Renewable Energy 50 (2013) 236–243

Contents lists available at SciVerse ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

Predicting the energy output of wind farms based on weather data: Important variables and their correlation

Ekaterina Vladislavleva [a], Tobias Friedrich [b], Frank Neumann [c,*], Markus Wagner [c]

[a] Evolved Analytics Europe BVBA, Veldstraat 37, 2110 Wijnegem, Belgium and Faculty of Computer Science and Engineering Science, Cologne University of Applied Sciences, Steinmüllerallee 1, 51643 Gummersbach, Germany
[b] Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany
[c] School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia

ARTICLE INFO

Article history:
Received 26 October 2011
Accepted 18 June 2012
Available online 24 July 2012

Keywords:
Wind energy
Prediction
Genetic programming
DataModeler

ABSTRACT

Wind energy plays an increasing role in the supply of energy world wide. The energy output of a wind farm is highly dependent on the weather conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction. In this paper, we take a computer science perspective on energy prediction based on weather data and analyze the important parameters as well as their correlation on the energy output. To deal with the interaction of the different parameters, we use symbolic regression based on the genetic programming tool DataModeler. Our studies are carried out on publicly available weather and energy data for a wind farm in Australia. We report on the correlation of the different variables for the energy output. The model obtained for energy prediction gives a very reliable prediction of the energy output for newly supplied weather data.

1. Introduction

Renewable energy, such as wind and solar energy, plays an increasing role in the supply of energy world wide. This trend will continue because global energy demand is increasing, and the use of nuclear power and traditional sources of energy such as coal and oil is either considered unsafe or leads to a large amount of CO_2 emission.

Wind energy is a key player in the field of renewable energy. The capacity of wind energy production has been substantially increased during the last years. In Europe, for example, the capacity of wind energy production has doubled from 2005 to 2007 [13]. However, levels of production of wind energy are hard to predict as they rely on potentially unstable weather conditions present at the wind farm. In particular, wind speed is crucial for energy production based on wind, and it may vary drastically over time. Energy suppliers are interested in accurate predictions, as they can avoid overproduction by coordinating the collaborative production of traditional power plants and weather-dependent energy sources.

Our aim is to map weather data to energy production. We wish to show that even data that is publicly available for weather stations close to wind farms can be used to give a good prediction of the energy output. Furthermore, we examine the impact of different weather conditions on the energy output of wind farms. We are particularly, interested in the correlation of different components that characterize weather conditions such as wind speed, pressure, and temperature.

A good overview on the different methods that were recently applied in forecasting of wind power generation can be found in [3]. Statistical approaches use historical data to predict the wind speed on an hourly basis or to predict energy output directly. On the other hand, short term prediction is often done based on meteorological data, and learning approaches are applied. Kusiak, Zheng, and Song [9] have shown how wind speed data may be used to predict the power output of a wind farm based on time-series prediction modeling. Neural networks are a very popular learning approach for wind power forecasting based on given time series. They provide an implicit model of the function that maps the given weather data to an energy output.

Jursa and Rohrig [4] have used particle swarm optimization and differential evolution to minimize the prediction error of neural networks for short-term wind power forecasting. Kramer and Gieseke [8] used support vector regression for short-term energy forecast and kernel methods and neural networks to analyze wind energy time series [7]. These studies are all based on wind data and do not take other weather conditions into account. Furthermore, neural networks have the disadvantage that they give an implicit model of the function predicting the output, and these models are rarely accessible to a human expert. Usually, one is also interested in

* Corresponding author.
E-mail address: frank.neumann@adelaide.edu.au (F. Neumann).

0960-1481/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.renene.2012.06.036

the function itself and the impact of the different variables that determine the output. We aim to study the impact of different variables on the energy output of the wind farm. Surely, the wind speed available at the wind farm is a crucial parameter [1,12]. Other factors that influence the energy output are, for example, air pressure, temperature and humidity. Our goal is to study the impact and correlation of these parameters with respect to the energy output.

Genetic programming (GP) (see [10] for a detailed presentation) is a type of evolutionary algorithm that can be used to search for functions that map input data to output data. It has been widely used in the field of symbolic regression and the goal of this paper is to show how it can be used for the important real-world problem of predicting energy outputs of wind farms from weather data. The advantage of this method is that it comes up with an *explicit* expression mapping weather data to energy output. This expression can be further analyzed to study the impact of the different variables that determine the output. To compute such an expression, we use the tool DataModeler [2], which is the state of the art tool for doing symbolic regression based on genetic programming. We will also use DataModeler to carry out a sensitivity analysis which studies the correlation between the different variables and their impact on the accuracy of the prediction.

We proceed as follows. In Section 2, we give a basic introduction into the field of genetic programming and symbolic regression, and describe the DataModeler. Section 3 describes our approach of predicting energy output based on weather data and in Section 4 we report on our experimental results. Finally, we finish with some concluding remarks and topics for future research.

2. Genetic programming and DataModeler

Genetic programming [6] is a type of evolutionary algorithm that is used in the field of machine learning. Motivated by the evolution process observed in nature, computer programs are evolved to solve a given task. Such programs are usually encoded as syntax expression trees. Starting with a given set of trees called the population, new trees called the offspring population are created by applying variation operators such as crossover and mutation. Finally, a new parent population is selected from among the previous parents and the offspring based on how well these trees perform for the given task.

