

Wind Energy Prediction Using Machine Learning

Adrian-Nicolae Buturache¹, Stelian Stancu²

¹Economic Cybernetics and Statistics Doctoral School, Bucharest University of Economic Studies, Bucharest, Romania

²Informatics and Economic Cybernetics Department, Bucharest University of Economic Studies, Bucharest, Romania

Email: ad.buturache@yahoo.ro, stelian.stancu@csie.ase.ro

How to cite this paper: Buturache, A.-N., & Stancu, S. (2021) Wind Energy Prediction Using Machine Learning. *Low Carbon Economy*, 12, 1-21.

<https://doi.org/10.4236/lce.2021.121001>

Received: December 8, 2020

Accepted: January 24, 2021

Published: January 27, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Wind energy prediction represents an important and active field in the renewable energy sector. Since renewable energy sources are integrated into existing grids and combined with traditional sources, knowing the amount of energy that will be produced is key in minimizing the operational cost of the wind farm and safe operation of the power grid. In this context, we propose a comparative and comprehensive study of artificial neural networks, support vector regression, random trees, and random forest, and present the pros and cons of implementing the aforementioned techniques. A step-by-step approach based on the CRISP-DM data mining framework reveals the thought process end-to-end, including feature engineering, metrics selection, model selection, or hyperparameter tuning. Using the selected metrics for model evaluation, we provide a summary highlighting the optimal results and the trade-off between performance and the resources expended to achieve these results. This research is also intended to provide guidance for wind energy professionals, filling the gap between purely academic research and real-world business use cases, providing the exact architectures and selected hyperparameters.

Keywords

Artificial Neural Networks, Support Vector Machine, Regression Tree, Random Forest, Hyperparameter Tuning, Renewable Energy, Data-Driven Decision

1. Introduction

Since the end of the 20th century and during the early 21st century, topics related to renewable energy have increasingly come to the fore. Indeed, it is necessary to develop complementary mechanisms to those that already exist in the energy

field. The need is to satisfy electricity demands (International Energy Agency, 2018), followed by the secondary need to replace fossil fuels that contribute to increased pollution levels and whose future existence is questionable (Ritchie & Roser, 2017). The sun, wind, and surface or underground water all represent energy sources that can be captured using various technologies that are under continuous development (Marsh, 2019). Besides the known uses of energy, a new use has arisen owing to technological leaps in electric-powered transportation. The demand for electricity generated by applications in electric transportation will not have a significant impact on total global electricity demand: expectations for growth are 1% by 2030 and 5% by 2050 (Engel, Hensley, Knupfer, & Sahdev, 2018). However, the peak in demand for electricity associated with electrical vehicles recharging routines is expected to have an impact. Accurate predictions of the amount of renewable energy that can be produced are necessary since these installations are becoming integrated into the existing power generation and distribution infrastructure. One of the major challenges for transmission system operators and electricity distribution operators is to find the optimal balance between supply and demand in the market. Moreover, the results of these predictions will represent the basis for future operational and strategic decisions with local and regional impact.

Investments in renewable energy are expected to reach \$ 230 billion in the next five years (Willuhn, 2019). Over time, global electricity consumption has increased steadily. Studies of energy consumption per sector of activity revealed that in the United States of America, energy consumption associated with residential and commercial buildings represented approximately 39% of the total energy consumed across the country, and this is expected to rise to 45.52% by 2035 (U.S. Department of Energy, 2011). In the European Union, the power consumption of residential and commercial buildings accounts for approximately 40% of the total (European Parliament, 2018b) and shows a continuously increasing trend. The European Parliament and the Council of the European Union have defined a legislative framework and set out a strategy that aims to increase the share of renewable energies. However, the transition to green energy is not only achieved using green and non-polluting resources but also by ensuring responsible electricity consumption. DeepMind AI (Evans & Gao, 2016) has created a recommendation system for the use of the heating, ventilation, and air conditioning (HVAC) systems for several Google's buildings. The main task of these HVAC systems is to cool the rooms in which the company's servers are located, including servers of popular websites in Google's portfolio, such as Google Search, YouTube, or Gmail. After the solution's deployment, a 40% reduction was observed in the energy used for cooling.

