

Machine Learning meets Mathematical Optimization to predict the optimal production of offshore wind parks

Martina Fischetti and Marco Fraccaro

^a *Vattenfall BA Wind and Technical University of Denmark, DTU Management Engineering, Produktionstorvet 426 B DK-2800 Kgs. Lyngby martina.fischetti@vattenfall.com*

^b *Technical University of Denmark, DTU Compute, Richard Petersens Plads, DK-2800 Kgs. Lyngby marfra@dtu.dk*

Abstract

In this paper we propose a combination of Mathematical Optimization and Machine Learning to estimate the value of optimized solutions. In particular, we investigate if a machine, trained on a large number of optimized solutions, could accurately estimate the value of the optimized solution for new instances. In this paper we will focus on a specific application: the offshore wind farm layout optimization problem. Mixed Integer Programming models and other state-of-the-art optimization techniques, have been developed to solve this problem. Given the complexity of the problem and the big difference in production between optimized/non optimized solutions, it is not trivial to understand the potential value of a new site without running a complete optimization. This could be too time consuming if a lot of sites need to be evaluated, therefore we propose to use Machine Learning to quickly estimate the potential of new sites (i.e., to estimate the optimized production of a site without explicitly running the optimization). To do so, we trained and tested different Machine Learning models on a dataset of 3000+ optimized layouts found by the optimizer. Thanks to the close collaboration with a leading company in the energy sector, our model was trained on real-world data. Our results show that Machine Learning is able to efficiently estimate the value of optimized instances for the offshore wind farm layout problem.

Keywords: Machine Learning, Mixed Integer Linear Programming, Wind Farm Optimization, Green Energy

1. Introduction

Mathematical Optimization (MO) and Machine Learning (ML) are two closely related disciplines that have been combined in different way. A very popular application of the two together is the so-called Prescriptive Analytics field [1], where ML is used to predict a phenomenon in the future, and MO techniques are used to optimize an objective over that prediction. A popular research area is now also to use Machine Learning to improve heuristics decisions in Mixed Integer Linear Programming (MILP) algorithms, for example in the branching procedure [2, 3, 4] or in decomposition techniques [5].

The paper by [6] introduces a framework to tackle combinatorial optimization problems using neural networks and reinforcement learning; the authors study the traveling salesman problem and train a recurrent network to predict a probability distribution over different solutions. In [7] the authors investigate instead how to learn a heuristic algorithm by exploiting the structure of the instances of interest; the resulting approach is applied to different optimization problems over graphs, namely the minimum vertex cover, the maximum cut and the traveling salesman problems. The work of [8] integrates ML and tree search to derive a truncated branch-and-bound algorithm using bound estimates; the proposed method, called deep-learning assisted heuristic tree search, is applied to the so-called container pre-marshalling problem. The problem of deciding at which node of a branch-and-bound tree a heuristic should be run is instead addressed in [9], where ML is used to predict whether a certain heuristic would improve the incumbent if applied at a given branching node. Finally, the very recent special issue [10, 11] contains a collection of papers that combine ML with constraint programming/optimization. In particular, the work of [12] studies the applicability of Boolean optimization (maxSAT) to clustering problems, while [13] uses integer linear programming for learning Bayesian network structures, and [14] investigates an architecture for acquiring constraint programming constraints from classified examples.

In the present work, we will instead investigate a different way to merge MO and ML, where the slow MO model comes first, and its (almost) optimal solutions are used as training set for a Machine Learning (ML) algorithm that can quickly estimate the

value of new optimized solutions. We then show that a ML model can leverage data collected by running state-of-the-art MO algorithms on a specific application to learn to predict how MO would perform in new instances of the same task, without the need to run the computationally expensive MO algorithm every time. In this work, we will focus on a specific application, already studied by the first author in [15], namely the offshore wind park layout optimization problem. This idea can however be used for many different optimization problems and applications, such as transport, logistics and scheduling.

The *wind farm layout optimization problem* consists in finding an optimal allocation of turbines in a given site, to maximize the park power production. Different techniques have been studied to solve this optimization problem; the reader is referred to, e.g. [16, 17, 18] for some heuristic methods, and to [19] for a more detailed literature overview on the problem. In this work we will use the mathematical optimization framework proposed by [15] as this is the tool currently used by our industrial partner Vattenfall BA Wind. Nevertheless, we expect our ML algorithm to perform similarly if trained on other analogous optimization models.

A particularly challenging feature of the wind farm layout problem is the interaction between turbines, also known as *wake effect*. The wake effect is the interference phenomenon for which, if two turbines are located one close to another, the upwind one creates a shadow on the one behind (see Figure 1). This is of great importance in the design of the layout since it results into a loss of power production for the turbine downstream, that is also subject to a possibly strong turbulence. For many years, this issue has been unknown or underestimated, and old wind parks have been designed with a very regular (and highly wake-affected) layout. Many of the parks operating today are on a regular layout. It was estimated in [20] that, for large offshore wind farms, the average power loss due to turbine wakes is around 10-20% of the total energy production.

It is then obvious that power production can increase significantly if the wind farm layout is properly optimized. The large size of the problem, the complexity of the wake effect and the presence of other constraints, makes it impossible to create a good layout without the usage of an advanced optimization tool. Since the difference in power



Figure 1: Wake effect on Horns Rev 1 wind park. Many offshore wind parks operating today are still on a regular grid, and so largely affected by wake effect. *Source: [21]*

production between optimized solutions and unoptimized ones can be significant, it is even difficult to estimate the potential power production of a site, without running a complete optimization of the layout.

