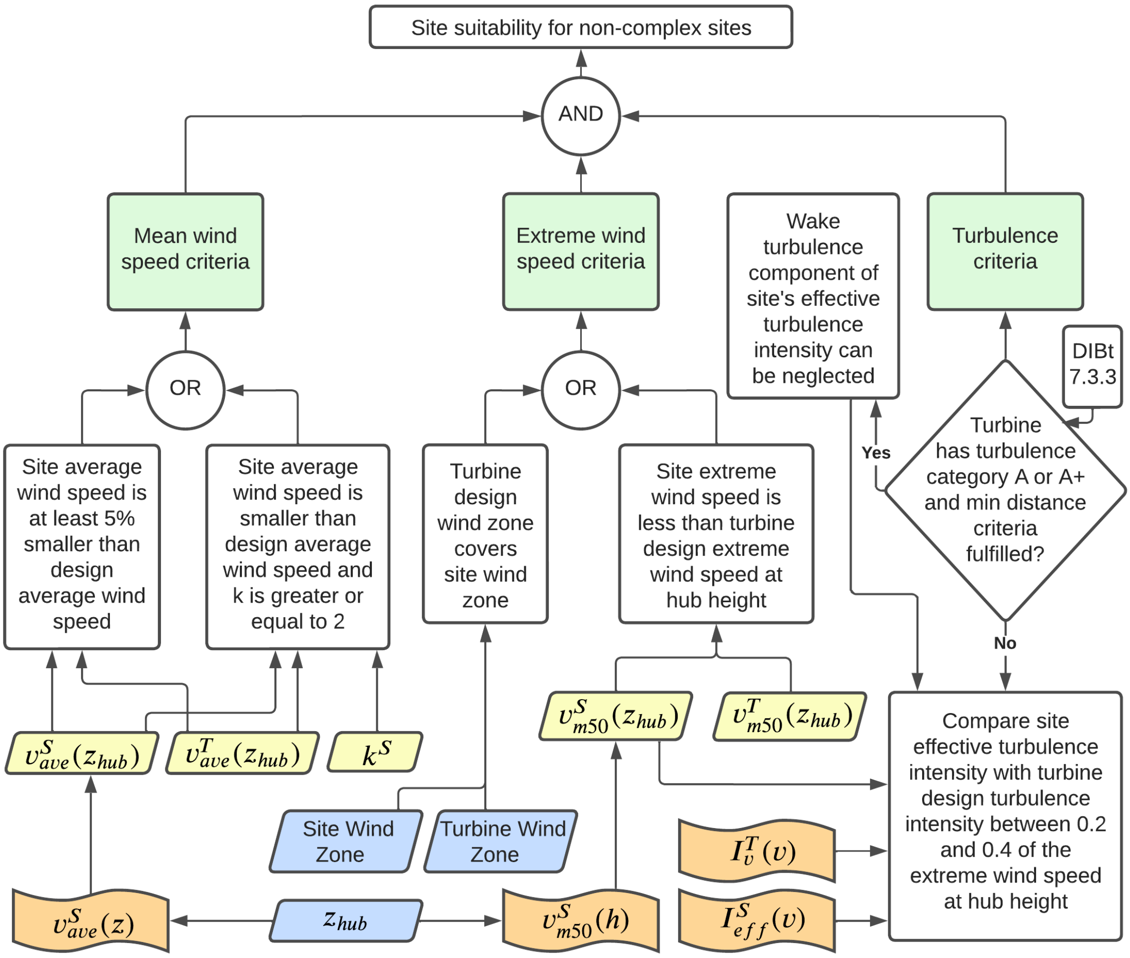

Figure 16: The flowchart for determining the operational safety of a turbine placed at a non-complex site.

3.5.4 Air density

If there is no data available to determine the site's air density, then a standard value of $\rho^S = 1.225 kg/m^3$ can be used.

3.5.5 Site suitability model for non-complex sites

the "Richtline für Windenergieanlagen" offers a simplified procedure to determine if a wind turbine is suitable for a particular site, if it can be shown that the site's terrain is non-complex. Section 3.3.6 describes how the complexity of a site can be computed.

Figure 16 shows all site and turbine parameters that are required for the non-complex site suitability model. Using this data, the simplified comparison can be performed as seen in Figure 16 with the following steps:

1. The mean wind speed needs to fulfill one of two conditions:

a) The mean wind speed $v_{ave}^S(z_{hub})$ at the site is at least 5% smaller than the design wind speed $v_{ave}^T(z_{hub})$ of the wind turbine. $v_{ave}^T(z_{hub})$ can be obtained by the design wind speed class of the wind turbine using Section 3.3.1 or has to be provided manually if the class is S.

b) The mean wind speed $v_{ave}^S(z_{hub})$ at the site is smaller than the design wind speed $v_{ave}^T(z_{hub})$ of the wind turbine and $k \geq 2$ holds for the Weibull distribution form parameter k.

2. Comparison of the site's effective turbulence intensity determined according to Section 3.4 with the wind turbine's design turbulence intensity determined according to Section 3.3.3 in the range from $0.2v_{m50}^S(z_{hub})$ to $0.4v_{m50}^S(z_{hub})$.

3. Compare the 50-year extreme wind speed:

   a) The wind zone at the site is smaller or equal to the design wind zone.

   b) The site's $v_{m50}^S(z_{hub})$ is smaller than the wind turbine's maximum design $v_{m50}^T(z_{hub})$.

If all conditions are fulfilled, then the wind turbine is considered suitable for the given site. When any condition is not met, the planner can still achieve a site suitability certificate, but a professional assessment based on load comparisons must be executed. Usually, these load comparisons are done by the turbine's manufacturer.

This simplified method can also only be used for topographically non-complex terrains.

3.6 Overall site suitability

Figure 17 shows the overall process of determining the suitability of a site for a particular turbine placement. At first, it should be determined whether the given site is complex or not. If a site is not classified as not complex, then we can use the model from the German policy for wind turbines [3], using the simplified site parameters, and the additional filtering of nearby turbines that is defined in the norm. If a site is complex, then it needs to fulfill the criteria for complex sites as defined by the model for complex sites in DIN EN IEC 61400-1 [6].

When an entire wind farm should be analyzed for the site suitability, then the overall model from Figure 17 needs to be evaluated for each turbine placement site.

3.7 Conclusion

It is difficult to give a definitive answer to whether a particular turbine is suitable to be placed at a particular site. Many factors like the soil conditions, availability of access roads for construction, the electrical grid capacity, zoning laws, and the complexity of determining the effective turbulence intensity make it hard to verify the site suitability with limited data. Moreover, to give a final site suitability report, the norms require a manual site inspection [14].

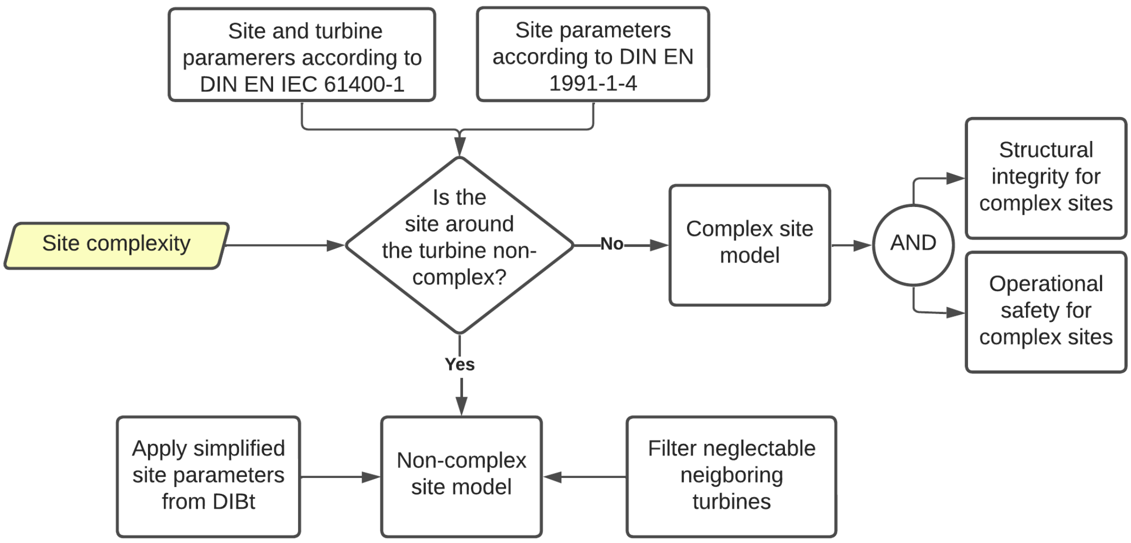

Figure 17: The flowchart for determining the operational safety of a turbine placed at a non-complex site.

But when focusing specifically on local wind conditions, as well as turbulences induced by the local topography and wind turbines around a site, we can evaluate the site suitability fully. It contributes to finding suitable sites for turbines more quickly, and reduces the number of professional site assessments that need to be done because we can exclude wind farm layouts beforehand based on the models we have.

4 Site suitability implementation and comparison

We have fully implemented the models from the previous section using Python. This allows us to compare our results with an existing report.

4.1 Data collection

To compute the site suitability of a turbine at a specific location, we first need adequately accurate topographic data from the placement site. We used OpenStreetMap's Overpass API[8] to collect the topographic data. The Overpass API allows us to query custom parts of the OpenStreetMap dataset, including precise elevation data. For our purposes, it was sufficient to collect 3-tuples of latitude, longitude, and elevation at a grid resolution of 50 meters, which is the minimum required resolution as specified by DIN EN IEC 61400-1 [6]. To make mathematical operations easier, we convert all latitude and longitude coordinates to UTM coordinates, which map to a three-dimensional Cartesian coordinate system. It was important to set the grid boundary with a big margin so that we have data points for a radius of at least 20 hub heights around each turbine placement.

Another indispensable data source is the site's wind data. We collected wind data from the closest available station from OneBuilding[9]. It is difficult to get wind data that is as accurate as required in the norms from a publicly available source. But the site's API delivers hourly measurements including the direction and speed at the nearest measurement station for a typical year. If the results need to be more accurate, then custom measured data can also be provided to the algorithm.

4.2 Comparison with assessment report of Heidsiek

In this section, we will compare our model's results with the site suitability report of I17 for the wind farm in Heidsiek [12]. The report does not provide the full data for every site and turbine, nevertheless most results can still be compared. Table 7 shows the relevant design parameters of the turbines for the comparison.

ID	z^T_{hub}	D^T	m^T	$v^T_{ave}(z_{hub})$	$v^T_{m50}(z_{hub})$	ρ^T	**Class**
W1 - W5	161.0	158.0	14.0	7.5	40.2	1.225	S

Table 7: The data of the five turbines of the report.

4.2.1 Site complexity

The starting point of the site suitability model is determining the site complexity of the currently evaluated wind turbine. Figure 18 shows the site around turbine W1.

[8] Overpass API: www.wiki.openstreetmap.org/wiki/Overpass_API
[9] OneBuilding: www.climate.onebuilding.org

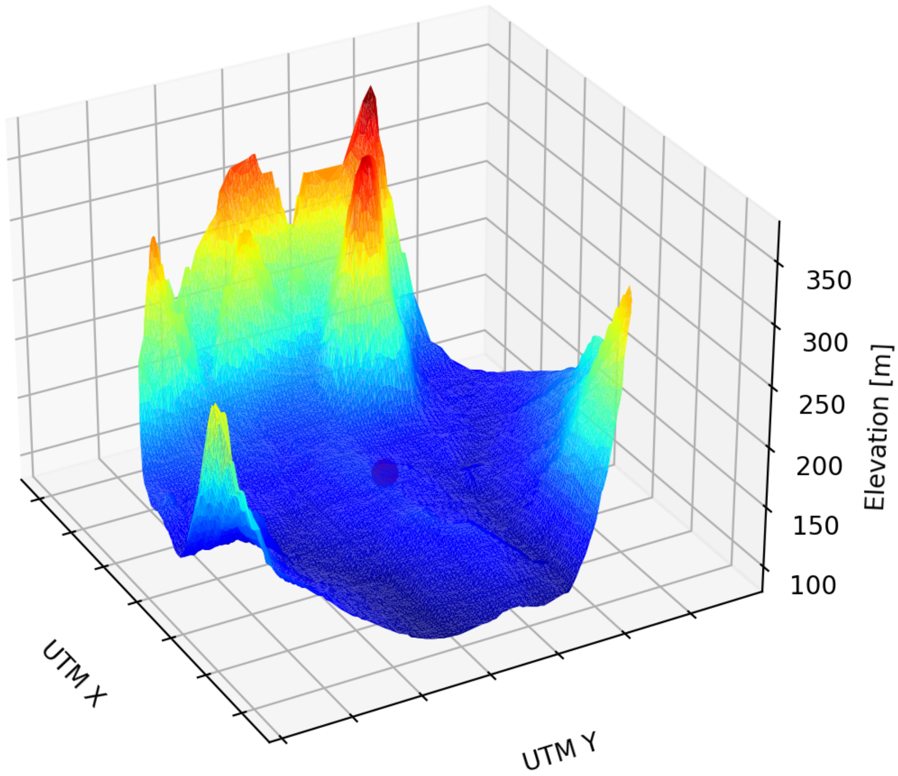

Figure 18: The terrain data around turbine W1 in the wind farm Heidsiek. Note that the scale of the z-axis is exaggerated. The turbine has a hub height of 158 meters, so the terrain points disc has a diameter of 3160 meters.