Of course, this is not a singular initiative, and many similar opportunities have arisen in this field (Kusiak & Xu, 2012; Gao, Li, & Wen, 2019; Wei, Wang, & Zhu, 2017). In accordance with targets set at the EU level through the Renewable Energy Directive (European Commission, 2020), by the end of 2020, 20% of

energy consumed in the European Union should be achieved using renewable resources. All European countries are part of this initiative, each proposing an action plan with specific goals for completion. These plans include intermediate targets for major activity sectors: transport, electricity, heating, and cooling. To these are added recommendations on technologies that should be used to obtain energy from renewable resources, suggested changes and improvements to legislative frameworks to facilitate the deployment of new technologies, and cooperation mechanisms between participating countries (European Commission, 2013). At the end of 2017, 17.5% of the energy consumed in the European Union was obtained from renewable resources, compared to 8.5% in 2004. Over the last 12 years, an average annual growth of 8.5% has been observed in the use of energy from renewable resources. Moreover, in 2017, 7.6% of the energy consumed during transport-related activities was obtained from renewable resources (Eurostat, 2020). The amount of hydroelectricity used has increased slightly over the last decade, while wind power is generated in amounts that are 3.7 times higher than those recorded in 2006 and solar energy in amounts that are 44.4 times higher than those from the same time. For the first time in history, the amount of wind energy produced at the EU level in 2017 exceeded the amount of hydroelectric energy.

For the period from 2021 to 2030, through the 2030 climate and energy framework, the European Commission is targeting at least 40% cuts in greenhouse gas emissions (from 1990 levels), a renewable energy share of at least 32%, and an improvement of at least 32.5% in energy efficiency (European Parliament, 2018a). The amount of wind energy depends not only on the size and number of installed windmills, but also on their geographic locations. Before engaging in detailed analysis, an initial prediction of the amount of energy provided by one or more wind turbines can be obtained from a set of site-related attributes, such as altitude, latitude, longitude, air pressure, date, weather, and so on. Predictability is one of the fundamental requirements in the energy business. Given that wind energy is characterized by non-stationarity, correct prediction of energy production facilitates the adoption of wind energy on an even larger scale than is presently possible. This comparative study adds the right level of depth to fill the gap between academic research and the challenges encountered by professionals working in the energy industry.

2. Literature Review

Machine learning facilitates an alternative to existing analytical models for predicting the amount of energy generated or the performance of wind power installations. Traditional methods rely on complex differential equations systems (Niayifar & Porté-Agel, 2015) requiring significant computational power and are slow to deliver results with acceptable accuracy. The use of machine learning to identify patterns in multidimensional data has proven to be an inspired decision, with the resulting models providing robustness, tolerance to outliers and errors,

and success in dealing with noisy data. Therefore, support vector regression (SVR), regression tree (RT), random forest (RF), and artificial neural networks (ANNs) are selected to model this problem. Depending on the approach adopted, prediction methods can be physical, statistical, machine learning, or hybrid (i.e., a combination of the first three). Moreover, the method can be direct—predicting the output power directly—or indirect—predicting wind direction and speed, which are further used to determine output power based on the power curves. Another classification may be made based on the time frame: short-, medium-, or long-term prediction. Physical modeling uses the complex physical properties of the environment in which the wind turbines are installed. These physical properties must be identified, collected, pre-processed, and further used to make predictions. Fortunately, historical data are not required in such large quantities as for statistical models. However, these systems are extremely complex both in terms of development and operation (Landberg & Watson, 1994; Focken, Lange, & Waldl, 2001; Perez, 2002).