Genetic programming has its main success stories in the field of symbolic regression. Given a set of input output vectors, the task is to find a function that maps the input to the output as best as possible, while avoiding over fitting. The resulting function is later often used to predict the output for a newly given input. Syntax trees represent functions in this case, and the functions are changed by crossover and mutation to produce new functions. The quality of a syntax tree is determined by how well it maps the given set of inputs to their corresponding outputs.

The task in symbolic regression can be stated as follows. Given a set of data vectors $(x_{1i}, x_{2i}, \cdots, x_{ki}, y_i) \in \mathbb{R}^{k+1}$, $1 \leq i \leq n$, find a function $f : \mathbb{R}^k \to \mathbb{R}$ such that the approximation error, e.g. the root mean square error

$$\sqrt{\frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{n}}$$

with $x_i = (x_{1i}, x_{2i}, \cdots, x_{ki})$, is minimized.

We chose to use a tool called DataModeler for our investigations. It is based on genetic programming and designed for solving symbolic regression problems.

2.1. DataModeler

Evolved Analytics' DataModeler is a complete data analysis and feature selection environment running under Wolfram Mathematica 8. It offers a platform for data exploration, data-driven model building, model analysis and management, response exploration and variable sensitivity analysis, model-based outlier detection, data balancing and weighting.

Data-driven modeling in DataModeler is done by symbolic regression via genetic programming. The Symbolic Regression function offers several evolutionary strategies which differ in the applied selection schemes, elitism, reproduction strategies, and fitness evaluation strategies. An advanced user can take full control over symbolic regression and introduce new function primitives, new fitness functions, selection and propagation schemes, etc., by specifying appropriate options in the function call. However, we used the default settings and the default evolution strategy, which in DataModeler is called ClassicGP.[1]

In the symbolic regression performed here, a population of individuals (syntax trees) evolves over a variable number of generations at the Pareto front in the three dimensional objective space of model complexity, model error, and model age [5,11].

Model error in the default setting ranges between 0 and 1, with the best value being 0. It is computed as $1-R^2$, where R is a scaled correlation coefficient. The correlation coefficient of the predicted output is scaled to have the same mean and standard deviation as the observed output.

The model complexity is the expressional complexity of models, and it is computed as the total sum of nodes in all subtrees of the given GP tree. The model age is computed as the number of generations that the model survived in the population. The age of a child individual is computed by incrementing the age of the parent contributing to the root node of the child. We use the age as a secondary optimization objective, as it is used only internally for evolution. At the end of symbolic regression runs, results are displayed in the two-objective space of user-selected objectives, in our case, these objectives are model expressional complexity and $1-R^2$.

The population-specific parameters of our genetic programming system are chosen as follows. The default population size is 300. The default elite set size is 50 individuals from the 'old' population closest to the 3-dimensional Pareto front in the objective space. These individuals are copied to the 'new' population of 300 individuals, after which the size of the new population is decreased down to the necessary 300. This is done by selecting models from the Pareto layers until the initially specified amount of models is found.

The selection of individuals for propagation is done by means of Pareto tournaments. By default, 30 models are randomly sampled from the current population, and Pareto-optimal individuals from this sample are determined as winners to undergo variation until a necessary number of new individuals are created.

Models are coded as parse trees using the GPmodel structure, which contains placeholders for information about model quality, data variables and ranges used to develop the model, and some settings of symbolic regression. For example, the internal GPmodel representation of the first Pareto front model from a set of models from Fig. 3 with an expression $-25.2334 + 3.21666\ \text{windGust}_2$ is presented in Table 1. Note that the first vector inside the GPmodel structure represents model quality. Model complexity is 11, model error is 0.300409. The parse tree of the same model is plotted in Fig. 1.

When a specified execution threshold of a run in seconds is reached, the independent evolution run terminates and a vector of model objectives in the final population is re-evaluated to contain only model complexity and model error. The set of models can

[1] All models reported in this paper were generated using two calls of Symbolic Regression with only the following arguments: input matrix, response vector, execution time, number of independent evolutions, an option to archive models with a certain prefix-name, and a template specification.

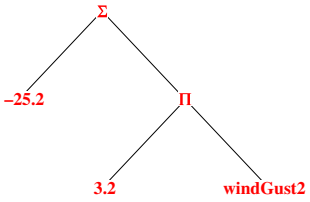


Fig. 1. Model tree plot of the individual from Table 1. Model complexity is the sum of nodes in all subtrees of the given tree (11). Model error is computed as $1-R^2 = 0.30$.

further be analyzed for variable drivers, most frequent variable combinations, behavior of the response, consistency in prediction, accuracy vs. complexity trade-offs, etc.