The terrain is divided into 12 sectors, and for each sector, a planed is fitted to the terrain points as seen in Figure 19. Based on the fitted planes for each sector each radius of 5, 10 and $20z_{hub}$, the terrain slope and variation indices are computed, which then provide us the information required to determine the site complexity. As seen in Table 8, all five sites are non-complex, according to both the I17 report and our results. Accordingly, the model for non-complex sites will be used in the following.

Turbine ID	Terrain complex? (theirs)	Terrain complex? (ours)
W1 - W5	No	No

Table 8: All five sites have been rated as not topographically complex by the I17 report and our algorithm.

4.2.2 Mean wind speed

In contrast to the I17 report, our result is that the mean wind speed of $8.03m/s$ for the sites is too high for the turbine's design wind speed, which is $7.5m/s$. This can be explained by the fact that we use a different wind data source. The I17 report has

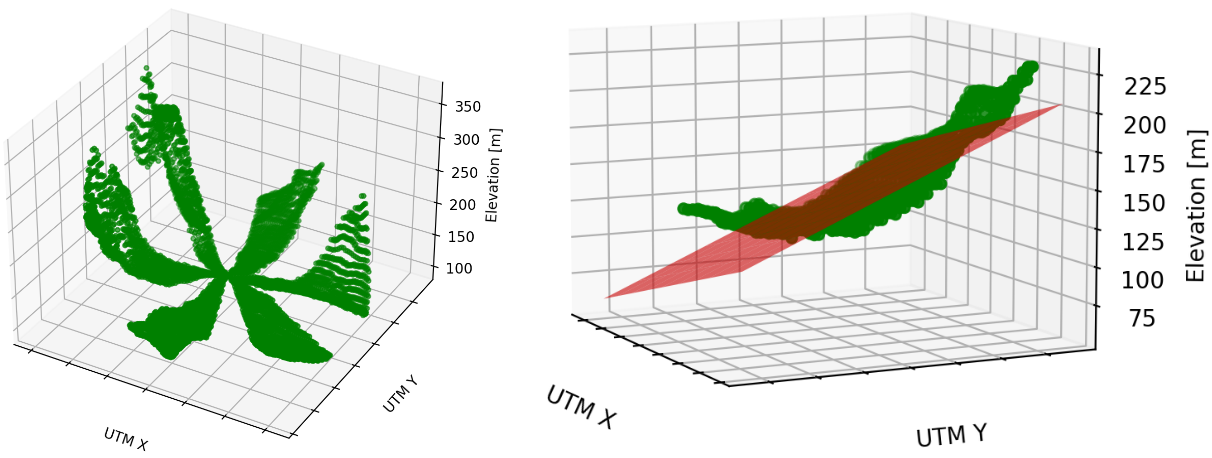

Figure 19: On the left, the terrain points of every other sector around turbine W1 with $20z_{hub}$ radius are plotted. The right shows only one of those sectors, with a plane fitted to it. The plane has a slope of 1.5 degrees.

been made with local measurements that were submitted by the client, which turns out to have a mean wins speed of $6.57m/s$ for each turbine at hub height. In contrast, our data is from a nearby measurement, and was measured at $100m$ height at $7.4m/s$ but needs to be scaled to $8.03m/s$ using the wind profile power law to estimate the wind speed at hub height. If we manually input the measured mean wind speed of the report to our algorithm, we get the same result, which is that all turbines are suitable for their site placements, regarding the mean wind speed.

4.2.3 Extreme wind speed

Since we estimate the extreme wind speed function according to the wind zone and terrain roughness category, we get the same results as the I17 report here. For each turbine placement site, we get an extreme wind speed value of $35.09m/s$. The I17 report gives a value of $35.1m/s$, so they probably rounded the number up. All turbine's design extreme wind speeds are $40.2m/s$, so the sites are suitable regarding the extreme wind speed as well.

4.2.4 Turbulence

The last aspect which needs to be compared is the effective turbulence intensity at the site and the wind turbine's design effective turbulence intensity according to the NTM model. Figure 20 compares them for turbine W1. As we can see, there is a difference in the effective turbulence intensity when comparing our result with the report. Again, the differences are most likely due to the different wind data sources. While our estimate is lower up to about $8m/s$, the report's estimate is significantly lower for higher wind speeds. But overall, both estimates are always below the design turbulence intensity of the turbine, so the overall site suitability is given.

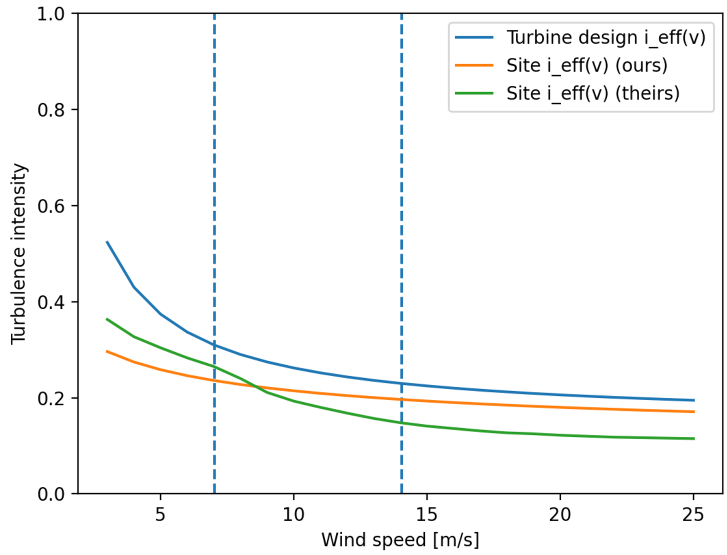

Figure 20: The effective turbulence has to be compared in the range of $0.2v^S_{m50}(z_{hub})$ to $0.4v^S_{m50}(z_{hub})$. For the turbine and site W1, the design function, the function from the I17 report, and our computed function are shown.

4.3 Conclusion

As we can see, the most important part of the site suitability assessment is having accurate data to work with. Because we did not have the measured wind data on site, our results are slightly different, but if we had their wind data, the results should theoretically be the same.

Based on the topographical structure and the wind conditions we have, the models allow us to exclude areas where the operating or extreme wind conditions are exceeded. Using the effective turbulence intensity, we can also tell whether the site suitability criteria are passed or not. If a wind turbine fails some criteria, a load comparison can be carried out by the manufacturer, but this has to be carried out manually. In that case, the process becomes more complicated because the manufacturer has to be contacted.

Overall, our implementation is particularly useful in the context of a wind turbine layout editor, where turbines can be moved around freely. It would allow a user to quickly narrow down the potential areas in a given region, when the editor supplies feedback for each changed turbine placement if it violates some of the criteria. To compensate for the lack of real wind data measurements, we could give the user an indication that according to the nearest weather station data, the site is probably or probably not suitable inside a certain margin of error.

5 User Experience Design

As applications get more complex, the user experience design - also called UX design - becomes increasingly important. A good UX design should help users achieve the goals they want to achieve with an application, instead of hindering them or generating confusion. Complex workflows need to be broken down so that users can be guided through self-explanatory logical steps. It goes hand in hand with an easy-to-understand navigation and app structure and easily reachable interface elements, whether the user is on a desktop computer or a mobile device.

To provide wind farm planners with an application that fulfills these standards, such that the UX supports their planning process as nicely as possible, several UX design laws and practices have been assessed. Various sources [40, 13] describe practices of UX design. They are grouped into four categories: Heuristics, principles, appearance-related, and cognitive biases. If not mentioned otherwise, the described UX law is taken from [13].

5.1 Heuristics

Here are the heuristics that describe how users perceive and interact with interfaces and provide rules of thumb to optimize them:

1. **Aesthetic Usability Effect**: Users often perceive aesthetically pleasing designs as more usable. In experiments with interfaces, a stronger correlation exists between perceived usability and aesthetic appeal than between actual usability and aesthetic appeal [39]. This implies that users will not notice smaller usability issues if the design looks good. On the other hand, if the focus is good usability, then building a less visually appealing application prototype might be helpful for uncovering usability issues.

2. **Goal-Gradient Effect**: The closer a user is to the completion of a task, the faster they work towards reaching that goal. Indicating the progress of a task, maybe even exaggerating how close it is to its completion, will motivate users to complete it.

3. **Fitts's Law**: The time it takes to select a target depends on the distance to the target and size of the target. According to the speed-accuracy tradeoff of the human motor system discovered by Paul Fitts, the further away and the smaller an element is, the harder it is for users to select it. It follows that the distance between related elements that require interaction should be as short as possible. It also follows that interaction targets should have enough space between them to account for inaccuracies. Figure 21 visualizes Fitts's law.

4. **Hick's Law**: The time it takes to make a decision increases with the number and complexity of the possible choices. Thus, choices should be minimized wherever

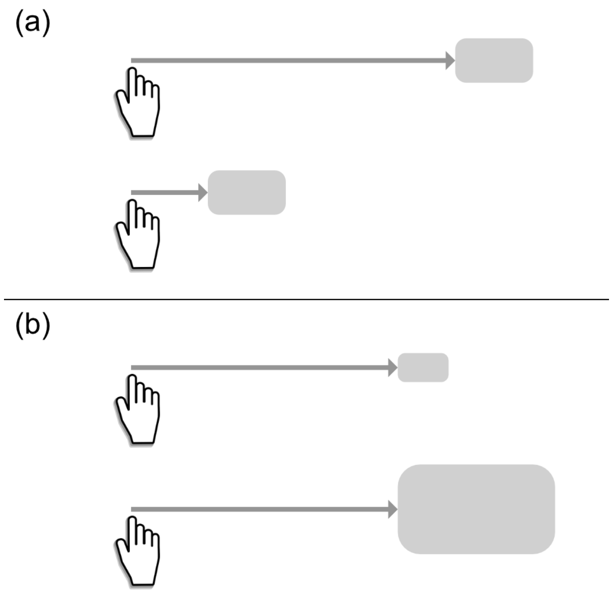

Figure 21: The implications of Fitts's law: (a) If an element of the same size is closer to the user's cursor or finger, the element becomes more accessible and quicker to press. (b) If an element at the same distance from the user's cursor or finger is larger, the element is also easier and quicker to press.

possible or broken up into multiple smaller steps to decrease cognitive load and increase the users' decision speed. New users should be gradually exposed to the features of an application so as not to overload them.

5. **Jakob's Law**: Users spend the majority of their time in other applications and sites, so they prefer an interface that works like the ones they already know. By adhering to commonly used interface elements and design practices, users do not have to learn new concepts and can leverage their existing knowledge, and users will understand the application faster.

6. **Miller's Law**: The average person can keep about seven items in their working memory. Like Hick's Law, this law also implies that complex workflows should be broken into smaller chunks. The user's working memory will not be overloaded by organizing content into logically cohesive pieces.

7. **Parkinson's Law**: Any task will inflate until all the available time is spent. That is why a task should be limited to the time a user expects it to take.

5.2 Principles

Here are the principles that guide the UX design process:

1. **Doherty Threshold**: A user's productivity is increased when responses of the computer happen at a quick pace (<400ms) that ensures that the user does not

have to wait. Waiting times should therefore be reduced to a minimum. If a delay is inevitable, for example, due to waiting for data, the computer should still quickly indicate that it is loading. Overall, multiple short waiting times are preferred over a single long one.

2. **Tesler's Law**: Also known as the law of conservation of complexity. It states that there is a certain amount of complexity for any system that cannot be reduced. The design should help users deal with the inherent complexity of a task and not hinder them. Laws like Hick's Law and Miller's Law show that simpler interfaces are desirable, but Tesler's Law also shows that there is a sweet spot. Too much simplification will make interfaces worse.

3. **Occam's Razor**: Among competing hypotheses that predict a situation equally well, the one with the fewest assumptions should be selected. The takeaway is that the best method for reducing complexity is to avoid it altogether, as far as Tesler's Law allows it. Elements should be removed when the removal does not reduce overall usability and function.