In the case of statistical modeling, certain assumptions must be fulfilled before the actual modeling can begin, and this may contradict the non-stationarity character of the wind. ANN, RT, RF, and SVR can be used for non-linear modeling use cases, such as wind energy prediction. Compared with other models that reduce the error over the training dataset, SVR aims to include as many data points as possible within an exact error interval, namely minimization of the upper bound of the expected risk. Thus, SVR is based on structural risk minimization. RT, RF, and ANN are based on empirical risk minimization. Based on general noise functions, a direct approach to building error intervals as functions of residual distributions can be created (Prada & Dorronsoro, 2015). Based on these error intervals, the model's parameters can be selected and further used to fit the data. SVRs, similar to RT and RF, are sensitive to the selected input variables. Various data preparation steps must be completed before the data are input into these models. Wavelet transform is used to decompose time series data into components that are approximately stationary (De Giorgi, Campilongo, Ficarella, & Congedo, 2014). Least-squares SVR is a variant of SVR that is ultimately reduced to a linear problem, being computationally more efficient than standard SVR or ANN. Combined with wavelet transformation, it can outperform ANN in predicting wind energy output for time frames of up to 24 hours. The same SVR standard approach can be used but with a Euclidian distance-based method for identifying and selecting the data segments that have comparable absolute values with the forecasting reference sequence (Zhu, Zhou, & Fan, 2016). Selected data segments are further used for model training. Compared with other methods, an indirect method is used, with wind speed as the dependent variable. Based on the wind speed prediction and power curves, the actual wind energy generated is calculated. Wind energy production discontinuities, such as ramps, are harmful for the grid. Preferably, predictions should be made in time to prepare and pursue the optimal strategy to minimize the negative impact on the grid (Liu et al., 2016). Before the entire existing data set is in-

put into the SVR, the importance of each variable in predicting the ramps is assessed to reduce the number selected. The technique deployed is called orthogonal testing, and compared with Spearman correlation, Gray correlation, and principal components analysis, is proven to yield better results. All four techniques were used with an SVR model. Ensemble methods such as RF can deliver very good accuracy in predicting wind energy, enabling easy feature importance assessment (Torres-Barrán, Alonso, & Dorronsoro, 2017). RF as a non-parametric model does not require parameter tuning, an area where usually has a lot of effort is spent. Quantile RF is an extension of the standard RF that is used to build confidence intervals for the prediction (Lahouar & Ben Hadj Slama, 2017).

This study highlights the importance of selecting the correct input data and of analyzing and discussing the topic from correlation and feature importance perspectives. In this way, they are used to model only those features that have the right amount of information gain for the studied problem (Wang, Sun, Sun, & Wang, 2017). ANNs are suitable for many applications in the field of wind energy. Among the most important of these are pattern detection, forecasting, monitoring, and control and design optimizations. To ensure a good prediction, it is necessary to select the correct independent variables that will explain the complexity of the studied phenomena. At first glance, the more variables the better: hence, a series of separate problems that must be approached punctually and embedded in a solution that does not lead to diminished advantage caused by the existence of so many variables. If the number of variables considered increases, the amount of data required for the analysis will increase concurrently (Shetty, 2019). Increasing the quantity of data may lead to a decrease in the generalization ability, increasing the likelihood of overfitting or underfitting. The required time for training and computing power also increases. Principal component analysis (PCA) can help to reduce the number of variables considered while preserving most of the information (He, & Liu, 2012).

A series of scientific studies shows that problems of such complexity can be successfully addressed using ANN (Kariniotakis, Stavrakakis, & Nogaret, 1996; Damousis & Dokopoulos, 2001; Barbounis, Theocharis, Alexiadis, & Dokopoulos, 2006). A sporadic and difficult-to-anticipate event, the burst has a negative impact on the results of the prediction (Kolhe, Lin, & Maunuksela, 2011). The use of ANN has proven successful in other instances in which bursts were not present. Post-prediction comparison of the prediction with the actual wind energy production revealed a maximum difference of 6.52%. The neural network used was optimized using genetic algorithms, resulting in a more accurate prediction than that obtained without optimization. The multi-layer perceptron (MLP) offers an architecture that is worth consideration for predicting the electric power produced by wind turbines (Catalao, Pousinho, & Mendes, 2011). The analysis was for a short-term horizon using the Levenberg-Marquardt learning algorithm. The mean absolute percentage error (MAPE) for the training set and the test set was less than 3.26%. An interesting approach is to collect and use da-