65 In this paper, we want to use a ML algorithm able to better estimate the potential of a site, without running a complete optimization. The MILP-based approach proposed in [15], takes about 10 hours for a complete optimization. Acceptable preliminary solutions, used for what-if analyses, may be found in 1 hour. Once a company has decided to invest on a site, this amount of time is not an issue, and the MO tools are used (1-hour runs for initial what-if analyses, and 10-hour runs for the final layouts). Nevertheless, there are other cases in which even a 1-hour optimization is not an option, due to the very large number of options to be compared. This happens, for example, when the site where to build the park is not decided a priori. Indeed, in the future it may be up to the constructing company to select the site where to build the park. The company would
75 then look for all the zones in the world with good wind conditions, and it will have to choose between a large amount of areas. Just assuming that 1000+ potential sites may be identified, running (even a fast 1-hour) optimization algorithm for all of them would be infeasible. A properly trained ML algorithm could instead be able to identify the most promising sites in a matter of minutes. Once the site is selected, a detailed (and
80 more time-consuming) optimization can be run to define the actual layout. A very fast tool to evaluate design choices is of key value for real-world park designers such as

Vattenfall, our industrial partner. Such fast tools can help them to investigate a huge number of possible design scenarios, and to better understand the full complexity of the wind park design process. In this work we will focus on the setting where a large
85 number of sites are available and we aim at having a fast ML algorithm to pre-select the most promising ones, and then run a full optimization on those only. In practice, equivalent ML+OR methods can be used for many different what-if analyses. One example could be the turbine model selection for a given site. So far, also due to the large optimization time needed, Vattenfall would only test two or three different tur-
90 bine types for each given site, in order to make an informed decision of which type to pick in the final layout. Given a fast ML+MO tool in the vein of the one we are here proposing, Vattenfall could potentially try hundreds of turbine types very quickly and even use this information to design a new site-specific turbine type. The overall goal of the present paper is therefore to show that such kind of tools can indeed be developed,
95 using the strength of both ML and MO.

Having in mind the site-selection application, we address the case where a company wants to construct a specific number of turbines in an offshore area. Even if the production would increase by spreading the turbines due to the wake effect, the infrastructure costs to connect turbines very far away from each other would also increase. Therefore
100 we assume that, even if a huge offshore area is available, the company would discretize it in a number of smaller rectangular sites. These sites can have different dimensions and the wind can largely vary from site to site. The potential of a site depends not only from the site itself but also on the turbine used. The company could also investigate the potential of different sites for different turbine types. We therefore considered rect-
105 angular instances of different dimensions, with different wind scenarios (taken from real-world parks) and with different turbine types. In this study we considered standard turbines already available on the market. We defined and optimized over 3000 instances using the MO tool developed in [15]. The power production of the heuristic solutions found is used as training set for our ML algorithm.

110 A distinguished feature of our work is that we do not expect to estimate the optimal *solution* (which is arguably very problematic for the state of the art), but we content ourselves with the estimate of the optimal *value* of it.

The paper is organized as follows. Section 2 gives an overview of basic energy concepts that can be useful to the reader to better understand the wind farm problem. Section 3 summarizes the MILP model used in the optimization phase. Section 4 defines the input instances used and studies the feature we will use for our ML model. Section 5 describes the alternative ML models we considered, namely Linear Regression, Neural Networks and Support Vector Regression. These models are tested on a large dataset and compared in Section 6. Finally Section 7 draws some conclusions and discusses possible extensions.

2. Basic wind energy notions

In this section we give a basic introduction to wind energy concepts that can help the reader to better understand the problem in hand. Our work can be valuable, indeed, both as a proof of concept on the possibility of estimating the value of optimized solutions using ML techniques, and from a practical application in the wind sector. For the reader interested in the second topic, we offer in this section a fast review of useful wind energy concepts. For further details, see [22].

The wind energy sector is a fast growing and more and more competitive sector. In order to build more profitable parks, it is very important to reduce costs and to increase incomes (i.e. increase power production). Nowadays, it is normally up to the State to define a specific site to construct a new offshore park, and different constructing/operating companies will enter a tender to have the rights to construct it. The company that can offer energy at the lowest cost will win the tender. A key feature to lower the KW/h price is to produce more energy at the same cost, and therefore the optimization of the layout (in order to reduce wake effect) is a key factor. In this context, MO techniques are used by many energy companies. Even if these techniques could require hours of computation, this is not a limitation, since the number of runs is quite limited. In this paper, instead, we look at a different context where we suppose that is up to the constructing/operating company to decide where to build the park. Supposing to have maximum freedom in this choice, the possible sites in a very large offshore area would be too many to run a complete optimization for each of them. In this context the

company would be interested in a first (fast) evaluation of the sites, in order to rank the best ones, and focus its attention on those. The way this fast (preliminary) evaluation of sites is done today is by evaluating the production of a regular layout, obtained by positioning turbines on a grid. It is nevertheless well known in the wind sector, that regular layouts are highly wake-affected, hence this estimate (as we will see later) fails at capturing the potential of new sites.

2.1. Wind turbines

Once a company has decided a site, it is a key factor to decide which kind of turbines to build in the site.

A wind turbine model is typically identified by its manufacturer (for example, Siemens, Vestas, Adwen etc.), by its dimensions and its power rating. The power rating is a measure of how powerful a turbine is. All turbines produce different power at different wind speeds: the higher the wind, the higher the production. A typical power curve is shown in Figure 2: the power production increases with wind and it saturates at a certain MW production. This value is the *rated power* of a turbine. Another interesting turbine specification is its rotor diameter, that identifies the area spanned by the turbine blades (see Figure 3). In general, the bigger the rotor, the higher the rated power of a turbine. The manufacturer can also design turbines with the same rated power but different rotor diameters. In this case, the difference mainly occurs at medium wind speeds (around 5 – 15 m/s), when a bigger rotor is able to catch more wind and produce more, even if at higher wind speeds the turbines are controlled to not produce more than the rated power. In Figure 4 we plot the power curve of two turbine models with the same rated power but different rotor diameters. It is therefore important, in the specification of a turbine model, to indicate also its rotor diameter.

It is then clear that, if we decide to build a fixed number of turbines, the turbine model selected will impact the total power production of the site. This is why, as we will see later, it is important to capture this informations both in the MO and in the ML models.