4. **Pareto Principle**: The Pareto principle states that, for many events, roughly 80% of the effects come from 20% of the causes. Therefore, the focus of an application's development should be targeted where it offers the most benefit to the most users.

5. **Postel's Law**: Be liberal in what you accept, and conservative in what you do. Regarding inputs and how a user interacts with an interface, almost anything needs to be expected, in the lines of "If something can break, the user will manage to do it". A good application anticipates this and tries to accept inputs as long as they can be parsed into the required input format. Alternatively, give the user one obvious way how the input should be provided (for example, a date picker instead of a free input field). Afterward, only proceed if the input respects the specification to provide resilience against inconsistent states or errors.

5.3 Appearance

The following laws deal with the appearance of interfaces. Humans perceive patterns based on specific rules. These are:

1. **Law of common Region**: Elements tend to be perceived in groups if they are sharing an area with a clearly defined boundary. A border, divider, or background can easily indicate such groups. Figure 22 shows this law visually.

2. **Law of Similarity**: As seen in Figure 22, the human eye tends to perceive similar elements in a design as a complete picture, shape, or group, even if those elements are separated. The color, shape, orientation, and movement can signal that elements belong together or do similar things. Accordingly, dissimilar things should look different.

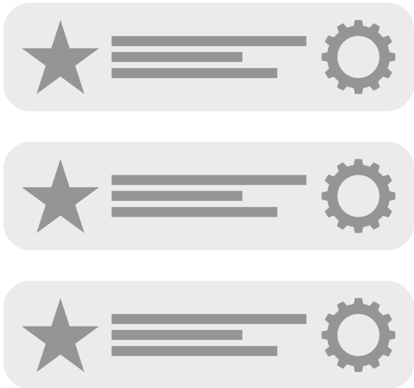

Figure 22: The law of common region and the law of similarity are demonstrated. By placing symbols inside the light gray background box, the law of common region can be observed. The box suggests that all symbols inside it relate to one specific item. Furthermore, the law of similarity is demonstrated. Symbols that look similar are perceived as related, even if separated by being in separate boxes. For example, the cog symbols are perceived to perform the same action on each element when clicked, like opening a settings page for the associated item.

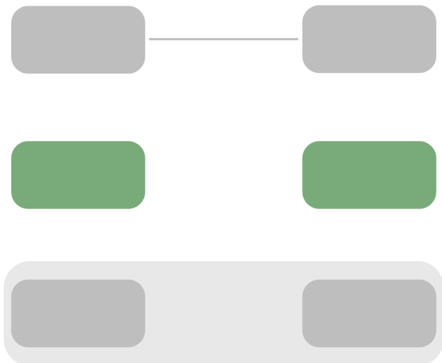

Figure 23: This figure depicts the law of uniform connectedness. Connecting elements with lines, by color coding, or with frames makes them visually appear to belong to each other.

3. **Law of Proximity**: Objects that are spatially close to each other are perceived to share properties or belong to the same task.

4. **Law of Prägnanz**: Humans are excellent at perceiving patterns. So, when a complex image is perceived, it will be interpreted based on the patterns that can be found in it. The simplest patterns are those that will be interpreted first, since they require the least mental effort.

5. **Law of Uniform Connectedness**: Elements that are visually connected are perceived as more related than elements with no connection. Colors, lines, or frames can indicate connectedness. Figure 23 displays how elements can appear to belong to each other according to the law of uniform connectedness.

5.4 Cognitive biases

The last category of laws describes cognitive biases:

1. **Peak-End Rule**: People judge an experience largely based on how they felt at its peak and its end. According to [23], humans prefer the more uncomfortable alternative of two experiences, but only if the end of it is more pleasant. Thus, attention should be paid to the final moments of a task. A more positive experience is achieved by creating positive feelings at the end of a task, such as a fun animation at completion.

2. **Serial Position Effect**: Humans have a predisposition best to remember the first and last items in a series. So, the top and bottom of a page and the left and rightmost areas are important places.

3. **Von Restorff Effect**: When multiple similar objects are present, the one that differs from the rest is most likely to be remembered. Thus, essential information and actions should be visually distinctive. However, if elements start to compete for attention because of too much emphasis, the overall experience will suffer.

4. **Zeigarnik Effect**: People remember uncompleted or interrupted tasks better than completed tasks. So if a task takes processing time to complete, it should be indicated that the task is incomplete. As in the Goal-Gradient Effect, a progress indicator will motivate users to complete a task. Indicating more significant progress than the actual progress will also motivate users to complete a task more.

5. **Amara's Law**[10]: We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run. As a result, adopting technologies very early and expecting great results is not the best approach.

[10] Amaras Law: www.oxfordreference.com/view/10.1093/acref/9780191826719.001.0001/q-oro-ed4-00018679

6 Planning application architecture

Implementing an application that allows planners to design and validate wind farms is one of the main objectives of our work. This section will discuss our requirements, the development, the design process, as well as the final architecture. In addition, we will present the available technologies and why we selected the specific technologies are used. Afterward, in Section 7, we will showcase the resulting application.

6.1 Requirements

Our application has a distinct target group of users: wind farm planners. Those planners are usually engineers that work for companies or municipalities, and their needs can range from checking a particular turbine layout for feasibility or changing a design to make it feasible to search for suitable sites over a larger region. Wind farm planners are usually experts in their domain. Therefore, we can expect a certain degree of skill and knowledge from the user. They will use the application at their workplace or on-site. We formulated the following requirements regarding our target user group:

1. **Responsive design**: The majority of the usage will be in the browser at the workplace of the planner. Preferably on a large screen, which makes it easier to edit the layout of a wind farm on a map. Nevertheless, planners will also want to use the software at potential sites to conduct changes while communicating with nearby residents or landowners in person. Thus, the application should be responsive enough to adapt to the screen sizes of tablets and smaller laptops as well. We have considered the usage of mobile phones as well, but have found that the small screens make it difficult to edit layouts accurately. Furthermore, due to the professional nature of users, it can be expected that they will use devices provided by their workplace. Thus, mobile phones are not a development target. The primary focuses are desktop computers, laptops, and tablets.

2. **Offline capabilities**: When planners use the application on-site, they might not have internet access at all times, since the potential site might be in a remote region. That is why the application should be capable of running offline whenever possible, while keeping track of local storage changes. The changes should be synchronized with the server as soon as the internet connection is established again.

3. **High usability**: As the application offers many features and will likely grow in the future, we want to follow the best practices of UX design to support the planning process and project administration with a good graphical user interface.

4. **Maintainable**: Because a continuing development is expected, we want to choose the most widely used and best-documented technologies available to us. Choosing standard technologies enables a bigger pool of potential future developers to advance the application's development and ensures that the used

technologies will be supported well into the future. We also want to make the software easy to comprehend and extend. We especially want to focus on writing concise code and following best practices of the respective development domains. A single codebase is highly desired to increase the maintainability, even when deploying to multiple platforms like the web, iOS, and Android.

5. **Secure**: Planners desire to keep their projects protected, so their business secrets stay confidential. The application should include essential security features like user authentication, project authorization, and encryption.

6. **Compatible**: The application should be compatible with the already existing broader project context. Simulations from other contributors should easily be incorporated into the application.

6.2 Overall application architecture

Before going into detail about the individual application components and choices, we will now present the overall architecture of the application. In Figure 24, the overall application architecture can be seen. It can be divided into the frontend and backend at the highest level of abstraction.

The backend is hosted on a web server, manages all the data, regulates access to the data, and performs simulations. It uses a Nginx reverse proxy for managing incoming traffic, which forwards traffic on the "/graphql" route to a Python FastAPI application using Strawberry as the GraphQL server framework. All other application routes, e.g., "/user_dashboard" are forwarded to a Node.js instance which serves the frontend application pages. The Python application stores its data in a PostgreSQL database, and directly calls the simulation scripts.

The backend serves data with a GraphQL API, through which it provides functionality to the frontend.

The frontend is the point of interaction with the user. It provides all the application's functionalities, from registering as a new user, to creating projects, modifying projects, and dispatching simulations. We use Ionic with Vue as the frontend frameworks of our choice. The frontend application gets all its data from the Apollo cache, which in turn manages all communication, while the store manages its data by sending GraphQL queries and mutations and saving its state to the capacitor store for offline capabilities.

In the following, we will go into detail on how the application was developed, and reason about why the architecture was chosen as presented.

6.3 Development process

The development process of an application requires careful consideration before starting with the implementation. Several techniques exist to manage this process.

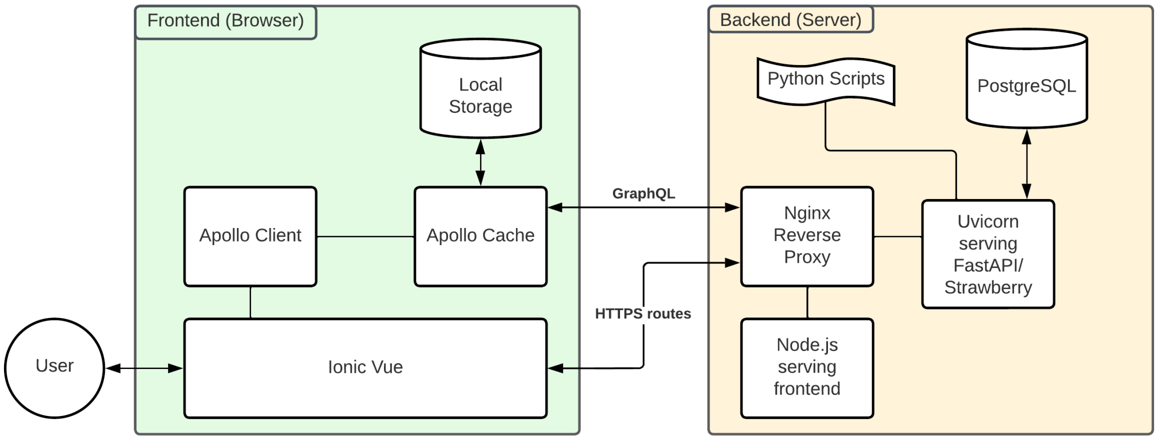

Figure 24: This figure shows the components of the overall application. A Frontend-Backend architecture is used.

6.3.1 Prototyping

One of those techniques that is critical to the early development phases is called prototyping. A prototype is a version of the application that resembles the final application in some respects, but does not implement its full functionality. For example, a prototype can model the visual appearance of some piece of an application without having to implement the associated code. Alternatively, it can model a specific workflow without the backend logic or database interactions. The key here is that building a prototype is faster and requires a lot less effort than the actual implementation of the modeled aspects of an application. Many things can be tested quickly by integrating prototyping into the development process so that potential issues can get discovered early on.

Prototypes can have varying degrees of complexity. The simplest form is sketching prototypes manually on paper or with graphics tools. Those prototypes are the least detailed and the quickest to create, and can effectively convey the general idea of a concept. If a prototype should be more elaborate, specialized prototyping tools can be utilized, like Figma[11], a collaborative interface design tool. At the highest degree of complexity, there are fully functioning prototypes that completely implement one aspect of an application. These prototypes will usually be created if a proof of concept is necessary. Generally, as a prototype's complexity increases, it also requires more effort to build but delivers more details at the same time. On the other hand, a simpler prototype is faster to build and thus easier to discard, but also less accurate.

We decided to emphasize prototyping in our development process due to all the benefits it provides: It allows us to model workflows with less effort than actually implementing them, so we can quickly find usability issues and identify missing or unclear components. With this methodology, we lose less time debugging or restructuring the application. Because we already had a general idea of what the applications should

[11]Figma: www.figma.com

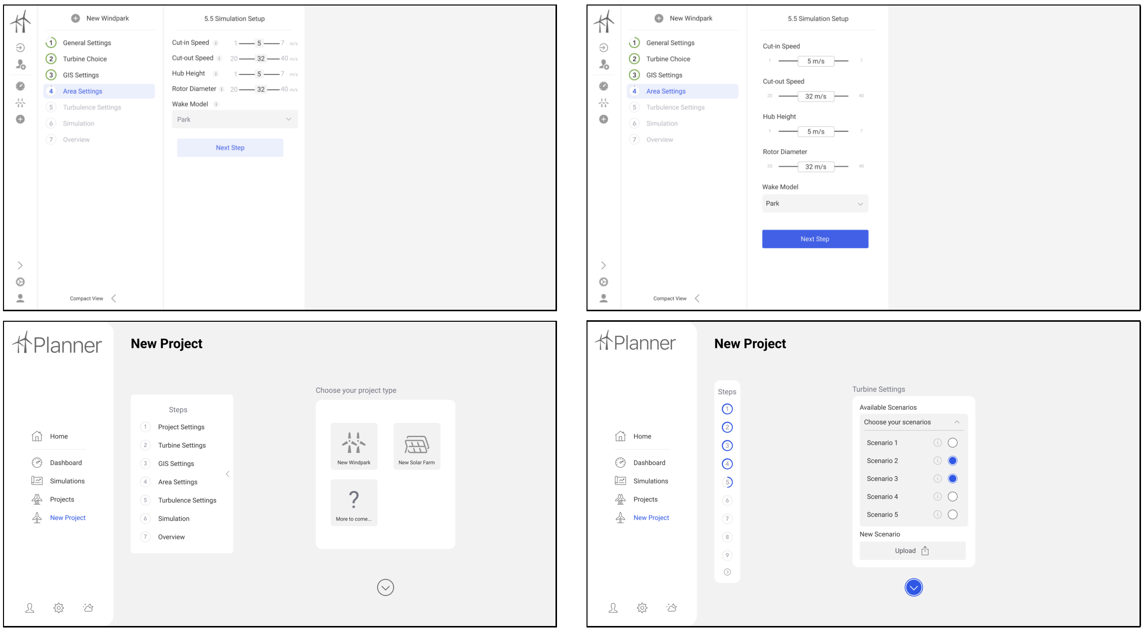

Figure 25: The interface prototype of the project configurator during the development process, built with Figma. The top two images show the project configurator from an early development stage, and the bottom two images show it in a later stage.

be able to do, we did not use low complexity prototypes like sketches. Furthermore, a proof of concept was also not required, so highly complex prototypes were not required as well. Instead, we went for a middle ground, and mainly worked with interface design prototypes.