When the goal is the prediction of the output in the unobserved region of the data space, it is essential to use 'model ensemble' rather than individual models for this purpose. Because of the built-in niching, complexity control, and independent evolutions used in DataModeler's symbolic regression, the final models are developed to be diverse (with respect to structural complexity, model forms, and residuals), but they all are global models, built to predict training response in the entire training region. Due to diversity and high quality, rich sets of final models allow us to select multiple individuals to model ensembles. Prediction of a set of individuals is then computed as a median or a median average of individual predictions of ensemble members, while disagreement in the predictions (standard deviation in this paper) is used to specify the confidence interval of prediction. When models are extrapolated, the confidence of predictions naturally deteriorates and confidence intervals become wider. This allows first, a more robust prediction of the response (since overfitting is further mitigated by choosing models of different accuracy and complexity into an ensemble), and second, more trustworthy predictions, since they are also supplied with confidence intervals.

To select ensembles we used a built-in function in DataModeler that focuses on the most typical individuals of the model set as well as on individuals that have the fewest correlated residuals. Because of space constraints, we refer the reader to [2] for further information.

3. Our approach

The main goal of this paper is to use public data to check the feasibility of wind energy prediction by using an industrial-strength off-the-shelf non-linear modeling and feature selection tool. In our study, we investigate and predict the energy production of the wind farm *Woolnorth* in Tasmania, Australia based on publicly available data. The energy production data is made publicly available by the Australian Energy Market Operator (AEMO) in real time to assist in maintaining the security of the power system.[2] For the creation of our models and the prediction, we associate the wind farm with the Australian weather station *ID091245*, located at Cape Grim, Tasmania. Its data is available for free for a running observation time window of 72 h.[3]

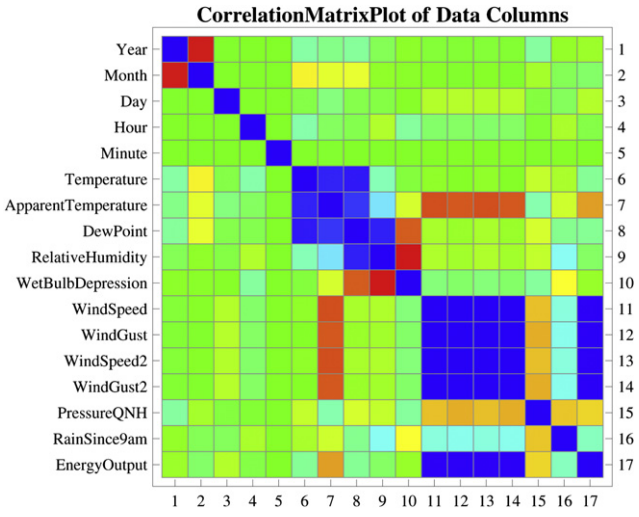


Fig. 2. Data variables are heavily correlated (Blue: positively, Red: negatively). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

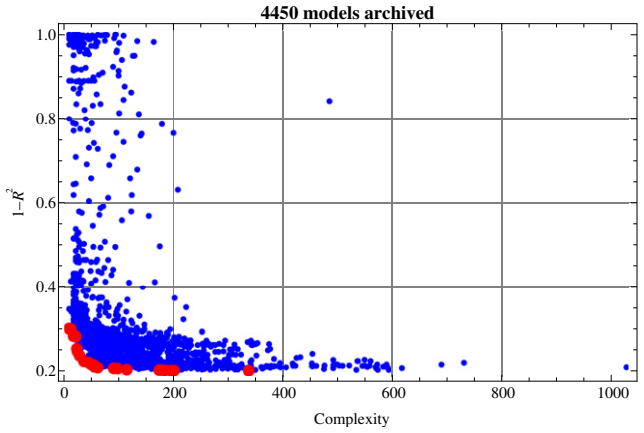


Fig. 3. A super set of models generated in the first stage of experiments with 10 independent evolutions using all inputs. Red dots are Pareto front models, which are non-dominated trade-offs in the space of model complexity and model error. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.1. Data

We collected both the weather and energy production data for the time window from September 2010 to July 2011. The output of the farm is available with a rate of one measurement every 5 min, and the weather data with a rate of one measurement every 30 min.

The wind farm's production capacity is split into two sites, which complicated the generation of models. The site "Studland Bay" has a maximum output of 75 MW, and "Bluff Point" has a maximum output of 65 MW and is located 50 km south of the first site. The weather station is located on the first site. For wind coming from the west (which is the prevailing wind direction), the difference in location is negligible. But if the wind comes from the north, there is an immediate energy and wind increase, plus another energy increase 1–2 h later (the time delay depends on the actual wind speed). Similarly, if the wind comes from the south, there will be an increase in the energy production (although no wind is indicated by the weather station) and then, 1–2 h later, an energy increase accompanied by a measured wind speed increase.

3.2. Data pre-processing

To perform data modeling and variable selection on collected data, we had to perform data pre-processing to create a table of weather and energy measurements taken at the same time

[2] Australian Landscape Guardians: AEMO Non-Scheduled Generation Data: www.landscapeguardians.org.au/data/aemo/(last visited August 31st, 2011).

[3] Australian Government, Bureau of Meteorology: weather observations for Cape Grim: www.bom.gov.au/products/IDT60801/IDT60801.94954.shtml (last visited August 31st, 2011).