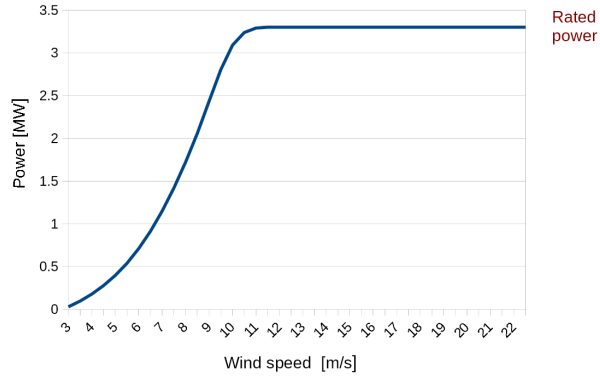


Figure 2: Example of power curve



Figure 3: Rotor diameter of a turbine. *Picture inspired by an illustration from [21]*

170 **2.2. Site and wind farm production**

In this paper we are considering the offshore wind farm design problem. The term *offshore* refers to the construction of wind farms in bodies of water, so the site is normally a regular area in a sea zone. This has some advantages compared to onshore sites, for example the wind is generally stronger and equally distributed. In addition,
 175 residents opposition to construction is usually much weaker for offshore parks, and the

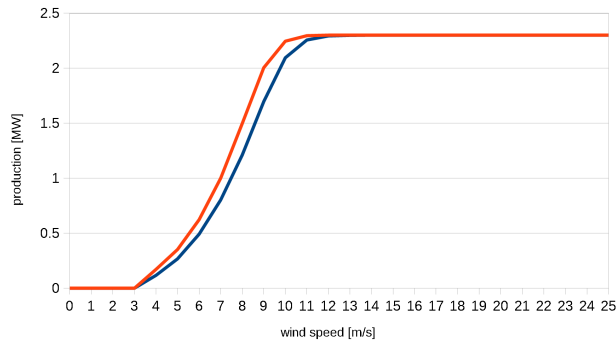


Figure 4: Example of power curve of two turbines with the same rated power but different diameters (101m in blue and 113m in red).

so-called NIMBY (“Not In My BackYard”) problem is easier to solve. In the offshore case, the height of the turbines above sea level is supposed to be equal everywhere, so the offshore case is actually a 2D case. This also means that, in offshore sites, we can consider the wind to be equally distributed all over the site, so the potential power production of all possible positions is the same.

A key role is then played by the interference between turbines, that causes a loss of production in some positions. In this paper we used Jensen’s model [23] to compute the interference caused by a turbine: the interference is modelled as a cone centered in the upwind turbine. The further away we go from the upwind turbine, the lower the interference. Figure 5 plots the interference cones for a wind blowing from South-East. In the plot, red dots represent built turbines, while the background colours represent the power loss [MW] due to interference (a turbine placed in a dark blue area, will not suffer any wake effect). Looking at this plot it is easy to imagine that regular layouts suffer from higher interference, since turbines often lay in the interference cone of other turbines.

The wake effect, of course, depends on wind intensity and wind direction. Nevertheless, once the position of the turbines (i.e., the layout of the park) is selected, the turbines cannot be moved depending on the wind. Therefore, when we optimize the turbine position, we have to consider all the possible wind scenarios with their probability (as we will more formally see in Section 3). Figure 6 shows the wake effect

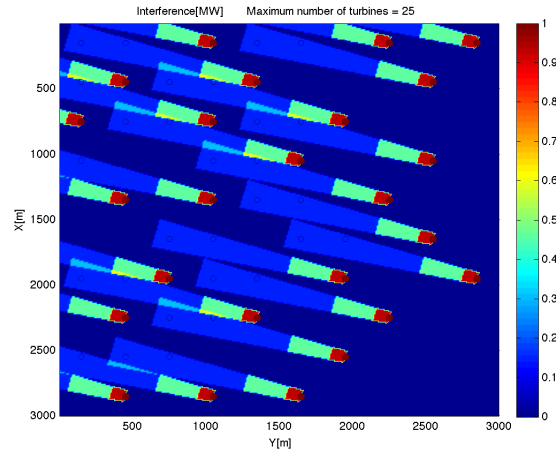


Figure 5: Wake effect. The red dots represent built turbines, the background colors represent the interference intensity according to Jensen's model (wind blowing from south-east)

values in a site, when considering all its different wind scenarios.

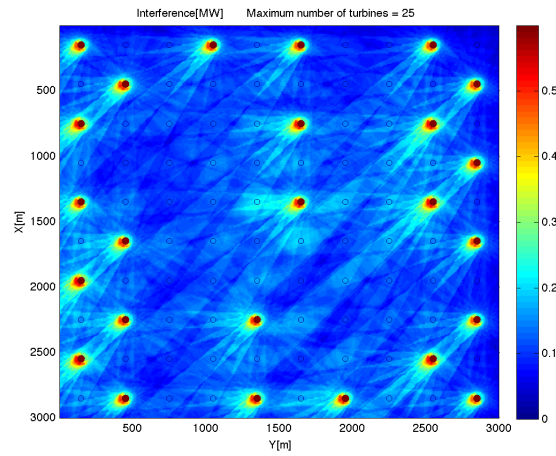


Figure 6: Wake effect. The red dots represent built turbines, the background colors represent the interference intensity according to Jensen's model (when considering wind statistics from a real-world park)

3. The optimization model

The aim of this section is to outline a MILP model for the offshore wind farm layout problem. Additional constraints and sophisticated algorithms to solve large scale instances have been developed by the first author (see [15]), but their description is out of the scope of the present paper.

At the optimization stage, an offshore site is given together with the wind statistics in the site. We are asked to determine the optimal allocation of turbines in the area, in order to maximize the park power production. The optimization needs to consider that a minimum and maximum number of turbines can be built, a minimal separation distance must be guaranteed between two turbines to ensure that the blades do not physically clash (turbine distance constraints), and the power loss due to wake effect is minimized.