Our tool of choice was Figma, which was used to create prototypes of the interface. Those prototypes were discussed and refined before actually starting to implement the application. Examples of our prototype during the early and final prototyping phases can be seen in Figure 25. The upper screenshots show the new project configurator early in the prototyping process, while the bottom screenshots show the new project configurator from the final prototype. Notably, the final application looks quite different from the early prototyping stage. Some prototype parts were deemed satisfactory and are now part of the application as the new project configuration progress indicator. Other ideas were discarded, like the left-aligned stacked tab layout. The layout was uncommon, and due to Jakob's Law (Section 5.1), we opted for a more traditional page-based layout. We increased the size of several interface elements due to Fitts's law (Section 5.1), and strategically used more whitespace due to the law of proximity (Section 5.3).

The saved amount of time demonstrates the effectiveness of prototypes. The time saved by not implementing the discarded layout is significant enough to offset the time it costs to create prototypes.

6.3.2 Versioning

To achieve a high efficiency during the implementation, we employed state-of-the-art tools to organize the development process. We used git as our version control tool, and Gitlab[12] was the git platform of our choice. Gitlab allow us to work with versioning, issues, and code branches, and provides many tools like the Kanban-style development boards, which help to organize issues and interlink them based on dependencies.

6.4 Application types

To arrive at an informed decision of what type of application is best suited for our use case, we will now review the most common application types and discuss the advantages and downsides of each type.

6.4.1 Native applications

One approach is to build one application for every operating system that we want to support natively. The main advantage of native applications is that they provide the most seamless user experience. They can take advantage of the specific platform's features and capabilities, such as native UI elements, and low-level access to a device's hardware capabilities, like GPS, sensors, and background tasks.

Because we want to make the application accessible on most computers and tablets, we would have to at least develop the same application frontend for each of the most popular platforms. This implies using the native development tools of each platform: Kotlin and Android Studio are used for Android, Swift, and Xcode for macOS[13] and iOS[14]. HTML, CSS, and JavaScript for the Web, and so forth.

To get proper support for enough devices, a codebase for at least Windows, macOS, Linux, Android, and iOS [15,16] would be required.

This emphasizes that the time-consuming process of developing for each platform makes this approach unsuitable because of our high maintainability requirement.

6.4.2 Web application

A web application is hosted on a server and serves the application under some route on the internet. Here, only one codebase exists, which makes the application easier to maintain. Moreover, since the application is accessed through the browser, it will be available on all types of user devices, as long as they have a reasonably modern web browser installed, and the application is responsive enough and adapts to the viewport dimensions well.

[12]Gitlab: www.gitlab.com
[13]macOS app development: www.developer.apple.com/tutorials/swiftui/creating-a-macos-app
[14]iOS app development: www.developer.apple.com/tutorials/app-dev-training
[15]PC operating system market shares: www.gs.statcounter.com/os-market-share
[16]Mobile operating system market shares: www.gs.statcounter.com/os-market-share/mobile/worldwide

However, since our applications also should be capable of operating without an internet connection, a regular web application will not meet our requirements.

6.4.3 Hybrid application

Hybrid applications can be built with many frameworks and only require one codebase, but also can get deployed to different platforms. They provide a decent tradeoff between native and web applications. On the one hand, they might not be as performant as native apps because they employ an additional abstraction layer. On the other hand, they provide an interface to advanced device capabilities, like offline operation and access to operating-level features such as notifications and background tasks. Thus, they are more versatile than regular web applications.

There are various types of hybrid applications, like Flutter[17] and React Native[18], which compile their code into native elements, and other ones like Ionic[19], which are based on web technologies.

Since we want to utilize widely used and well-documented technologies, Ionic is the framework of our choice. Ionic uses HTML, CSS, and JavaScript (or Typescript) to build applications and is compatible with the most used front end frameworks like React, Vue, and Angular. According to the Stack Overflow 2021 developer survey[20], JavaScript was the most popular programming language in 2021. In the second place, the markup languages HTML and CSS are listed. Ionic also integrates tightly with Capacitor[21], allowing us to deploy to the web, iOS, and Android using a single codebase. Therefore, high maintainability can be achieved. Because web technologies include several tools intended to make responsive design easy (like Flexbox[22] or CSS media queries[23]), it becomes easier to fulfill the requirement of adequate responsiveness.

6.5 Web-based frontend

As we have decided to use a web-based hybrid application, we will now discuss the possible choices for the underlying web application. The naive approach would be to use the default languages of the web: HTML, CSS, and JavaScript.

However, frameworks exist to solve the problems that arise when an application scales and becomes more complex. Frameworks like Vue[24], React[25] and Angular[26] introduce advanced concepts like reactive programming, components, lifecycle manage-

[17] Flutter: www.flutter.dev
[18] React Native: www.reactnative.dev
[19] Ionic Framework: www.ionicframework.com
[20] Stack Overflow 2021 Developer Survey: www.insights.stackoverflow.com/survey/2021#technology-most-popular-technologies
[21] Capacitor: www.capacitorjs.com
[22] Flexbox: www-developer.mozilla.org/docs/Learn/CSS/CSS_layout/Flexbox
[23] www.developer.mozilla.org/docs/Web/CSS/Media_Queries/Using_media_queries
[24] Vue: www.vuejs.org
[25] React: www.reactjs.org
[26] Angular: www.angular.io

ment, and routing, giving developers the means to build better applications. Although frameworks will usually increase the application size, require a compilation step, and come with many dependencies, their advantages outweigh the negatives in most cases. Those three frameworks are the most widespread ones, and all three can be used with Ionic.

For our application's frontend framework, we picked Vue because it is regarded to be the easiest to learn and the most popular[27] on GitHub of the three, is well documented, and open source while having no apparent disadvantages over the others.

In combination with the Ionic router, it is possible to build progressive web apps that do not reload the page when navigating from view to view, creating a quick and native-feeling experience.

6.5.1 Styling

Since we listed high usability as one of our goals, delivering a good interface is essential. We evaluated multiple CSS libraries. Tailwind was promising at first and is quick to prototype with, but also produces cluttered HTML. Furthermore, the official documentation reveals that there are some serious flaws associated with the library, as they recommend "multi cursor editing" as a suggestion for how to reuse styles [28], which is less than ideal. Others like Bootstrap or Foundation do not interoperate well with Ionic.

That is why we have decided to stick with Ionic's UI components wherever possible. As a result, the application becomes more maintainable, developers will not need to learn another library, and will be familiar with everything if they learn Ionic, which is a foundational requirement anyway. Furthermore, Vue's components and scoping rules make it straightforward to style interface elements with plain CSS and avoids code repetition.

6.5.2 Frontend state management & offline capabilities

We use Apollo Client and Apollo cache[29] to manage data in the frontend. This is the most logical choice, since Apollo is the industry standard when it comes to GraphQL client libraries. The addition of the cache helps to reduce network requests to the server, and also makes it easier to manage complex interactions between components of the application. All data in the cache is reactive by default. This means that if one component changes the data in the cache, like adding a new project, then this change is automatically reflected in all other components, for example in the projects overview page. Furthermore, the cache can also return data if the application is offline.

We use the Apollo cache as a data authority. It will handle fetching the requested data from the server, or return it from the local storage if already stored, or no internet connection is available, and the operation is permitted.

[27]Most starred JavaScript repositories: www.github.com/topics/javascript?o=desc&s=stars
[28]Tailwind - Reusing Styles: www.tailwindcss.com/docs/reusing-styles
[29]Apollo cache: www.apollographql.com/docs/react/caching/overview/

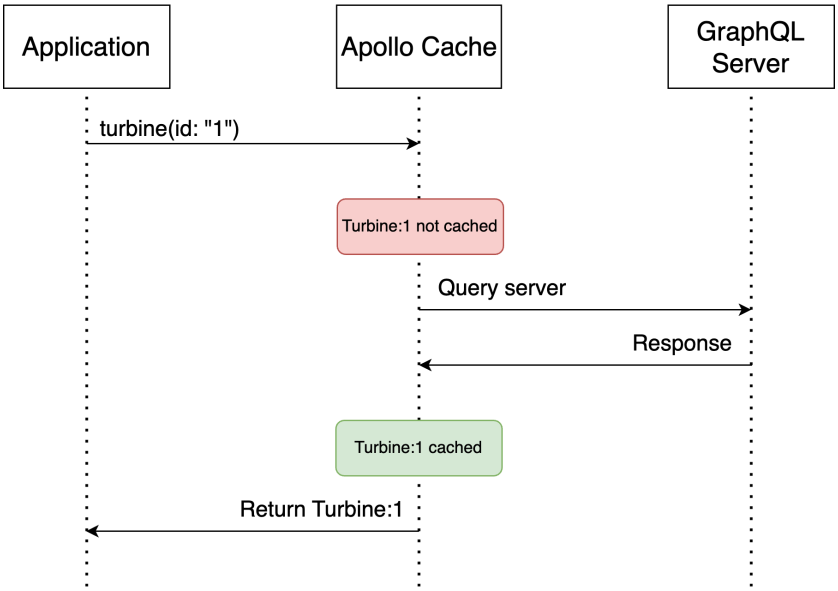

Figure 26: An example how Apollo cache works when an object is queried the first time.

In Figure 26, we can see what happens when the application queries an object for the first time. The turbine is not cached, so the server is queried. Afterwards, the result is cached by the object type and ID, and is returned to the application. Subsequent requests will return the cached turbine instead of querying the server. Apollo has a wide variety of cache policies, so the maximum age and re-fetch conditions can be defined easily.

Figure 27 shows what happens if the application is offline. If the turbine is cached, it will be returns without issues. However, if it is not cached yet and the internet connection is lost, an error has to be returned since there is no possibility to get the turbine's data. This also works similarly for mutations, e.g., if a turbine's position is moved while offline. Then, the cache is updated instantly, but dispatching the mutation to the server is deferred until the internet connection is back.

6.5.3 Cross-platform runtime

As stated in the requirements, the planning app is also intended to be used on a tablet or laptop at the planning site. At some remote sites, it will not be guaranteed that the used device will have an internet connection at all times. Despite having no connection, a planner may still like to revise their designs, such as moving a wind turbine or replacing it with another one. That is why Capacitor[30] is another chosen library. Capacitor is a cross-platform runtime that integrates tightly with Ionic, in fact, the Ionic team develops it. It comes with tools to export an application as an iOS or Android application, that will run without an active internet connection. It

[30]Capacitor: www.capacitorjs.com

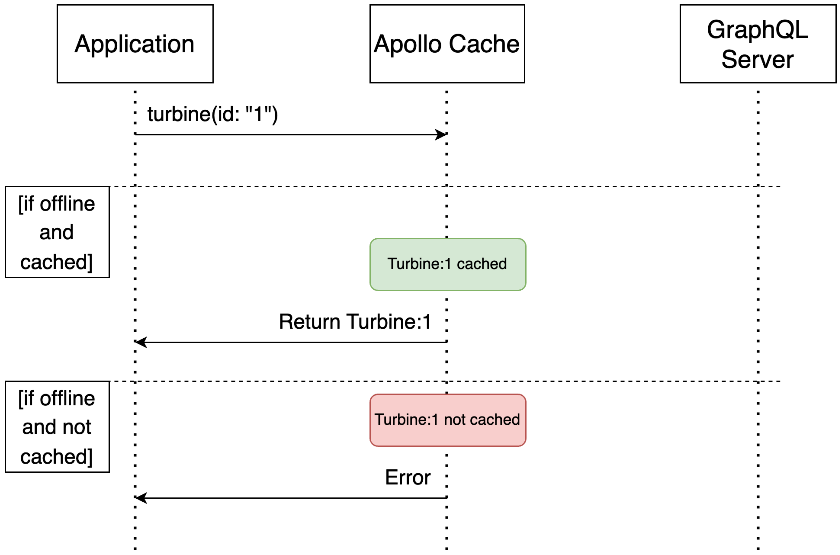

Figure 27: An example of what happens when the app queries an object while offline. Two cases are considered: The object is cached, or not.

also provides utilities to access native device features. Our interest was specifically the Capacitor storage API, which allows us to permanently store data offline. And since Apollo cache's storage provided can easily be swapped to Capacitor's persistent storage, the offline capabilities also easily works with the exported mobile. Of course, some actions, like dispatching a simulation to the server, still require internet access, but our goal is to provide as much functionality as possible when the device is offline.