Table 1
Internal regression model representation in DataModeler for the model with an expression $-25.2334 + 3.21666 \cdot \text{windGust}_2$ (see also Fig. 1).

$$
\begin{aligned}
&\textbf{GPModel}\ [\\
&\quad \{11, 0.300409\},\\
&\quad \Sigma[-25.2334, \Pi[3.21666, \text{windGust2}]],\\
&\quad \{\ \textbf{ModelAge} \to 1,\\
&\qquad \textbf{ModelingObjective} \to (\{\text{ModelComplexity}[\#1],\\
&\qquad\qquad 1 - \text{AbsoluteCorrelation}[\#2, \#3]^2\}\ \&)\ ,\\
&\qquad \textbf{ModelingObjectiveNames} \to \{\text{Complexity}, 1\text{-}R^2\},\\
&\qquad \textbf{DataVariables} \to \{\text{year}, \text{month}, \text{day}, \text{hour}, \text{minute}, \text{temperature},\\
&\qquad\qquad \text{apparentTemperature}, \text{dewPoint}, \text{relativeHumidity},\\
&\qquad\qquad \text{wetBulbDepression}, \text{windSpeed}, \text{windGust}, \text{windSpeed2},\\
&\qquad\qquad \text{windGust2}, \text{pressureQNH}, \text{rainSince9am}\},\\
&\qquad \textbf{DataVariableRange} \to \{\{2010, 2011\}, \{1, 12\}, \{1, 31\}, \{0, 23\}, \{0, 30\},\\
&\qquad\qquad \{4.2, 23.4\}, \{-14.2, 24.\}, \{-3.2, 19.1\}, \{40, 100\}, \{0., 6.6\}, \{0, 106\},\\
&\qquad\qquad \{0, 130\}, \{0, 57\}, \{0, 70\}, \{987.8, 1037.5\}, \{0., 50.4\}\},\\
&\qquad \textbf{RangeExpansion} \to \text{None},\\
&\qquad \textbf{ModelingVariables} \to \{\text{year}, \text{month}, \text{day}, \text{hour}, \text{minute}, \text{temperature},\\
&\qquad\qquad \text{apparentTemperature}, \text{dewPoint}, \text{relativeHumidity},\\
&\qquad\qquad \text{wetBulbDepression}, \text{windSpeed}, \text{windGust}, \text{windSpeed2},\\
&\qquad\qquad \text{windGust2}, \text{pressureQNH}, \text{rainSince9am}\},\\
&\qquad \textbf{FunctionPatterns} \to \{\Sigma[\_, \_\_], \Pi[\_, \_\_], \mathbb{D}[\_, \_], \mathbb{S}[\_, \_], \mathbb{P}2[\_], \mathbb{SQ}[\_], \mathbb{IV}[\_], \mathbb{M}[\_]\},\\
&\qquad \textbf{StoreModelSet} \to \text{True},\\
&\qquad \textbf{ProjectName} \to \text{fullDataAllVars},\\
&\qquad \textbf{TemplateTopLevel} \to \{\Sigma[\_, \_\_]\},\\
&\qquad \textbf{TimeConstraint} \to 2000,\\
&\qquad \textbf{IndependentEvolutions} \to 10\\
&\quad \}\\
&].
\end{aligned}
$$

intervals. Energy output of the farm is measured at the rate of 5 min, including the time stamps of 0 and 30 min of every hour when the weather is measured. Our approach was to correlate weather measurements with the average energy output of the farm reported in the [0, 25] and [30, 35] min intervals of every hour. Such averaging makes modeling more difficult, but uses all available energy information.

Different time scales used in the weather and energy data were automatically converted to one scale using a DateList function in Wolfram Mathematica 8, which is the scientific computing environment in which DataModeler operates.

Because of many missing, erroneous, and duplicate time stamps in the weather data, we obtained 11,022 common measurements of weather and averaged energy produced by the farm from October 2010 to June 2011. These samples were used as training data to build regression models. From 18 variables of the weather data at Cape Grim, we excluded two variables prior to modeling: more than 75% of values for the *Pressure MSL* variable were missing and the *Wind Direction* variable was non-numeric.

As test data we used 1408 common half-hour measurements of weather and averaged energy in July 2011.

3.3. Data analysis and model development

As soon as weather and energy data from different sources were put in an appropriate input–output form, we were able to apply a standard data-driven modeling approach to them.

A good approach employs iterations among three stages: Data Collection/Reduction, Model Development, and Model Analysis and Variable Selection. In hard problems, many iterations are required to identify a subspace of minimal dimensionality where models of appropriate accuracy and complexity trade-offs can be built.

Our problem is challenging for several reasons. First, it is hard to predict the total wind energy output of the farm in the half-hour following the moment when weather is measured, especially when the weather station is several kilometers away from the farm.

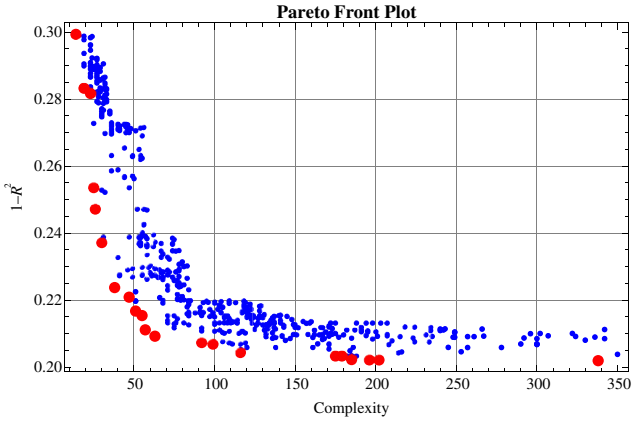


Fig. 4. Selected set $\mathscr{M}_1$ of 'best' models in all variables and two modeling objectives.