We first discretize the area in a number of possible turbine positions. Let V denote this set and let

- P_i be the power that a turbine would produce if built (alone) at position i .
- N_{MIN} and N_{MAX} be the minimum and maximum number of turbines that can be built in the park, respectively;
- D_{MIN} be the minimum distance between two turbines;
- $dist(i, j)$ be the symmetric distance between positions i and j .
- I_{ij} be the interference (loss of power) experienced at position j when a turbine is installed at position i , with $I_{jj} = 0$ for all $j \in V$. We used a Jensen's model to compute it [23];

In addition, let $G_I = (V, E_I)$ denote the incompatibility graph with $E_I = \{[i, j] : i, j \in V, dist(i, j) < D_{MIN}, j > i\}$.

Note that the interference matrix I is not symmetric, as the loss of power due to interference experienced by i when a turbine is installed in site j depends on the relative position of i with respect to j but also on their relative position with respect to the wind direction.

In our model, we define binary variables x_i for each $i \in V$ as

$$x_i = \begin{cases} 1 & \text{if a turbine is built at position } i \in V; \\ 0 & \text{otherwise.} \end{cases}$$

225 The quadratic objective function (to be maximized) reads $\sum_{i \in V} P_i x_i - \sum_{i \in V} (\sum_{j \in V} I_{ij} x_j) x_i$ and can be restated as

$$\sum_{i \in V} (P_i x_i - w_i) \quad (1)$$

where

$$w_i := \left(\sum_{j \in V} I_{ij} x_j \right) x_i = \begin{cases} \sum_{j \in V} I_{ij} x_j & \text{if } x_i = 1; \\ 0 & \text{if } x_i = 0 \end{cases}$$

denotes the total interference caused by position i . Our MILP model then reads

$$\max \quad z = \sum_{i \in V} (P_i x_i - w_i) \quad (2)$$

$$\text{s.t.} \quad N_{MIN} \leq \sum_{i \in V} x_i \leq N_{MAX} \quad (3)$$

$$x_i + x_j \leq 1, \quad \forall \{i, j\} \in E_I \quad (4)$$

$$\sum_{j \in V} I_{ij} x_j \leq w_i + M_i (1 - x_i), \quad \forall i \in V \quad (5)$$

$$x_i \in \{0, 1\}, \quad \forall i \in V \quad (6)$$

$$w_i \geq 0, \quad \forall i \in V. \quad (7)$$

where the big-M terms $M_i = \sum_{j \in V: [i, j] \notin E_I} I_{ij}$ are used to deactivate constraint (5) in case $x_i = 0$. This model works on larger instances compared with equivalent models
230 in the literature involving 2-index variables $y_{ij} = x_i x_j$; see [22] for details. Note that constraint (3) can be used to impose the construction of a fixed number of turbines by setting $N_{MIN} = N_{MAX}$.

Of course, the power production P_i and the interference value I_{ij} vary with the wind. Using statistical data, one can in fact collect a large number, say K , of wind scenarios
235 k , each associated with its own P_i^k, I_{ij}^k and with a probability π_k . As shown in [15], one can then take wind scenarios into account by simply defining $P_i := \sum_{k=1}^K \pi_k P_i^k$ ($i \in V$) and $I_{ij} := \sum_{k=1}^K \pi_k I_{ij}^k$ ($i, j \in V$).

To solve large-scale instances (with 20 000+ possible positions) some ad-hoc heuristics and a MILP-based proximity search [24] heuristic have been used on top of this basic model. We refer the interested reader to [15] for details.

4. Data Generation

We used the model presented in Section 3 to determine the optimized power production of a large number of realistic instances. These instances have been artificially created by considering rectangular areas of different sizes, different real-world turbine types, and different wind statistics from different real-world sites. In particular, we generated different sites by generating sets of possible points on a regular grid (10m point-to-point distance) inside rectangles of different dimensions (all possible combinations of edge sizes 6000, 7000, 8000, 9000, 10000, 11000, 12000, 13000 and 14000m). We computed power production and interference based on the data from the following real-world turbines:

- Adwen 8 MW, with a rotor diameter of 180m;
- Vestas 8.4 MW, with a rotor diameter of 164m;
- Siemens 7 MW, with a rotor diameter of 154m;
- Vestas 8 MW, with a rotor diameter of 164m;
- Siemens 3.2 MW, with a rotor diameter of 113m;
- Siemens 2.3 MW, with a rotor diameter of 101m.

For each line of the list above we have specified the turbine manufacturer (Adwen, Vestas, Siemens), the rated power of the specific model (i.e. the maximum MW power that the turbine can produce), and the turbine rotor diameter. Note that different rated powers and different rotor diameters affect the power production P_i and the interference $I_{i,j}$ of each turbine and therefore the total power production of the park. Finally, we considered different real-world wind statistics for the wind scenarios, namely those from the real offshore wind parks named Borssele 1 and 2, Borssele 3 and 4, Denish

Krigers Flak, Ormonde, Hollandse Kust Zuid and Horns Rev 3. These are in-operation
265 or under-construction parks located in The Netherlands, Denmark, and United King-
dom. By considering all the possible combinations of sites, turbine types and winds,
we obtained about 3000 instances. We imposed that a fixed number of 50 turbines
needs to be located in the site. For each instance we computed:

- 270 1. Grid production: the power production of a solution obtained by locating the
turbines on a regular grid; and
2. True production: the (almost) optimal power production computed by our MO
tool.

We decided to compute the production of a layout on a regular grid for two reasons.
First, this represents how a manual operator would evaluate the potential of a site,
275 and so it acts as a benchmark value for our ML models. Secondly, this value is very
informative: as we will better see in Section 4.1, it captures the dimension and the wind
of the site, while providing a lower bound for our optimized solution.

It is important to observe that the grid production requires short computing time
and can be calculated in a pre-processing step. Optimization for the difficult case (true
280 production) was instead obtained through the MILP-based heuristic of [15], with a time
limit of 1 hour on a standard PC using IBM ILOG CPLEX 12.6.