6.6 Server communication

There are two prevailing principles for building a communication interface between the frontend and backend: REST [41] and GraphQL[31].

6.6.1 REST API

REST stands for representational state transfer and loosely defines the architectural properties of an API. Resources are accessible through URLs, and standard HTTP methods are used, GET to retrieve a resource, POST, PUT and DELETE to dispatch modifications. Here, a resource like a wind turbine is transferred in its entirety, like seen in the example of Listing 1.

So in case a page only requires the model and ID of a turbine, there is a lot more data transferred than necessary. This is called over-fetching. On the opposite, if a page needs information about a turbine and a project, the frontend needs to send two requests. This is called under-fetching. To deal with over- and under-fetching, more sophisticated methods like GraphQL exist.

[31]GraphQL: www.graphql.org

6.6.2 GraphQL

Listing 1: Example of a standard HTTP GET request.

```
GET https://server_url.com/
turbine?id=232

-> {
  "id": 232,
  "model": "MM92",
  "turbine_cost": 1000000,
  "foundation_cost": 280000,
  "output_mw": 2,
  "height_m": 100,
  "rotor_diameter_m": 92,
  "octave_frequencies_db": [
     90, 91, 92, ...
  ],
  "cut_in_mps": 3,
  "cut_out_mps": 24,
  "ct_by_wind_speed": {
    "4": 0.863,
    "5": 0.871,
    "6": 0.877,
    ...
  },
  "mw_by_wind_speed": {...},
  ...
}
```

Listing 2: Example of a GraphQL query. The configuration of the server URL is done in the used GraphQL library (Apollo in our case).

```
query {
  turbine(id="232") {
        model,
        turbine_cost,
  }
}

-> {
  "model": "MM92",
  "turbine_cost": 1000000,
}
```

GraphQL stands for graph query language. It is a little different from a REST API, since it requires the requester to specify which properties of a resource must be returned, as seen in Listing 2. This contrasts with traditional REST APIs, which often return more information than required. GraphQL also allows clients to send a request that contains multiple queries at once, which can be useful in situations where an application needs to retrieve data from multiple different APIs to provide a complete response to the user.

In the end, we decided to use GraphQL since it allows us to design a semantic graph-like API. This makes GraphQL nicely suited for precise queries, so the frontend can request only the data it requires, and all the data it requires with a single request. Despite the higher complexity of GraphQL compared to a traditional REST API, we think that the advantages are worth it.

6.6.3 User authentication

Cookies and JSON Web Tokens (JWTs) are both commonly used for user authentication over the internet.

JWTs use token-based authentication, while cookies are use session-based authentication. This means that JWTs are stateless and independent of the server, while cookies are stored on the server and managed by the server. JWTs are encrypted using a symmetric cryptographic function where the secret key is stored on the server, so only the server can read the tokens contents, which is usually the user's email and an expiry date. This means that the server can be sure that the token has not been modified. If a token is expired or no token exists yet, the user can obtain one by logging in with their email and password.

Overall, JWTs are the slightly more secure option, so we opted to use them instead of session cookies.

6.7 Backend

The task of the backend is to store data, serve data to users when requested, and execute the simulation algorithms. There is an abundance of backend frameworks available. Nearly every programming language, from Python to Rust, Go, JavaScript, and even C[32], can nowadays be used to implement a backend server. This plenitude of choice allows us to select a framework that specifically suits our needs.

6.7.1 Language

Since all simulations - like shadow case, noise propagation and site suitability - are all written in Python, the choice was clear to use Python since it allows seamless compatibility with the existing scripts. We can simply use Python's ProcessPoolExecutor to schedule simulations on the server.

6.7.2 Framework

One of the most popular GraphQL frameworks for Python is Strawberry[33]. It integrates with FastAPI[34], which is the underlying web framework that hosts the GraphQL API. We chose to use it because Strawberry is modern, easy to learn, and extremely well documented.

6.7.3 Database

The two most common database architectures are the following:

[32]Facil: www.facil.io
[33]Strawberry: www.strawberry.rocks
[34]FastAPI: www.fastapi.tiangolo.com/

- **Relational**: This is the most common type of database architecture, and is used by most traditional SQL databases. In this architecture, data is organized into tables, with each table consisting of rows and columns. Tables can be related to one another using primary and foreign keys, allowing for complex data relationships.

- **Document-based**: This type of architecture is also called "NoSQL", and is based on JSON. In this architecture, data is organized into documents, which can have attributes and methods. This allows for more flexible and complex data structures than in a relational database.

We decided to go with a relational SQL database called PostgreSQL[35] in combination with an object-relational mapper called SQLAlchemy[36] Using an object-relational mapper allows us to abstract most complexities associated with SQL away so that we can use database entries as native python objects, which are then mapped to database operations.

6.7.4 Load balancing and security

Instead of exposing the GraphQL application directly to the web, a reverse proxy Nginx server is used. Nginx[37] is one of the most used web server software in the world. It is secure, handles HTTPS signing, does load balancing and protects against denial-of-service attacks.

[35]PostgreSQL: www.postgresql.org
[36]SQLAlchemy: www.sqlalchemy.org
[37]Nginx: www.nginx.com

7 Planning application implementation

This section presents and describes the final wind farm planning application, that has been implemented according to the architecture described in Section 6. The implemented pages will be shown, and the workflows will be described.

7.1 Design decisions

Let's discuss the most important design decisions that impact the entire application.

7.1.1 Navigation

We opted for a permanent side sidebar on the left side of the screen, since it is a very common design pattern. Since users spent most of their time on other websites (see Jakob's law in Section 5.1), users get used to the application quicker if it uses elements that they already know.

Since mobile phones are not a development target, the permanence of the sidebar is not an issue because of the sufficient screen estate, and provides the benefit of allowing the user to quickly jump to different views of the application. It reacts to the authentication status of the user, and displays links accordingly. Compare Figure 28 and for example 32 to see the difference. It is also collapsible, to take up less space if needed.

7.1.2 View layout

The overall layout of the application can be seen in Figure 29. Next to the sidebar is the scrollable page view area, which takes up the rest of the screen. It has a large header to tell the user on which page they are, with an optional back arrow button to get to the previous page if possible. Content is always semantically grouped in cards, which can be seen all over the application. Fitts's law describes the speed-accuracy tradeoff of the human motor system. As a consequence, we designed the cards to be left-aligned, such that the distance to the sidebar is minimized, again, as seen in Figure 29. All over the application, we paid attention to the mouse movements during typical workflows, and tried to minimize the travelled distances where it made sense.

7.1.3 Dark mode

Since our users are mostly professionals, it is likely that they spend a lot of time in front of their displays. Interfaces with a dark theme have been proven to reduce visual fatigue [24], so we have designed every view of the application in two themes, a standard light mode, and a dark mode. Figure 29 shows both modes with the example of the landing page. The choice whether to show the light or dark mode depends on what the browser or system default setting is. But after a logged-in user explicitly changes the dark mode setting in the application settings, the system setting will be ignored.

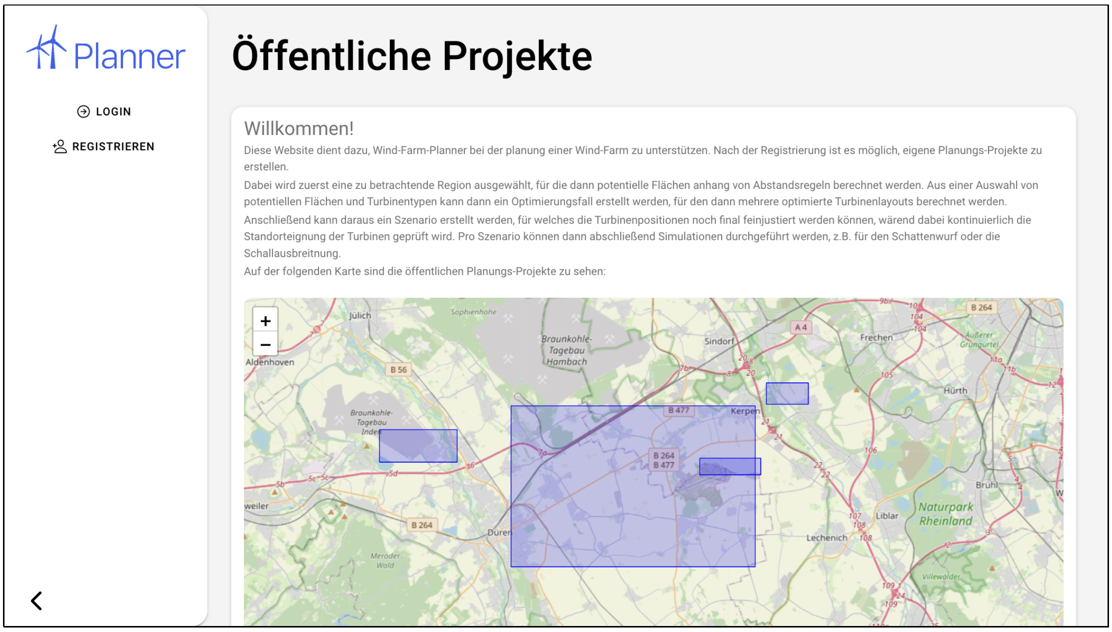

Figure 28: The landing page is the first page a new user will see when opening the application.

We will not present both versions of every page, due to the amount of space it would take, but only show some pages in the dark mode version.

7.2 Landing page

Figure 28 shows the landing page. It has a "Welcome" section, where the user is greeted, and a short text explains the main purpose of the application. Below, a map containing all public planning projects is displayed. These are projects that are free for anyone to view, which is useful if a public institution wants to make their project visible to any planner.

7.3 User accounts and administration

The application has standard user account functionalities, which include:

- **Register and login pages**: Users can register an account, and then use the login page to log in to their account with their email and password.
- **Profile and application settings page**: The profile page allows users to manage their account, including changing their password. The application settings page allows users to customize the application to their needs, for example by toggling the dark mode on or off.

To keep this section concise, we only show the register and profile settings page in Figure 29.

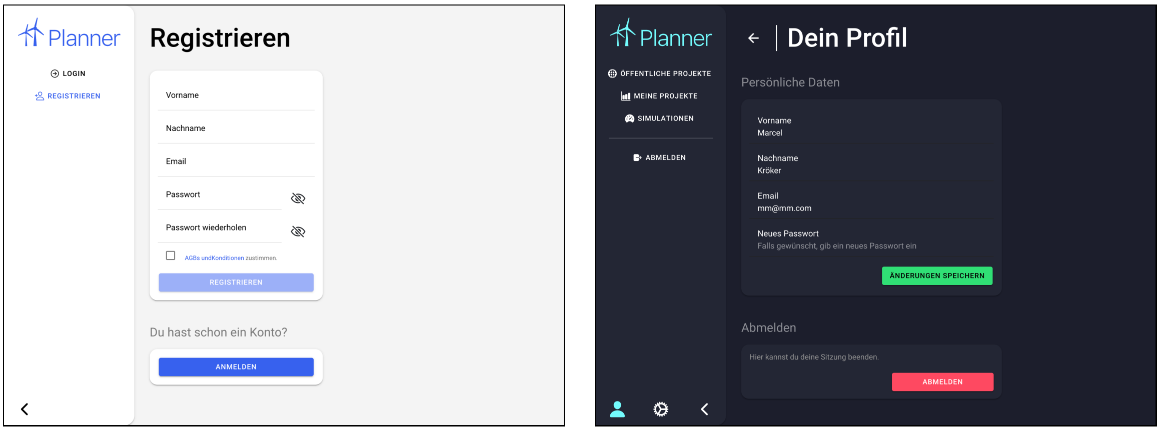

Figure 29: On the left, the register page in light mode is shown. On the right, the profile settings page in dark mode is shown.