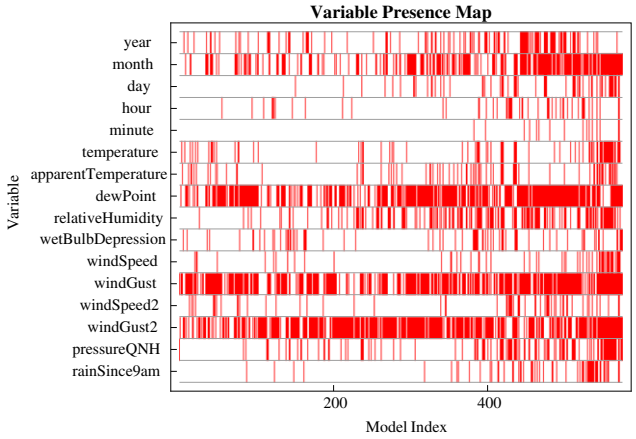


Fig. 5. Presence of input variables in the selected set $\mathscr{M}_1$.

Second, public data does not offer any information about the wind farm except for wind energy output. Third, our training data covers the range of weather conditions observed only between October 2010 and June 2011, while the test data contains data from July, implying that our models must have good generalization capabilities as they will be extrapolated to the unseen regions of the data space. And lastly, our most challenging goal is to use all *16* publicly available numeric weather characteristics for energy output prediction, although many of them are heavily correlated (see Fig. 2).

Multi-collinearity in hard high-dimensional problems is a major hurdle for most regression methods. Symbolic regression via GP is one of the very few methods that does not suffer from multi-collinearity and which is capable of naturally selecting variables from the correlated subset for final regression models.

Because ensemble-based symbolic regression and robust variable selection methodology are implemented in DataModeler, we settled on a standard model development and variable selection procedures using default settings.

The modeling goals of this study are:

(1) to identify the minimal subset of driving weather features that are significantly related to the wind energy output of the wind farm,
(2) to let genetic programming express these relationships in the form of explicit input–output regression models, and
(3) to select model ensembles for improved generalization capabilities of energy predictions and to analyze the quality of produced model ensembles using an unseen test set.

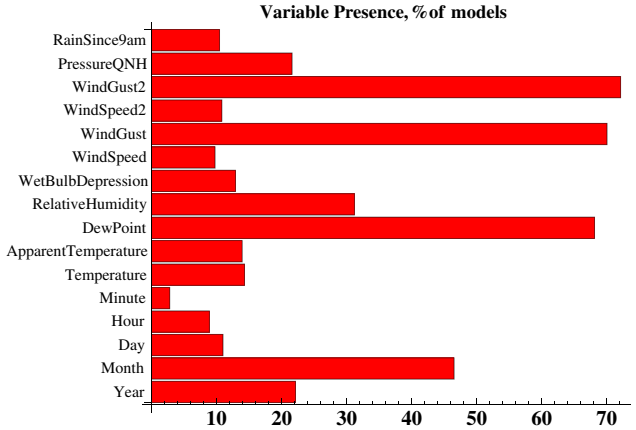


Fig. 6. Presence of input variables in the selected set of models.

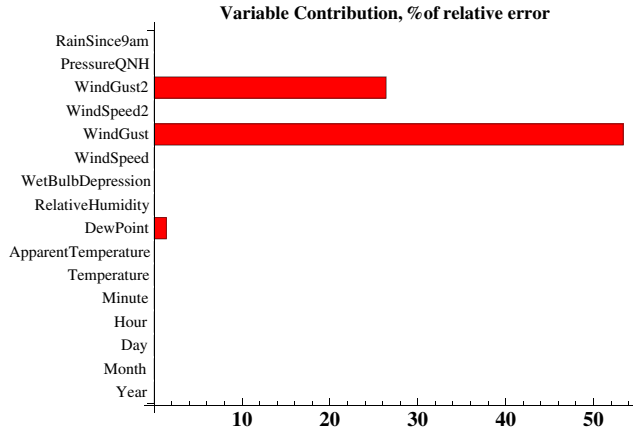


Fig. 7. Individual contributions of input variables in the selected set of models to the relative training error.

Our approach is to achieve these goals using two iterations of symbolic regression modeling. At the first exploratory stage, we run symbolic regression on training data to identify driving weather characteristics significantly related to the energy output. At the second modeling stage, we reduce the training data to the set of selected inputs and run symbolic regression to obtain models, and model ensembles for predicting energy output.