The output of the optimization has been used to train the ML models presented
in Section 5, where the grid production is considered an input feature, while the true
production is the figure that we aim at estimating (hence its name).

285 4.1. Feature definition

In order for our ML models to capture the wind park problem, it is very important
to describe its characteristics in a meaningful way. In particular, we need to give to
the ML models valuable information on the turbine type used, on the wind and on the
site. In order to asses which are the most useful information to include we used our
290 knowledge of the problem and different visualizations of the data. As a result of our
analysis, the following features have been selected:

- the rated power of the turbine: this is the maximum power (MW) that the turbine can produce (at high wind speeds); it describes the turbine model, and impacts both park production and interference;
- 295 • the rotor diameter of the turbine: this describes the dimensions of the turbine; it impacts the production, the interference and the minimum distance between turbines;
- the square root of the area of the site (expressed in rotor diameters);
- the ratio between the two edges of the rectangle: this captures the shape of the site;
- 300 • the production of a regular layout on a grid: this is the output of case 1) in Section 4, and captures both the wind in the site and the site dimensions (as the bigger the site, the lower the interference).

Figure 7 plots the relation between our selected features and the optimized production we aim at estimating. It can be noticed, in particular, that the production increases 305 having bigger sites, since the optimizer can better spread the turbines (Figure 7a). For an easier reading of our plot we here indicate the square root of the area in m (while we express it in rotor diameters when we give it on input to the ML models). Analysing our data we can notice that production also depends on the shape of the site: squared sites have slightly higher production (Figure 7b). Since in our problem we suppose to build a fixed number of turbines (50) at, at least, 5 rotor diameters of distance, the optimized production highly depends also on the selected turbine model: the more powerful (bigger) the turbine, the higher the production (Figures 7c and 7d). Finally, we can see in Figure 7e that the production of a regular layout is highly correlated to 310 the optimized production: if we have a big site with good wind and powerful turbines, both the the grid production and the optimized one will be high. Nevertheless, the relation is not linear as there are some sites that are underestimated by the production on a grid: in Figure 7e, one can notice that multiple sites are evaluated as equivalent by the production on a grid, but not by the optimizer. This is the case, for example,

320 for sites with grid production 185 (vertical dotted line in Figure 7e) that have different optimized productions.

In order for our ML models to work, we need to encode our features in a way that is easy for the model to interpret. This is why, for example, we preferred to use the ratio between edges instead of their length. In the same way, we do not explicitly pass
325 the wind statistics of the site to the ML model, but we prefer to use the regular-grid production (that contains a much richer information, since it relates already the wind with the site dimensions and the turbines used).

5. Machine Learning

We defined three different ML models to estimate the reduction in power produc-
330 tion due to the interference, namely Linear Regression, Neural Networks (NNs) and Support Vector Regression (SVR). In order to benchmark our solutions we used the production on a regular grid layout as a baseline. This is, indeed, the approach that practitioners would use to quickly estimate the potential of a site.

The features described in the previous session are provided on input to the ML models through a vector \mathbf{x} . The estimated optimized power production \hat{y} is then modelled through a function f that depends on some unknown parameters \mathbf{w} learned during the training phase, i.e.

$$\hat{y} = f(\mathbf{w}, \mathbf{x}) .$$

We used the *Root Mean Squared Error (RMSE)* [25] to measure the quality of our ML models. RMSE is a widely-used measure in regression models, as it gives a good description of the deviation between the predictions of the model and the “true” values. Given a training data set containing input-output pairs $\{(\mathbf{x}_n, y_n), n = 1, \dots, N_{train}\}$, the RMSE formula reads:

$$E(\mathbf{w}) = \sqrt{\frac{1}{N_{train}} \sum_{n=1}^{N_{train}} (y_n - f(\mathbf{w}, \mathbf{x}_n))^2}$$

where $E(\mathbf{w})$ is minimized when our estimate $f(\mathbf{w}, \mathbf{x}_n)$ is as close as possible to the
335 “real value” y_n . The optimal parameters \mathbf{w}^* for our ML models are therefore found

as $\mathbf{w}^* = \operatorname{arg\,min}_{\mathbf{w}} E(\mathbf{w})$. For Linear Regression this minimization problem is solved easily, as it is a convex problem with an analytic solution. The drawback of linear regression, however, is that it can only model linear dependencies in the data. We therefore defined more powerful models by introducing some non-linearities, in the form of stacked layers of non-linear functions for the *NN* model, or kernels for the SVR. A detailed description of these models and their optimization procedures is out of the scope of this work. The interested reader is referred to [25] for an in-depth discussion on the topic.

6. Results

We divided our 3000+ instances in training and test set, constructed to reflect the practical application of this work: the company would use our ML model to evaluate the potential of new sites (not seen during training), where the main difference is in the wind statistics. The training set consist of 2268 instances, corresponding to the winds statistics of the real sites Borssele 1 and 2, Borssele 3 and 4, Horns Rev 3, and Danish Krigers Flak. These instances are used to train and calibrate our ML models. The remaining 1134 instances, that correspond to the wind statistics of Ormonde and Hollandse Kust Zuid, are used as test set to evaluate the performances of the different ML models, and will give us a measure of how much our models *generalize* to previously-unseen data.

As these models are very sensitive to different scaling of the input features, we standardized all the features to have mean 0 and standard deviation 1 over the training set. The hyperparameters of the models (e.g., the number of layers and units of the NN or the kernel type in the SVR) are chosen using the *scikit-learn* [26] function *GridSearchCV*, that exhaustively considers all parameter combinations on a grid (5-fold cross-validated on the training set). According to our tests, the best NN is a single-layer network with 20 hidden units and hyperbolic tangent non-linearity. For the SVR, the best performing kernel function is the Radial Basis Function (Gaussian) kernel. All the models were again implemented using Python's machine learning library *scikit-learn*.

Model	RMSE
Grid	4.15
Linear Regression	1.15
Neural Network	2.24
Support Vector Regression	0.74

Table 1: RMSE of the test set for the different models.