There is also an administrative interface, which only users with the administrator role can access. We will not get into details about it since it is not very relevant for the core application functionality, but here are the basic functions of it:

- **Admin overview**: An overview page that links to all sub-pages.
- **User management**: A simple overview of all registered users, where they can be edited or deleted.
- **Turbine management**: A page to manage the turbine database. It allows administrators to create new wind turbine types, and edit existing ones.

7.4 Planning Projects

The most important part of the application is the project workflow. After giving a summary, we will show the relevant pages and describe more of their details. Figure 30 visualizes how all pages of the project workflow are interconnected. At first, the user creates the project, where they enter the required project information and select an overall region that should be considered. At the creation, the server will start downloading the region's GIS and wind data, and the user is redirected to the project details page. There, they will see a loading indicator until the downloads finish. Afterwards, potential areas can be generated in the compute areas page. Potential areas are sub-regions in the overall project region, which have a minimum required distance to residential buildings. The user can compute potential areas with different minimum distances to estimate different regulators conditions. When the user is happy with the resulting potential areas, they can use them to create an optimization case. An optimization case contains a selection of potential areas, and a selection of turbines from which a layout should be computed. On creation, the optimal positioning algorithm is started in the backend, which tries to place turbines from the selection inside the

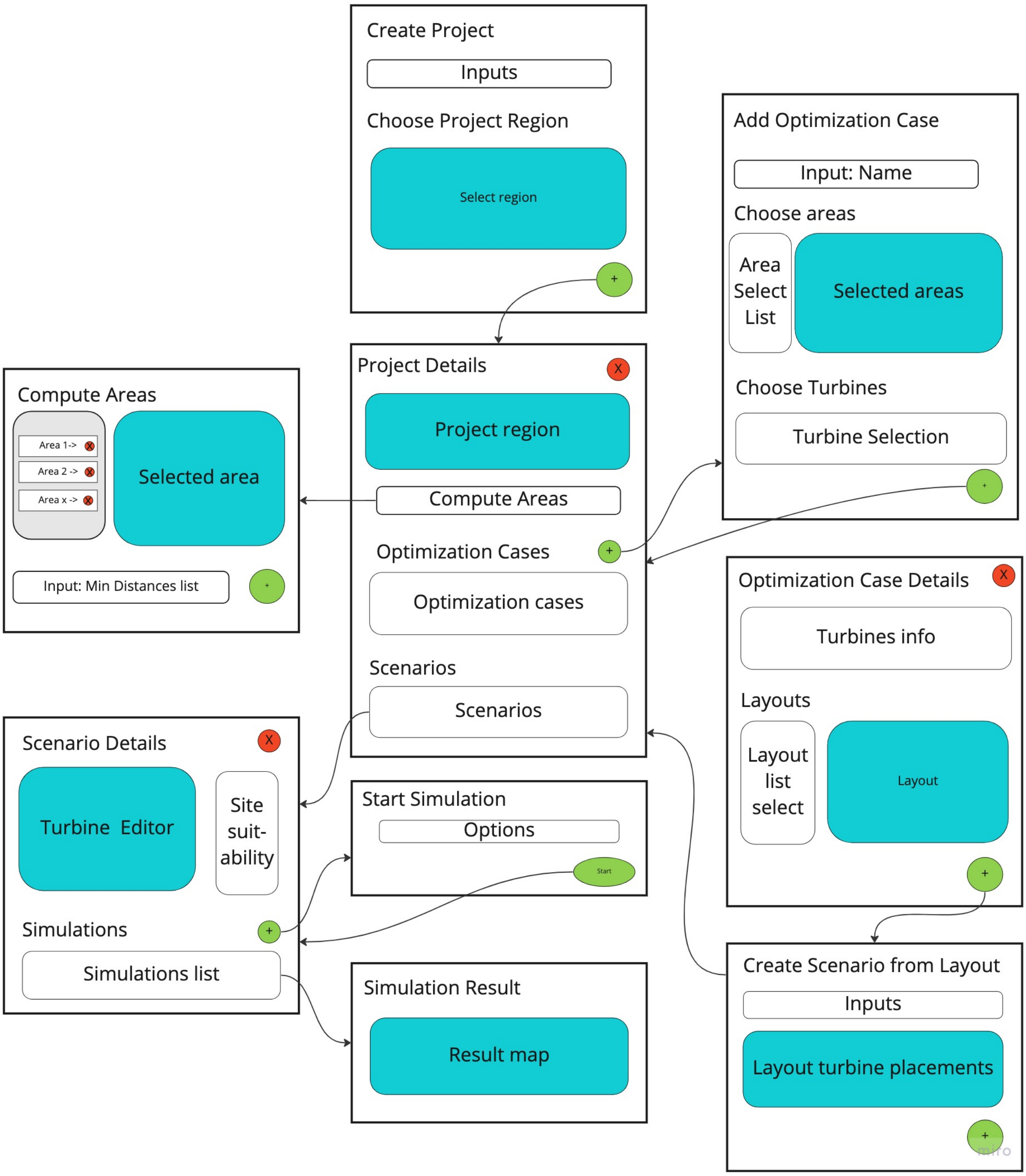

Figure 30: The overall project workflow. Each square box represents a page, and rounded boxes represent interface components. A light blue background color means that the component is a map.

selected potential areas of the optimization case. When viewing the optimization case, the user can inspect the different layouts. Finally, the user can create a simulation scenario from one of those wind turbine layouts. On the scenario details page, the wind turbines can be moved around, removed, or new ones can be added. After each modification, the site suitability model should be applied to the wind farm, and on the right side of the editor, a list of potential problems is displayed, e.g., if a turbine's design extreme wind speed is exceeded. Using the scenario, the shadow cast, and noise propagation algorithms can be executed.

In the overall design process of the workflow, we paid attention to Hick's law and Miller's law, from Section 5.1. Each view is kept concise and usually only has one specific task it does, as to not overload the user's working memory. Decisions are isolated from another, so a user can concentrate on the given task at hand and make decisions quicker.

7.4.1 Projects overview and project details

The project overview page simply shows all projects of a user, as seen in Figure 31 (left). There are a few key information about all projects at the top, presented with info cards. The project overview card has an action area on the top right. Here, a button to create a new project is displayed.

In Figure 31 (right), we can see the details page of a project. It shows some basic information, and links to the compute area view, the optimization cases and the scenarios.

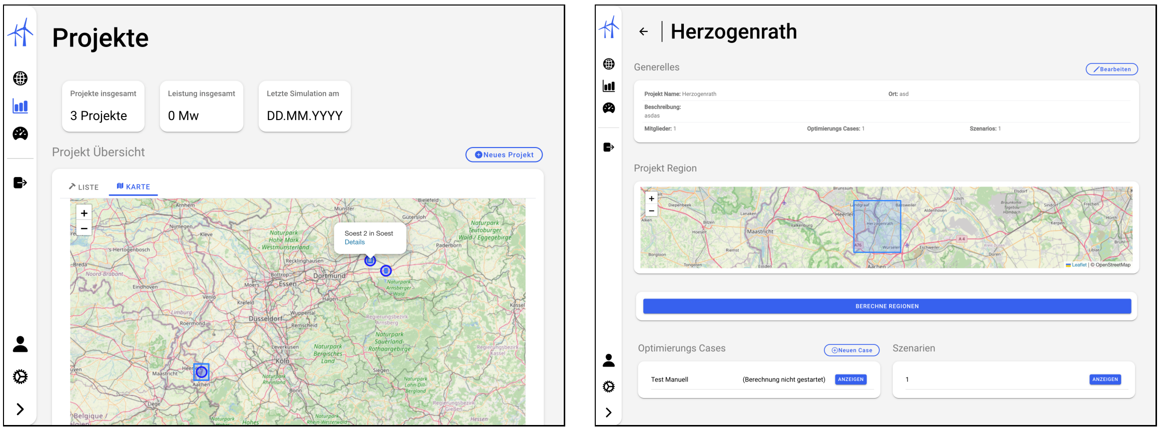

Figure 31: On the left, we see the projects overview page. It has a tab switcher which makes it possible to view all projects either in a list, or on a map. On the right, we can see the detail view of a project.

7.4.2 Create new project

When creating a new project, the user has to provide some basic information about the project, like the title, a description, the location and if it should be public. Public projects will be visible on the landing page for everyone. Afterwards, the project region must be selected. This is done by setting two corner points of a rectangle, as seen in Figure 32. On the map is an interactive selection tool, which allows choosing the current node and then set it by clicking on the map. This is a bit unusual, but we found that it allows better precision than just dragging over the map to select the region. To make the selection even more accurate, the latitude and longitude of both nodes can also be entered manually.

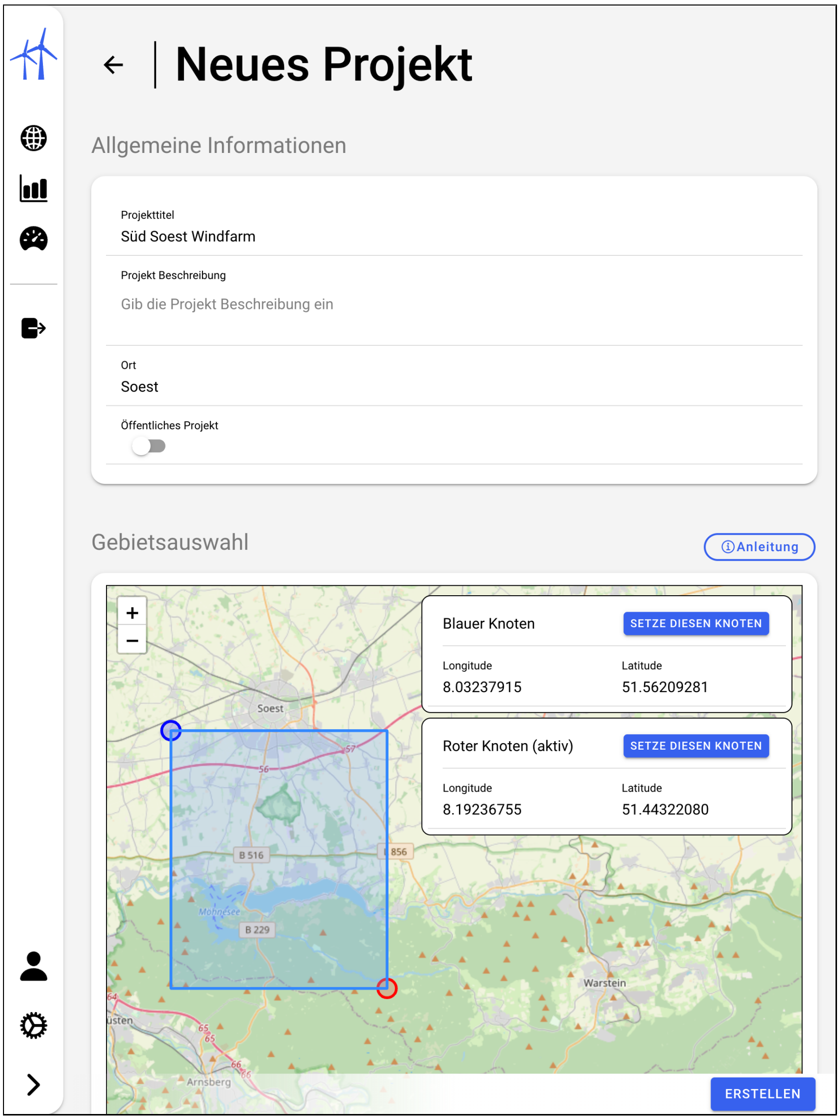

Figure 32: On this page, a new planning project can be created.