4. Experimental results

4.1. Experimental setup

The setup of symbolic regression used the default settings of DataModeler except for the number of independent runs, execution time of each run, and the template operator at the root of the GP trees. We executed 10 independent evolutionary runs of 2000 s in both stages. The root node of all GP trees was fixed to a Plus. The primitives for regression models consisted of an extended set of arithmetic operators:

$$\{\text{Plus}, \text{Minus}, \text{Subtract}, \text{Divide}, \text{Times}, \text{Sqrt}, \text{Square}, \text{Inverse}\}.$$

The maximum arity of Plus and Times operators is limited to 5.

Model trees have terminals labeled as variables or constants (random integers or reals), with a maximum allowed model complexity of 1000. Population size is 300; elite set size is 50. Population individuals are selected for reproduction using Pareto

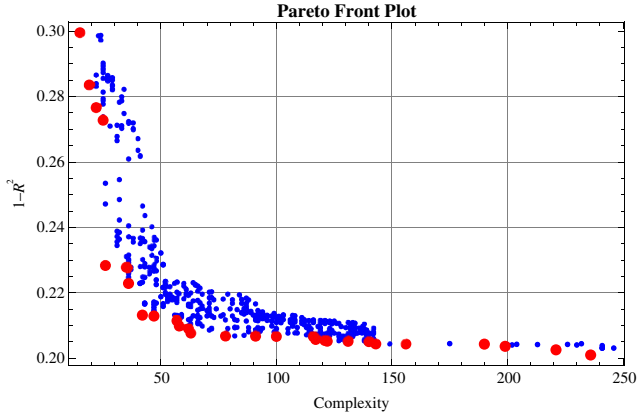


Fig. 8. Selected set $\mathscr{M}_2$ of 'best' models in up to two-dimensional input space and two modeling objectives.

Variable Combination Table

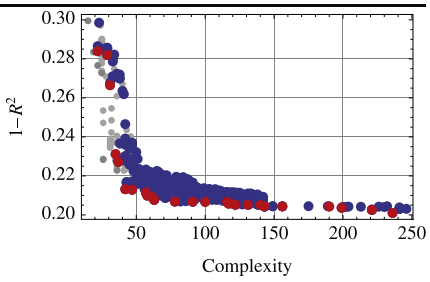
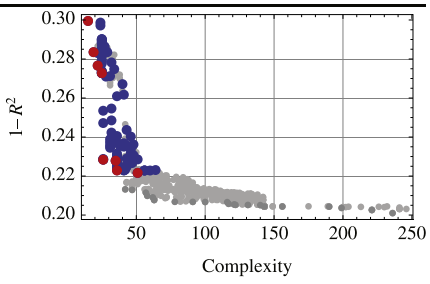
	% of Models	# of Models	Variables Used	ParetoFrontPlot
1	83.3	489	dewPoint windGust2	
2	16.7	98	windGust2	

Fig. 9. Visualization of models in $\mathscr{M}_2$ niched per driving variable combination. Note, that windGust_2 alone is insufficient to predict energy output with the accuracy that is achieved when windGust_2 and dewPoint are used. The model error is computed using training data.

tournaments with a tournament size of 30. Propagation operators are crossover (at rate 0.9), subtree mutation (rate 0.05), and depth-preserving subtree mutation (rate 0.05). At the end of each independent evolution, the population and archive individuals are merged to produce a final set of models. At each stage of experiments the results of all independent evolutions are merged together to produce a superset of solutions (see an example in Fig. 3).

For model analysis, we applied additional model selection strategies to these supersets of models. We describe the additional model selection strategies, discovered variable drivers, final models, and the quality of predictions in the next section.

4.2. Feature selection

The initial set of experiments targets the feature selection, using all 16 input variables and all training data from October 2010 to June 2011. In the allowed 2000 s, each symbolic regression run completed at most 217 generations.

The 10 independent evolutions generated a superset of 4450 models. We reduced this set to robust models only, by applying interval arithmetic to remove models with potential for pathologies and unbounded response in the training data range. This generated 2559 unique robust models, and from those we selected the final set $\mathscr{M}_1$. This set contained 587 individuals with the model error not exceeding 0.30, and model complexity not exceeding 350, that lie closest to the Pareto front in model complexity versus model error objective space. The set $\mathscr{M}_1$ is depicted in Fig. 4 with Pareto front individuals indicated in red. The limit of 350 on model complexity preserved the best model of the run (the rightmost red dot), but excluded dominated individuals with model complexities up to 600.

We used the set $\mathscr{M}_1$ to perform variable presence and variable contribution analysis to identify the variable drivers significantly related to energy output. The presence of input variables in models from $\mathscr{M}_1$ is visualized in Figs. 5 and 6. We can observe from Fig. 6 that the six most frequently used variables are (in order of decreasing importance) windGust_2, windGust, dewPoint, month, relativeHumidity, and pressureQNH. While we observe that these variables are most frequently used in a good set of candidate solutions in $\mathscr{M}_1$, it is somewhat hard to define a threshold on these presence-based variable importances to select variable drivers. For example, it is unclear whether we should select the top three, four, or five inputs.

For a crisper feature selection analysis, we performed a variable contribution analysis using DataModeler to see how much each variable contributes to the relative error of the model where it is present. The median variable contributions computed using the model set $\mathscr{M}_1$ are depicted in Fig. 7. The plot clearly demonstrates that the contribution of other variables, besides the top three mentioned above and identified using variable presence analysis, is negligible.