365 In Table 1 we compare the performance of the models in terms of RMSE on the
test set. We see that all models outperform by a large margin the baseline given by the
production on a regular grid (denoted as *Grid* in the table), with Support Vector Re-
gression being the best among them. Importantly, all the ML models can be evaluated
on new instances in less than a second, as opposed to the 1-hour runs needed when
370 using the MO algorithm.

In Figure 8 we visualize the predictions of the models on the test set: on the y -axis
we report the true optimized production, and on the x -axis its estimate from the model
(each point in the graph represents a different test instance). If the predictions were
perfect, all the points would lay on the $y = x$ line (in red in the plot).

375 An alternative view of the predictions from the model can be found in Figure 9
that shows, for some of the instances, the *predicted* optimal production from the dif-
ferent ML models, the one obtained with the *grid* solution (what a practitioner would
estimate) and the optimized (*true*) production. From the different plots it is clear the
importance of both MO and ML in this problem. The optimization of the layout is
380 extremely valuable, since it can increase the production of the park (as the difference
between the green and the orange line shows). In order to fast and accurately evaluate
the potential of a new site, ML models can provide accurate prediction. In particu-
lar SVR models highly outperform the manual operator strategy (grid) and are able to
predict very closely the optimized production value.

385 As we shortly discussed already by commenting Figure 7e, it is very important to
use MO+ML optimization to evaluate a site rather than only grid production, as the
grid production may result in a wrong ranking of the sites. In Figure 10 we show

the difference between the true production and the grid production, and the difference between true and the predicted production (for SVR). From the plot one can notice that
390 the difference between the real and the predicted production is always very low (around 0), while the grid tends to underestimate the sites in almost all cases. The difference between the true and the grid production varies a lot from instance to instance: this makes it impossible to simply improve the grid estimate (for example, using a scaling factor), and proves the value of more complex ML techniques.

395 **7. Conclusions and future work**

In the present work we showed the relevance of using MO and ML techniques together. We have shown that ML techniques (SVRs in particular), trained on a large number of (almost) optimal solutions, can predict the optimal value of new instances of the same problem rather well. In this work, we have focused on the wind park
400 layout problem. The exact value of optimized solutions for this specific problem is very difficult to compute, given the large number of constraints and non-linearities involved, and can only be obtained using a sound MO tool. In our tests we have shown that the ML estimate highly outperforms the human one.

A possible extension of the model can be to allow for different numbers of turbines
405 or different shapes of the park site (not only rectangles). More ambitiously, future work could investigate the application of our approach to different OR problems. One could, indeed, address the problem of estimating the optimal *value* of an optimization problem by using ML algorithms trained on optimized solutions computed off-line by time-consuming MO solvers. This estimate can be of interest by itself (as in the wind
410 farm application studied in this paper), but can also be very useful, e.g., for heuristic node pruning in a branch-and-bound solution scheme.

References

- [1] D. Bertsimas, N. Kallus, From predictive to prescriptive analytics, eprint arXiv:1402.5481.

- 415 [2] A. Lodi, G. Zarpellon, On learning and branching : a survey Technical Report
(2017) 1–29.
- [3] A. M. Alvarez, Q. Louveaux, L. Wehenkel, A Supervised Machine Learning
Approach to Variable Branching in Branch-And-Bound, *INFORMS Journal on
Computing* 29 (1) (2017) 185–195.
- 420 [4] E. B. Khalil, P. L. Bodic, L. Song, G. Nemhauser, B. Dilkina, Learning to Branch
in Mixed Integer Programming, in: *Proceedings of the Thirtieth AAAI Confer-
ence on Artificial Intelligence (AAAI-16) Learning*, 2016.
- [5] M. Kruber, E. L. Marco, Learning when to use a decomposition, in: *Proceeding
of CPAIOR2017*, 2017.
- 425 [6] I. Bello, H. Pham, Q. V. Le, M. Norouzi, S. Bengio, Neural combinatorial opti-
mization with reinforcement learning, *CoRR abs/1611.09940*. *arXiv:1611.*
09940.
URL <http://arxiv.org/abs/1611.09940>
- [7] H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, L. Song, Learning combinatorial
430 optimization algorithms over graphs, *CoRR abs/1704.01665*. *arXiv:1704.*
01665.
URL <http://arxiv.org/abs/1704.01665>
- [8] A. Hottung, S. Tanaka, K. Tierney, Deep learning assisted heuristic tree search for
the container pre-marshalling problem, *CoRR abs/1709.09972*. *arXiv:1709.*
435 09972.
URL <http://arxiv.org/abs/1709.09972>
- [9] E. B. Khalil, B. Dilkina, G. L. Nemhauser, S. Ahmed, Y. Shao, Learning to run
heuristics in tree search, in: *Proceedings of the Twenty-Sixth International Joint
Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 659–666. *doi:10.*
440 24963/ijcai.2017/92.
URL <https://doi.org/10.24963/ijcai.2017/92>

- [10] Combining constraint solving with mining and learning, *Artificial Intelligence* 244 (2017) IFC –. doi:[https://doi.org/10.1016/S0004-3702\(17\)30008-5](https://doi.org/10.1016/S0004-3702(17)30008-5).
445 URL <https://www.sciencedirect.com/science/article/pii/S0004370217300085>
- [11] A. Passerini, G. Tack, T. Guns, Introduction to the special issue on combining constraint solving with mining and learning, *Artificial Intelligence* 244 (2017) 1 – 5, combining Constraint Solving with Mining and Learning. doi:<https://doi.org/10.1016/j.artint.2017.01.002>.
450 URL <https://www.sciencedirect.com/science/article/pii/S0004370217300036>
- [12] J. Berg, M. Jrvisalo, Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability, *Artificial Intelligence* 244 (2017) 110 – 142, combining Constraint Solving with Mining and Learning. doi:<https://doi.org/10.1016/j.artint.2015.07.001>.
455 URL <https://www.sciencedirect.com/science/article/pii/S0004370215001022>
- [13] M. Bartlett, J. Cussens, Integer linear programming for the bayesian network structure learning problem, *Artificial Intelligence* 244 (2017) 258 – 271, combining Constraint Solving with Mining and Learning. doi:<https://doi.org/10.1016/j.artint.2015.03.003>.
460 URL <https://www.sciencedirect.com/science/article/pii/S0004370215000417>
- [14] C. Bessiere, F. Koriche, N. Lazaar, B. O’Sullivan, Constraint acquisition, *Artificial Intelligence* 244 (2017) 315 – 342, combining Constraint Solving with Mining and Learning. doi:<https://doi.org/10.1016/j.artint.2015.08.001>.
465 URL <https://www.sciencedirect.com/science/article/pii/S0004370215001162>
470