7.4.3 Compute potential areas in a project region

From the details page of a project, the user can go to the potential area page. A potential area is a sub-area of the entire project region, where the minimum distance requirement to residential areas is fulfilled. Depending on the region, these can be multiple scattered non-connected sub-areas in the project region. On the potential area computation page as seen in Figure 33, we can see a list of the already computed

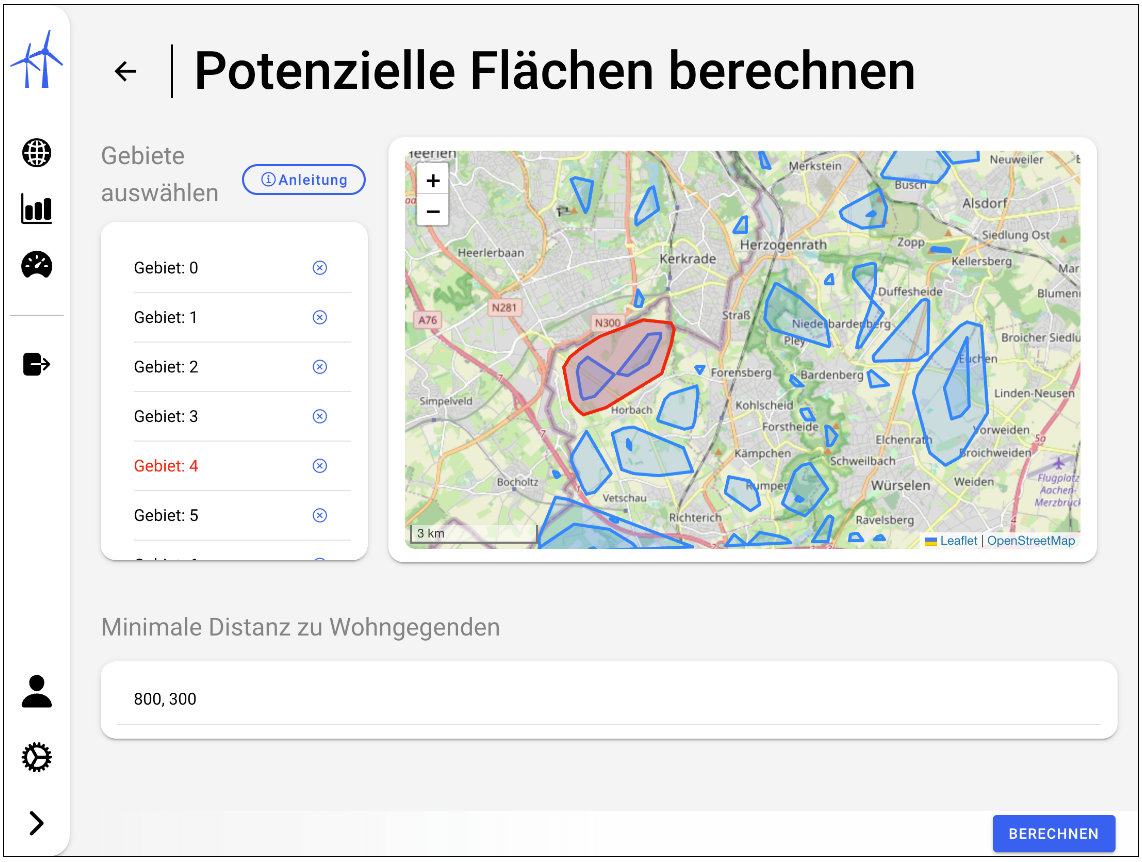

Figure 33: The page which is used to manage and compute new potential areas of a project.

potential areas, including a map that displays them. When an area is selected on the list, it will be highlighted on the map as well. If the planner wants to discard some areas, they can delete them from the project. On the bottom, an input field exists, where a list of minimum distances can be entered. When submitting, the server starts to compute new potential areas with the new distance regulations. This allows planners to consider multiple possible regulatory conditions for planing future wind farms, possibly anticipating reduced limits in the future.

7.4.4 Adding and viewing optimization cases

After the user is satisfied with the set of potential areas they computed, they return to the project details page. Next up, they can add optimization cases to their project. An optimization case is created from a selection of one or multiple computed potential areas, a selection of turbine types and maximum counts for each turbine and an overall maximum count, as seen in Figure 34. On creation, the server start computing multiple sets of optimized turbine placements, called wind farm layouts. When clicking on an optimization case on the project details page, the user is directed to the optimization details page, as seen in Figure 35. Here, the selected turbine types and maximum

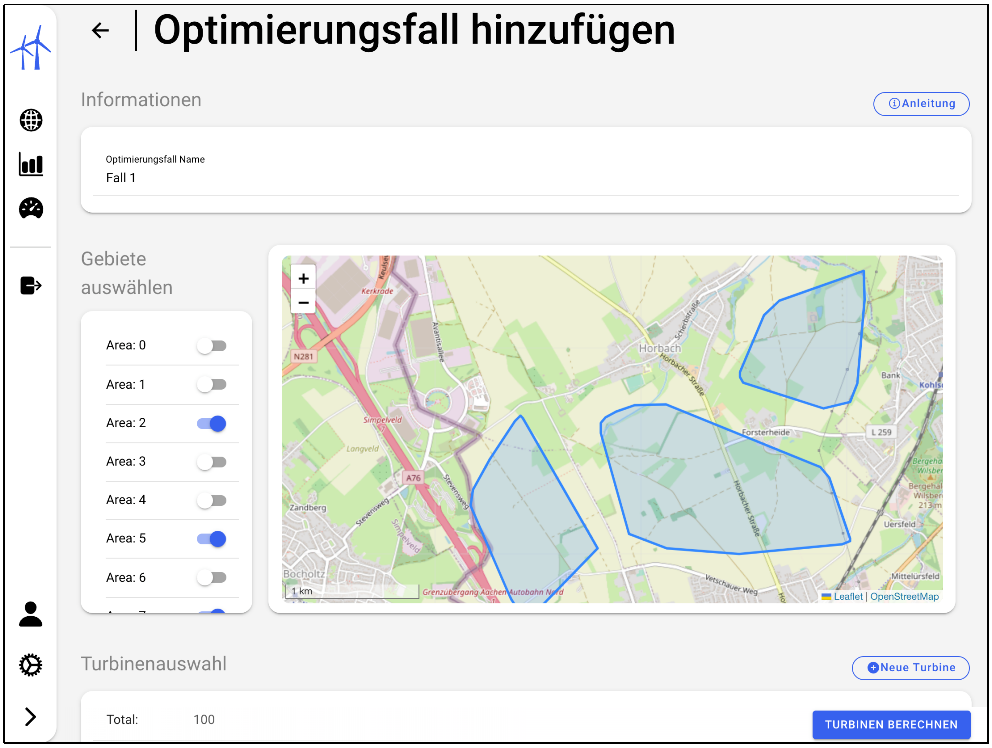

Figure 34: The page which is used to manage and compute new potential areas of a project.

counts are displayed. Below that, the optimized turbine layouts are shown on a map. On the side is a list to select which layout should be shown, with the optimization case's selected areas shown as a reference.

7.4.5 Creating scenarios

When a user inspects the optimized wind farm layouts on the optimization case details page, they can decide to create a new simulation scenario from that. This is done by selecting a given layout, and then clicking the button which creates a scenario from the layout. Here, the user first only sets the name of the scenario, which appears in the scenarios list of the project details page after submitting. Figure 36 shows how a scenario is created. When going to the scenario details, the user first sees a map with a turbine editor and a site suitability reporting section. Below that is a list of all simulations that have been executed or are ongoing.

7.4.6 Editing scenarios and checking site suitability

The scenario details page has a turbine editor at the top, as seen in Figure 37. The editor allows planners to move turbines around if they need to modify the layout. A precision moving tooltip is shown when a turbine is selected. The editor also allows users to add new turbines, or change the type of a turbine. When a change is made

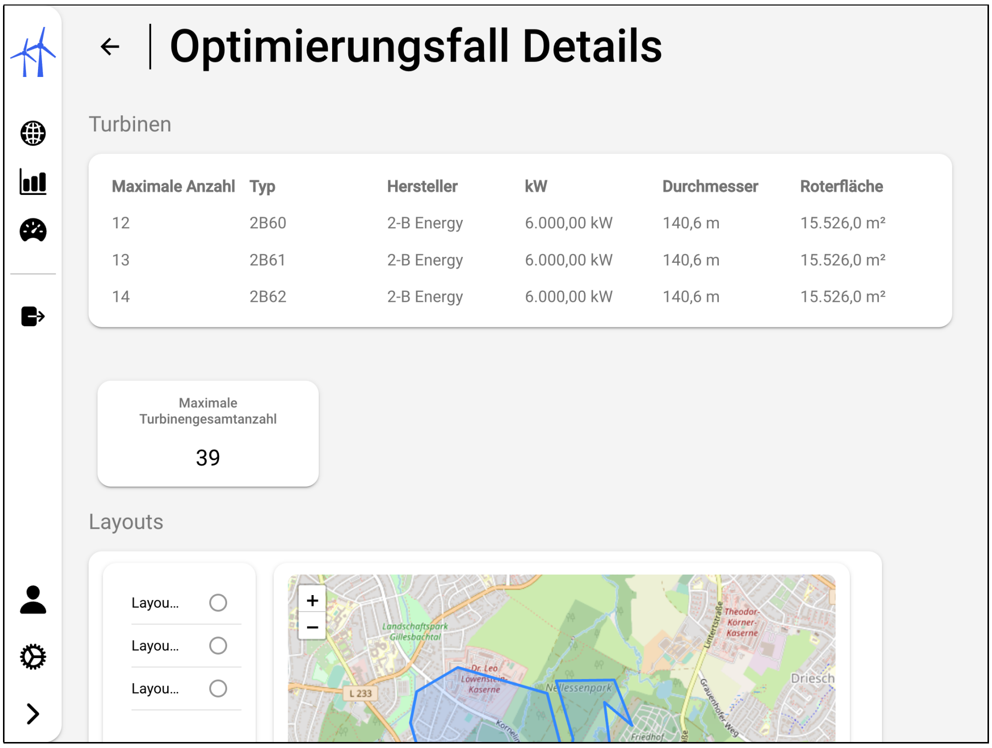

Figure 35: The page which is used to manage and compute new potential areas of a project.

to a turbine placement, the server is requested to run our site suitability algorithm for each turbine, and the result will be displayed on the right side in the information section. With this design, it is straightforward to find out quickly if a given turbine layout is feasible in relation to the governmental regulations and the turbine's design limits.

7.4.7 Dispatch simulations

At last, when the planner is happy with their turbine layout, they can decide to run shadow cast and noise propagation simulations on it. This is done from the simulations section of the scenario details page. Those views exist, but cannot visualize the simulation results yet. But they can be implemented using the simulation outputs, which are also convex hulls, just like the potential areas.

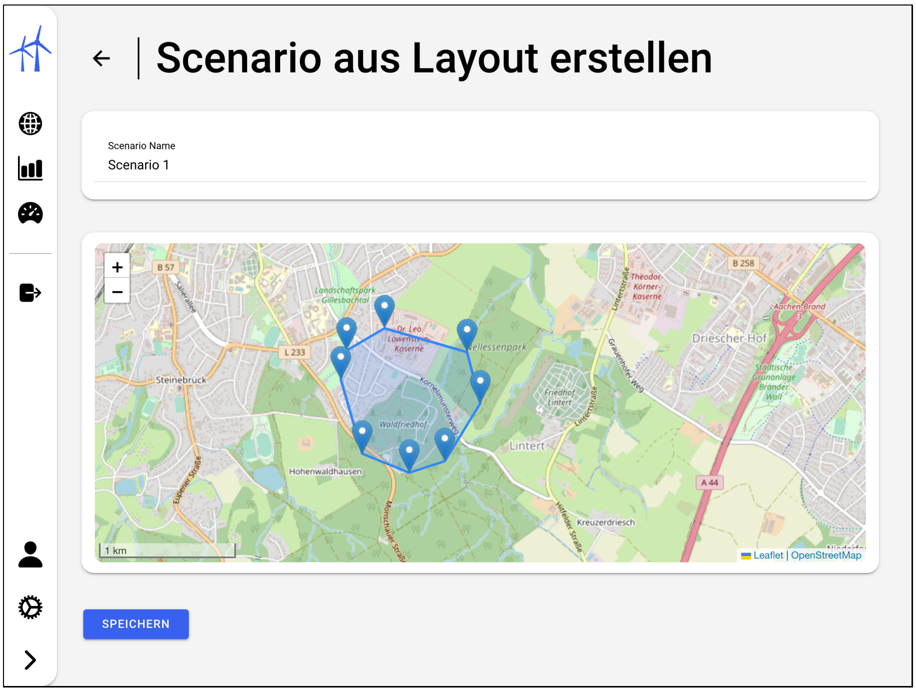

Figure 36: A simulation scenario is created from a wind farm layout that was optimized from an optimization case

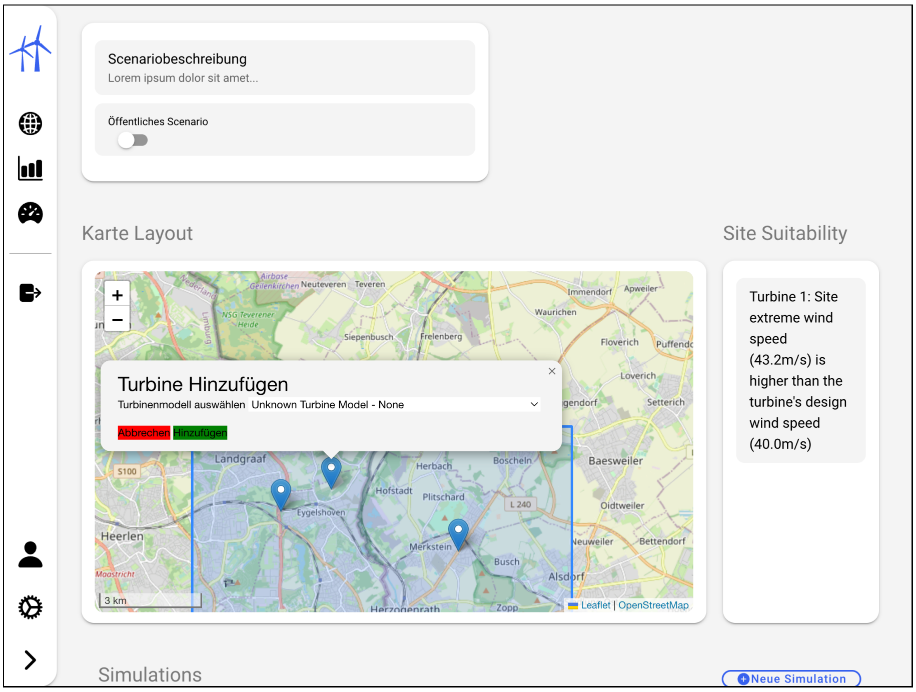

Figure 37: The scenario details page, which also contains the turbine layout editor, and the site suitability column.