Results of the first stage of experiments suggest that the weather inputs windGust_2, windGust, and DewPoint are 1) the most frequently present in $\mathscr{M}_1$ and 2) have the highest contribution to the relative errors of models in $\mathscr{M}_1$ and are sufficient to achieve the accuracy of $\mathscr{M}_1$. In other words these inputs are sufficient to predict energy output with accuracy between 70% and 80% R^2 on the training data.

Table 2
Model ensemble (six models) selected from $\mathscr{M}_2$. Constants are rounded to one decimal place.

Model	c	e_{train}	e_{test}
$-32.1 + 2.9(\sqrt{\text{windGust}_2} + \text{windGust}_2)$	24	0.299	0.426
$112.0 - 3.5{*}10^{-5}(-1956.3 + \text{dewPoint}^2 + \text{windGust}_2^2)^2$	42	0.247	0.472
$-6.4 + 1.3{*}10^{-4}(9 - \sqrt{\text{windGust}_2})^2\text{windGust}_2^2 \cdot (-9.9 + \text{dewPoint} + 2\text{windGust}_2)$	63	0.209	0.146
$-4.5 + 4.3{*}10^{-4}(-8.9 + \sqrt{\text{windGust}_2})(-\sqrt{\text{windGust}_2} + 0.1\text{windGust}_2)\text{windGust}_2 \cdot (-12 + \text{dewPoint}^2 + \text{windGust}_2^2)$	78	0.207	0.149
$-3.1 + 1.5{*}10^{-4}(-3\ \text{dewPoint}\ \text{windGust}_2^2 + (9 - \sqrt{\text{windGust}_2})^2\text{windGust}_2^2 \cdot (-16.3 + \text{dewPoint} + 2\text{windGust}_2))$	121	0.205	0.145
$-11.2 + 9.4{*}10^{-7}(9 - \sqrt{\text{windGust}_2})^2\sqrt{\text{windGust}_2} \cdot (39.4 + 4\text{dewPoint} + 7\text{windGust}_2)\left(\frac{1}{9} + \text{dewPoint} + (10 + 2\text{windGust}_2)^2\right)$	124	0.211	0.145

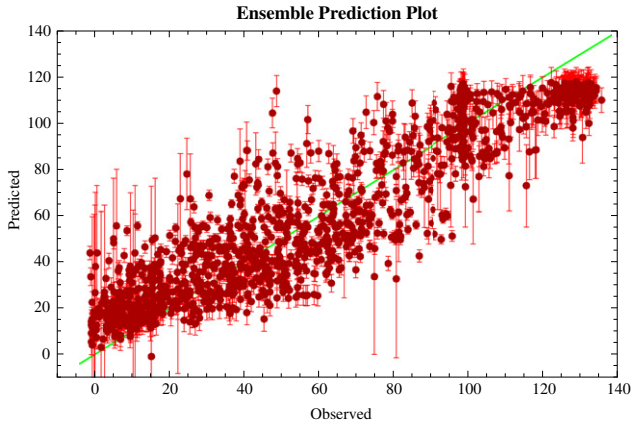


Fig. 10. Ensemble prediction versus observed energy output in July (Test Data) of the final model ensemble. Whiskers correspond to ensemble disagreement measured as a standard deviation between predictions of individual ensemble members for any given input sample.

associated with different levels of prediction accuracy, and by the time needed to perform measurements of associated design spaces. The responsibility of a good model development tool is to supply experts with robust information about the trade-offs.

At the last stage of model analysis, we used the CreateModelEnsemble function of DataModeler to select an ensemble of regression models from $\mathscr{M}_2$, allowing only models with model complexities not exceeding 150. As can be seen in Fig. 8, an increase of model complexity does not provide a sufficient increase in the training error. Since our goal is to predict energy production over a completely new interval of weather conditions (here: July 2011) we choose the simplest models to avoid potential overfitting.

The selected model ensemble consists of six models presented in Table 2. The values of model complexity c, training error e_{train}, and test error e_{test} for six models in the ensemble are listed in this table as well. The test error is evaluated *post facto*, after the models are selected into the model ensemble.

The created model ensemble can now be evaluated on the test data. As mentioned in Section 2.1 ensemble prediction is computed

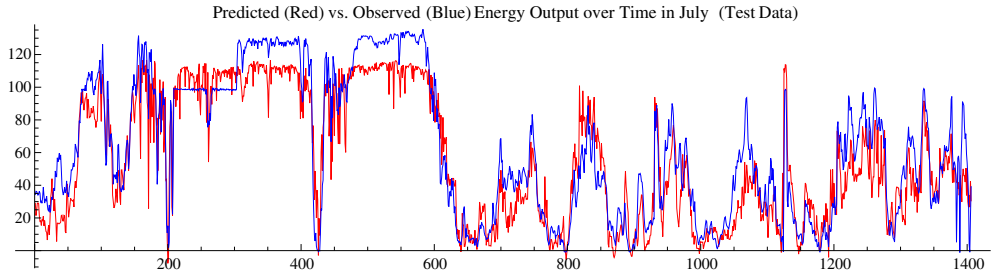


Fig. 11. Ensemble prediction versus actual energy output over time on the test data.