- [15] M. Fischetti, M. Monaci, Proximity search heuristics for wind farm optimal layout, *Journal of Heuristics* 22 (4) (2016) 459–474.
- [16] Optimal siting and sizing of wind farms, *Renewable Energy* 101 (2017) 51 – 58.
- [17] A method for optimizing the location of wind farms, *Renewable Energy* 48 (2012) 287 – 299.
- 475
- [18] Optimization of wind turbine layout position in a wind farm using a newly-developed two-dimensional wake model, *Applied Energy* 174 (2016) 192 – 200.
- [19] M. Fischetti, D. Pisinger, Mathematical optimization for offshore wind farm design: an overview, *Business and Information Systems Engineering*.
- 480
- [20] R.J.Barthelmie, K.Hansen, S. T. Frandsen, O. Rathmann, J. G. Schepers, W. Schlez, J. Phillips, K. Rados, A. Zervos, E. S. Politis, P. K. Chaviaropoulos, Modelling and measuring flow and wind turbine wakes in large wind farms offshore, *Wind Energy* 12 (2009) 431444.
- [21] Vattenfall, internal image storage (2017).
- 485
- [22] M. Fischetti, Mixed-integer models and algorithms for wind farm layout optimization, Master’s thesis, University of Padova, <http://tesi.cab.unipd.it/45458/1/tesi.Fischetti.pdf> (2014).
- [23] N. Jensen, A note on wind generator interaction, Tech. rep., Technical Report Ris-M- 2411(EN), Ris National Laboratory, Roskilde, Denmark (1983).
- 490
- [24] M. Fischetti, M. Monaci, Proximity search for 0-1 mixed-integer convex programming, *Journal of Heuristics* 20 (6) (2014) 709–731.
- [25] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- 495

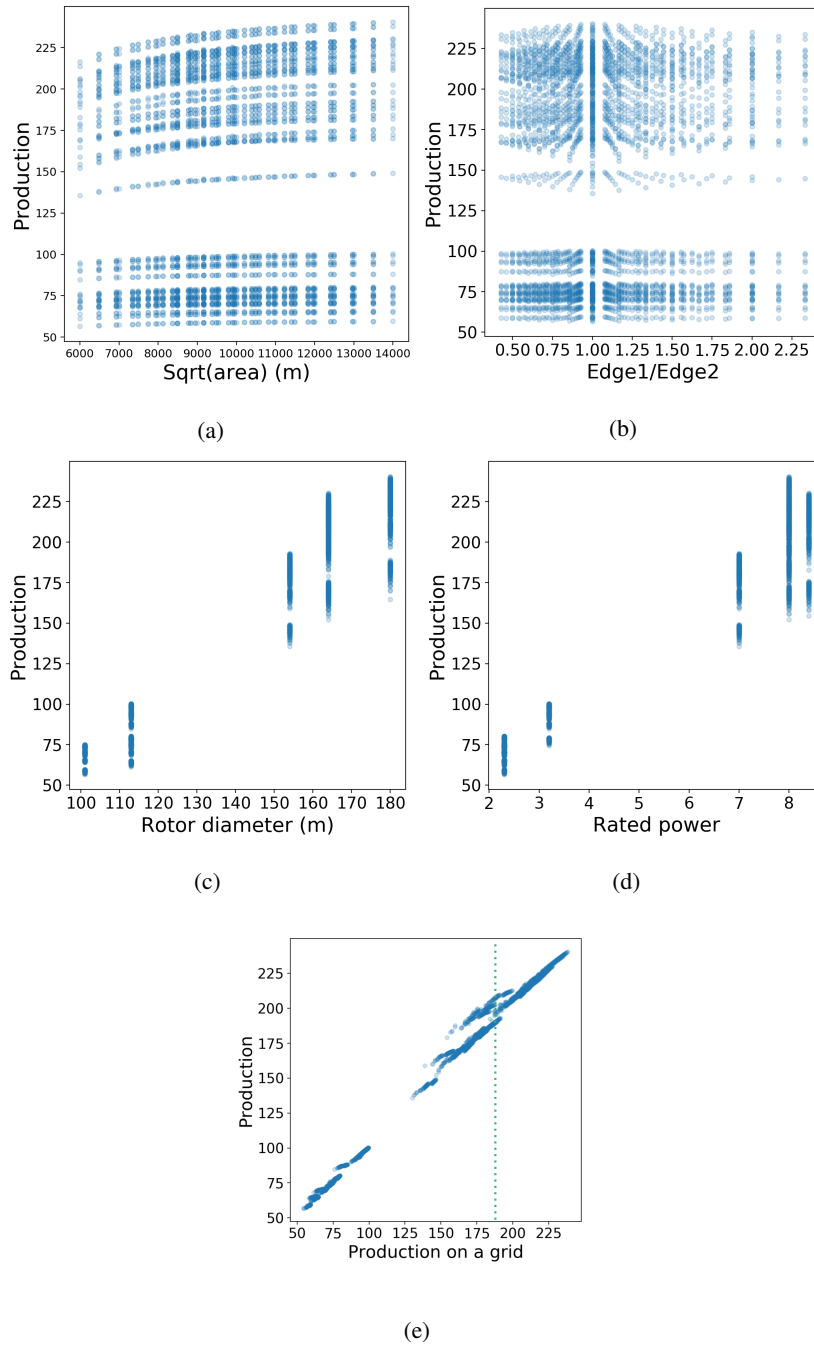
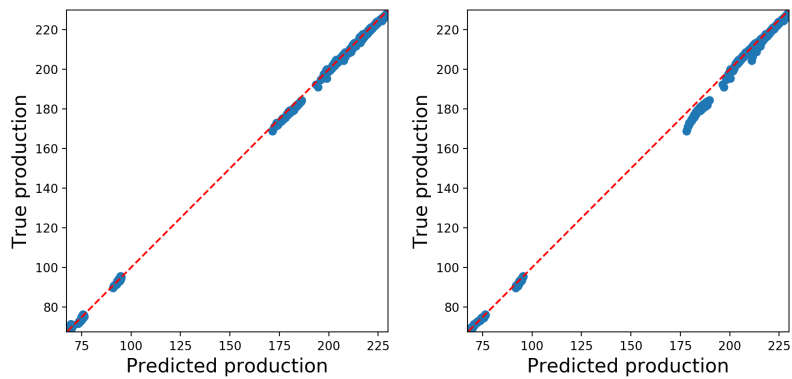
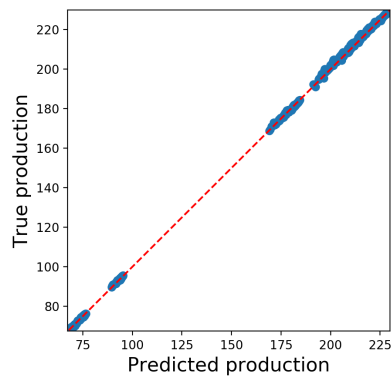