8 Conclusions

In this work, we set out to address the problems associated with the planning process of wind farms. Through extensive research and development, we have succeeded in achieving our goals.

First, we developed a model and implementation for evaluating the site suitability of a wind turbine in relation to topography, wind, turbulence, and local regulations in Germany. This is a crucial step in the planning process, as it allows planners to determine whether a particular wind turbine type is suitable for a given site. By knowing this information early on, costly and time-consuming iterations of professional site assessments can be avoided, and wind farms can be realized more quickly.

Second, we created an interactive tool for wind farm planners that they can use to plan and validate their designs. This tool allows users to create projects containing different possible wind turbine placement layouts, and to evaluate various metrics such as site suitability, optimal layout, shadow cast, noise propagation, and site suitability. By using this tool, planners can save time and effort by performing these computations quickly and easily within a single platform.

Finally, we placed a strong emphasis on user experience, examining user interface design principles and applying them to our tool in order to make the planning process as intuitive and easy as possible. By doing so, we have helped to make the tool more accessible and user-friendly, further improving the efficiency and effectiveness of wind farm planning.

In conclusion, we hope that our work has made a positive contribution to improving the process of wind farm planning. Through the development of a site suitability model and implementation, as well as an interactive planning tool, we have helped to streamline the planning process, making it faster and more cost-effective. By doing so, sustainability goals can be realized sooner, which helps us to pave the way for a more sustainable future.

8.1 Future work

Several things could still be done to improve the site suitability algorithm and the planning application:

1. **Integrate site suitability model with application**: Because the application and the site suitability model were developed in parallel, there was not enough time to integrate the site suitability model into the planning application. However, the interface elements already exist, so it is only a matter of adding the required endpoints in the backend and connecting the frontend to it.

2. **Probable suitability**. Local measurements of wind data are not available in the planning application, so we resort to data which is available online. To account for that, the model could be changed so that the criteria have a transition zone around the limits, so that a certain margin of error is allowed and tells the user that a site is only probably suitable.

3. **Test on multiple devices** The application should be built for each platform and get tested in detail. Android and iOS apps need to be tested, especially the appearance of the design needs to be checked if it stays consistent on every device.

4. **Localization** The application is currently only planned to be used in Germany. A next step would be to translate the application to multiple languages, if non-german-speaking wind farm planners should be targeted.

5. **Layout**. If mobile phones should become a development target in the future, then the fixed sidebar should be replaced with a top or bottom toolbar to free up already limited horizontal space.

6. **Report Generator**: Another idea is to add a report generator, which outputs a PDF for a planned project, listing all results from the site suitability, noise propagation and shadow cast algorithms.

References

[1] Enercon e-126 7.580, . URL https://en.wind-turbine-models.com/turbines/14-enercon-e-126-7.580#powercurve.

[2] Siemens gamesa sg 14-222 dd, . URL https://www.siemensgamesa.com/products-and-services/offshore/wind-turbine-sg-14-222-dd.

[3] Richtlinie für windenergieanlagen. URL https://www.dibt.de/fileadmin/dibt-website/Dokumente/Referat/I8/Windenergieanlagen_Richtlinie_korrigiert.pdf.

[4] Din en 1991-1-1-4: Eurocode 1: Einwirkungen auf tragwerke. . URL https://www.niedersachsen.de/download/80801/Nds._MBl._Nr._37_2012_vom_30.10.2012_Anlagenband_37b.pdf.

[5] Din en 1991-1-1-4: Eurocode 1: Einwirkungen auf tragwerke nationaler anhang. . URL https://www.niedersachsen.de/download/80801/Nds._MBl._Nr._37_2012_vom_30.10.2012_Anlagenband_37b.pdf.

[6] Din en iec 61400-1. .

[7] Dauer und kosten des planungs- und genehmigungsprozesses von windenergieanlagen an land. URL https://www.erneuerbare-energien.de/EE/Redaktion/DE/Downloads/marktanalyse-windenergie-an-land-workshop-01-vortrag-fachagentur-windenergie-an-land.pdf.

[8] Electricity production by source, world. URL https://ourworldindata.org/grapher/electricity-prod-source-stacked.

[9] The fantanele and cogealac wind farm in romania. URL https://www.cez.cz/en/energy-generation/wind-power-plant/cezs-wind-power-plants/fantanele-cegealac-wind-park.

[10] Germany wants 100 percent green electricity by 2035. URL https://www.faz.net/aktuell/wirtschaft/klima-nachhaltigkeit/ukraine-krieg-bundesregierung-will-100-prozent-oekostrom-bis-2035-17839894.html.

[11] Gutachten zur standorteignung von windenergieanlagen nach dibt 2012 fuer den windpark elbe. URL https://uvp.niedersachsen.de/documents/ingrid-group_ige-iplug-ni/FB6F06D5-57E4-4902-9832-1F71492F440D/Turbulenzgutachten.pdf.

[12] Gutachten zur standorteignung von windenergieanlagen nach dibt 2012 für den windpark heidsiek. URL https://www.hameln-pyrmont.de/media/custom/3767_146_1.PDF?1645696482.

[13] Laws of ux. URL https://lawsofux.com.

[14] Minimum standards for determining turbulence conditions. URL https://www.wind-energie.de/fileadmin/redaktion/dokumente/publikationen-oeffentlich/beiraete/windgutachterbeirat/2016_Mindeststandards_Turbulenz_Hinweise_Zusammenfuehrung.pdf.

[15] MA Baseer, S Rehman, Josua P Meyer, and Md Mahbub Alam. Gis-based site suitability analysis for wind farm development in saudi arabia. *Energy*, 141:1166–1176, 2017.

[16] Adul Bennui, Payom Rattanamanee, Udomphon Puetpaiboon, Pornchai Phukpattaranont, and Kanadit Chetpattananondh. Site selection for large wind turbine using gis. In *PSU-UNS international conference on engineering and environment*, pages 561–566, 2007.

[17] Dale SL Dolan and Peter W Lehn. Simulation model of wind turbine 3p torque oscillations due to wind shear and tower shadow. *IEEE Transactions on energy conversion*, 21(3):717–724, 2006.

[18] Sten Frandsen. Turbulence and turbulence-generated structural loading in wind turbine clusters. 2007.

[19] Kira Grogg. Harvesting the wind: the physics of wind turbines. *Physics and Astronomy Comps Papers*, 7, 2005.

[20] Ryan Haac, Ryan Darlow, Ken Kaliski, Joseph Rand, and Ben Hoen. In the shadow of wind energy: Predicting community exposure and annoyance to wind turbine shadow flicker in the united states. *Energy Research & Social Science*, 87: 102471, 2022.

[21] T Ryan Haac, Kenneth Kaliski, Matthew Landis, Ben Hoen, Joseph Rand, Jeremy Firestone, Debi Elliott, Gundula Huebner, and Johannes Pohl. Wind turbine audibility and noise annoyance in a national us survey: Individual perception and influencing factors. *The Journal of the Acoustical Society of America*, 146(2): 1124–1141, 2019.

[22] Gundula Hübner, Johannes Pohl, Ben Hoen, Jeremy Firestone, Joseph Rand, Debi Elliott, and Ryan Haac. Monitoring annoyance and stress effects of wind turbines on nearby residents: A comparison of us and european samples. *Environment international*, 132:105090, 2019.

[23] Daniel Kahneman, Barbara L Fredrickson, Charles A Schreiber, and Donald A Redelmeier. When more pain is preferred to less: Adding a better end. *Psychological science*, 4(6):401–405, 1993.

[24] Kangsoo Kim, Austin Erickson, Alexis Lambert, Gerd Bruder, and Greg Welch. Effects of dark mode on visual fatigue and acuity in optical see-through head-mounted displays. In *Symposium on spatial user interaction*, pages 1–9, 2019.

[25] Andrew Lothian. Scenic perceptions of the visual effects of wind farms on south australian landscapes. *Geographical Research*, 46(2):196–207, 2008.

[26] James J McCarthy, Osvaldo F Canziani, Neil A Leary, David J Dokken, Kasey S White, et al. *Climate change 2001: impacts, adaptation, and vulnerability: contribution of Working Group II to the third assessment report of the Intergovernmental Panel on Climate Change*, volume 2. Cambridge University Press, 2001.

[27] Gorica Micic, Branko Zajamsek, Leon Lack, Kristy Hansen, Con Doolan, Colin Hansen, Andrew Vakulin, Nicole Lovato, Dorothy Bruck, Ching Li Chai-Coetzer, et al. A review of the potential impacts of wind farm noise on sleep. *Acoustics Australia*, 46(1):87–97, 2018.

[28] Nobuo Mimura. Sea-level rise caused by climate change and its implications for society. *Proceedings of the Japan Academy, Series B*, 89(7):281–301, 2013.

[29] Natalia Moskalenko, Krzysztof Rudion, and Antje Orths. Study of wake effects for offshore wind farm planning. In *2010 Modern Electric Power Systems*, pages 1–7. IEEE, 2010.

[30] JG Pedersen, E Svensson, L Poulsen, and NG Nygaard. Turbulence optimized park model with gaussian wake profile. In *Journal of Physics: Conference Series*, volume 2265, page 022063. IOP Publishing, 2022.

[31] Ernest W Peterson and Joseph P Hennessey Jr. On the use of power laws for estimates of wind power potential. *Journal of Applied Meteorology and Climatology*, 17(3):390–394, 1978.

[32] Andrey N Petrov and Jordan M Wessling. Utilization of machine-learning algorithms for wind turbine site suitability modeling in iowa, usa. *Wind Energy*, 18 (4):713–727, 2015.

[33] Henk Polinder, Jan Abraham Ferreira, Bogi Bech Jensen, Asger B Abrahamsen, Kais Atallah, and Richard A McMahon. Trends in wind turbine generator systems. *IEEE Journal of emerging and selected topics in power electronics*, 1(3):174–185, 2013.

[34] SRJ Powles. The effects of tower shadow on the dynamics of a horizontal-axis wind turbine. *Wind Engineering*, pages 26–42, 1983.

[35] Laura C Rodman and Ross K Meentemeyer. A geographic analysis of wind turbine placement in northern california. *Energy Policy*, 34(15):2137–2149, 2006.

[36] Auday H. Shaban, Ali K. Resen, and Nathalie Bassil. Weibull parameters evaluation by different methods for windmills farms. *Energy Reports*, 6:188–199, 2020. ISSN 2352-4847. doi: https://doi.org/10.1016/j.egyr.2019.10.037. URL `https://www.sciencedirect.com/science/article/pii/S2352484719308388`. Technologies and Materials for Renewable Energy, Environment and Sustainability.

[37] John Dalsgaard Sørensen, S Frandsen, and NJ Tarp-Johansen. Effective turbulence models and fatigue reliability in wind farms. *Probabilistic Engineering Mechanics*, 23(4):531–538, 2008.

[38] Jawad S Touma. Dependence of the wind profile power law on stability for various locations. *Journal of the Air Pollution Control Association*, 27(9):863–866, 1977.

[39] Noam Tractinsky, Adi S Katz, and Dror Ikar. What is beautiful is usable. *Interacting with computers*, 13(2):127–145, 2000.

[40] Jon Yablonski. *Laws of UX - Using Psychology to Design Better Products & Services*. O'Reilly, 2020.

[41] Wei Zhou, Li Li, Min Luo, and Wu Chou. Rest api design patterns for sdn northbound api. In *2014 28th international conference on advanced information networking and applications workshops*, pages 358–365. IEEE, 2014.