The high correlation between the windGust and windGust$_2$ variables motivated us to select only one of them for the second round of modeling, together with dewPoint, to generate prediction models. Symbolic regression does not guarantee that only one particular input variable out of the set of correlated inputs will be present in final models. It might be that either only one out of two is sufficient to predict the response with the same accuracy, or that both are necessary for success. Our choice was to select windGust$_2$ (as the most frequent variable in the models) together with dewPoint for the second stage of experiments, and to see whether the predictive accuracy of new models in the new two-dimensional design space would not decrease, when compared to the accuracy of $\mathscr{M}_1$ models developed in the original space of 16 dimensions.

4.3. Energy output prediction

The second stage of experiments used only the two input variables windGust$_2$ and dewPoint, with all other symbolic regression settings identical to the first stage experiments. As a result, a new set of one and two-variable models were generated. We again applied a selection procedure to the superset of models by selecting only 25% of robust models closest to the Pareto front with the training error of at most $1-R^2 = 0.30$ and model complexity of at most 250. The resulting set of 587 simplest models, denoted as $\mathscr{M}_2$, is depicted in Figs. 8 and 9.

Fig. 9 is obtained using the VariableContributionTable function of DataModeler, and it exposes the trade-offs for input subspaces and prediction accuracy for energy prediction.

We emphasize here that this it is up to the domain expert to choose an appropriate input space for the energy prediction models. This decision will be guided by the costs and risks

as a median of predictions of individual ensemble members, while ensemble confidence is computed as a standard deviation of individual predictions. We report the normalized root mean squared error of ensemble prediction on the test data as $\text{RMSE}_{\text{Test}} = 12.6\%$.

Fig. 10 presents the predicted versus observed energy output in July 2011, with whiskers corresponding to ensemble confidence. Note that the confidence intervals for prediction are very high for many training samples. This is normal and should be expected when prediction is evaluated well beyond the training data range. Fig. 11 presents ensemble prediction versus actual energy production over time in July 2011.

5. Conclusions

In this study, we showed that wind energy output can be predicted from publicly available weather data with accuracy up to 80% R^2 on the training range and up to 85,5% on the unseen test data. We identified the smallest space of input variables (windGust$_2$ and dewPoint) where reported accuracy can be achieved, and provided clear trade-offs in prediction accuracy when decreasing the input space to the windGust$_2$ variable. We demonstrated that an off-the-shelf data modeling and variable selection tool can be used with mostly default settings to run the symbolic regression experiments as well as variable importance, variable contribution analysis, ensemble selection, and validation.

We are pleased that the presented framework is so simple that it can be used by literally everybody for predicting wind energy production on a smaller scale—for individual wind turbines on private farms or urban buildings, or for small wind farms. For future work, we are planning further study of the possibilities for longer-term wind energy forecasting.

References

[1] Brown BG, Katz RW, Murphy AH. Time series models to simulate and forecast wind speed and wind power. Journal of Climate and Applied Meteorology 1984;23(8):1184–95.

[2] Evolved Analytics LLC. Data Modeler 8.0. Evolved analytics LLC. URL: www.evolved-analytics.com; 2010.

[3] Foley AM, Leahy PG, Marvuglia A, McKeogh EJ. Current methods and advances in forecasting of wind power generation. Renewable Energy 2012;37:1–8.

[4] Jursa R, Rohrig K. Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. International Journal of Forecasting 2008;24:694–709.

[5] Kotanchek M, Smits G, Vladislavleva E. Pursuing the Pareto paradigm: tournaments, algorithm variations & ordinal optimization. In: Genetic programming theory and practice IV. Genetic and evolutionary computation, chapter 12, vol. 5. Springer, ISBN 0-387-33375-4; 11–13 May 2006. p. 167–86.

[6] Koza JR. Genetic programming II: automatic discovery of reusable programs. Cambridge Massachusetts: MIT Press, ISBN 0-262-11189-6; May 1994.

[7] Kramer O, Gieseke F. Analysis of wind energy time series with kernel methods and neural networks. In: Seventh international conference on natural computation (ICNC); 2011. p. 2381–5.

[8] Kramer O, Gieseke F. Short-term wind energy forecasting using support vector regression. In: International conference on soft computing models in industrial and environmental applications. Springer; 2011. p. 271–80.

[9] Kusiak A, Zheng H, Song Z. Short-term prediction of wind farm power: a data mining approach. IEEE Transactions on Energy Conversion 2009;24(1):125–36.

[10] Poli R, Langdon WB, McPhee NF. A field guide to genetic programming. lulu.com;, ISBN 978-1-4092-0073-4; 2008.

[11] Schmidt M, Lipson H. Age-fitness Pareto optimization. In: Genetic programming theory and practice VIII, genetic and evolutionary computation, chapter 8. Springer; 2010. p. 129–46.

[12] Sánchez I. Short-term prediction of wind energy production. International Journal of Forecasting 2006;22(1):43–56.

[13] Webb M, Scuglia S. Wind power: a favoured climate change response. Global Economic Research: Fiscal Pulse (Scotiabank); 2007.