Figure 7: Relation between optimal power production and our input features.



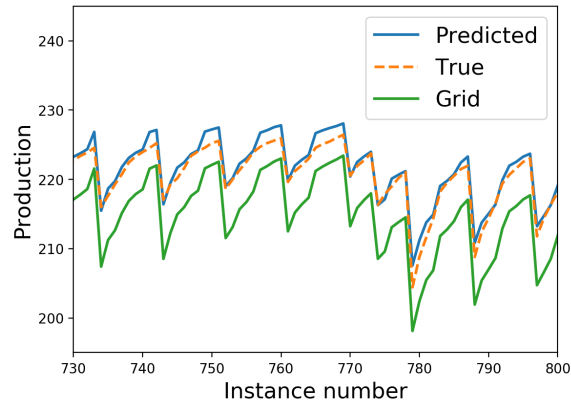
(a) Linear Regression

(b) Neural Network

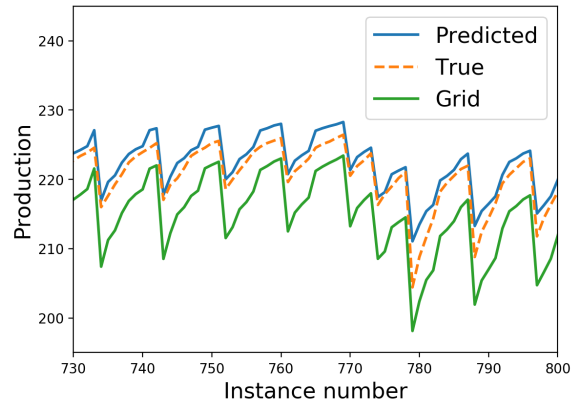


(c) Support Vector Regression

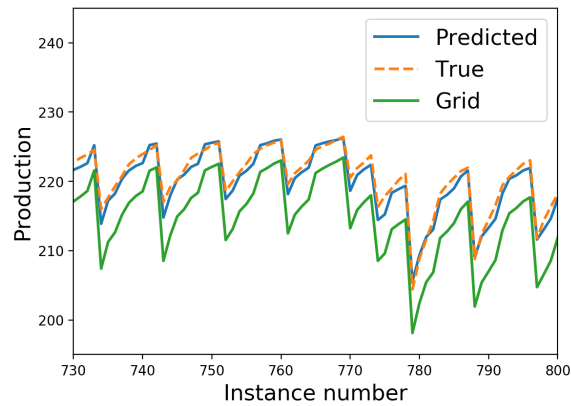
Figure 8: Comparison between the predicted optimal productions (x axis) and the true optimal productions (y axis) of the test set. The ideal predictions are shown with the dashed red line.



(a) Linear Regression



(b) Neural Network



(c) Support Vector Regression

Figure 9: Power production (y axis) for different instances (x axis). In green the power production of a layout on a regular grid, in orange the power production (for the same site and turbines) obtained with an optimized layout. In blue the prediction from the different ML models (the closer to the orange line, the better).

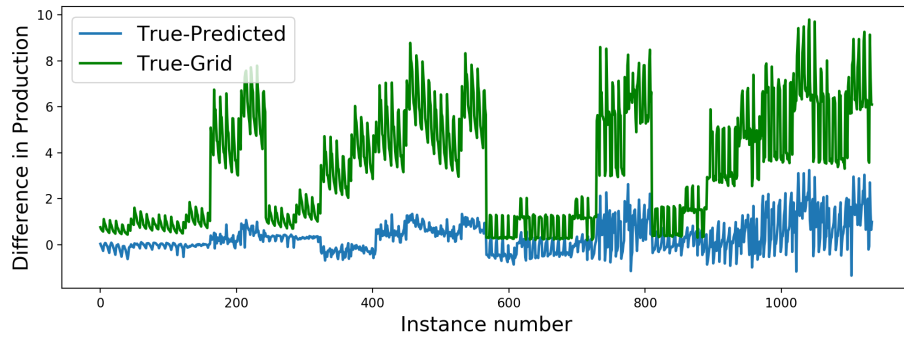


Figure 10: Difference between true and grid production (in green) and true and predicted production for our SVR model (in blue). The comparison shows the importance of using ML instead of a simple grid evaluation, for a correct ranking of possible new sites.