ta from two different geographical areas (Flores, Tapia, & Tapia, 2005) that have different particularities—one characterized by strong winds and the other by low winds—in the same model. Another analysis used a feed-forward neural network (FFNN), demonstrating that accuracy depends on the considered time horizon. For long-term predictions, accuracy decreases, while it increases for short-term predictions (Singh, 2016).

3. Theoretical Considerations

3.1. Artificial Neural Networks

The ANN model is inspired by biology and represents an abstraction of the human brain (Jain, Mao, & Mohiuddin, 1996). ANNs are capable of setting the benchmark in various industries for applications such as computer vision, text processing, or speech recognition and various regression and classification use cases. These achievements have generated considerable excitement both inside and outside the machine learning community. FFNNs represent one of the simplest types of ANN. Multiple neurons are organized in layers. Neurons from consecutive layers are connected, and each connection has its own weight. For this architecture, the information passes through the network in one direction from the input layer through the hidden layer to the output layer. The single-layer perceptron (SLP) is the simplest type of FFNN and does not contain any hidden layer. If the network contains one or more hidden layers, it is categorized as an MLP (Figure 1).

In the neural network, the computation unit is called a neuron. Each neuron receives inputs from other neurons inside the network or from outside the network through the input layer and computes an output. As in Figure 2, each input has an associated weight (w_i), and weight values are assigned based on the importance of each input.

Function f is applied to the weighted sum of the inputs (x_i). Besides, the weights and numerical inputs, another term—“bias”—can be added to the neuron. The output of the neuron becomes

$$y = f(u) = f(w_1x_1 + w_2x_2 + w_3x_3 + b) \quad (1)$$

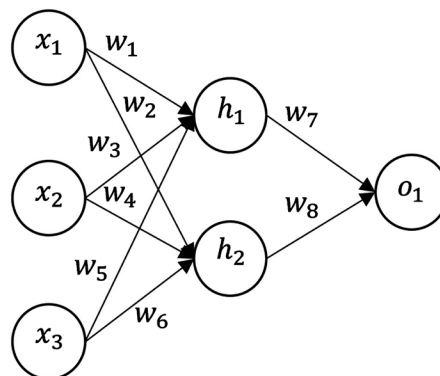


Figure 1. Multi-layer perceptron.

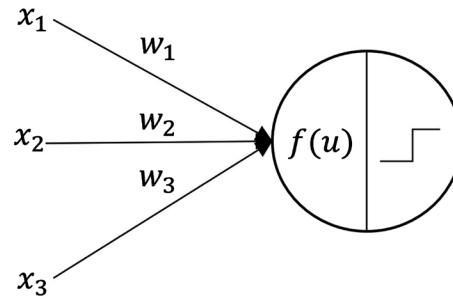


Figure 2. Neuron abstraction.

The activation function introduces non-linearity into the output neuron. This feature of neural networks is important because many of the potential use cases using real-world data will be non-linear. For a step function, the output would be something like

$$y = f(u) = \begin{cases} 1, & \text{if } wx + b \geq \tau \\ 0, & \text{if } wx + b \leq \tau \end{cases} \quad (2)$$

where τ represents the threshold.

The learning process within the ANNs is achieved by adjusting the synaptic weights during the training. The model's cost function is computed based on the output layer results regarding how close or far the prediction is from the actual values. This cost function is also known as a "loss" or "error" function, depending on the bibliographic resource consulted:

$$\text{Mean square error} = \frac{\sum_{i=1}^n (y_i - y_i^p)^2}{n} \quad (3)$$

Mean square error or L2 loss is one of the most widely used loss functions for regression problems. Mean square error represents the sum of the squared distances between predicted values and target variables. When the total error approaches a reasonable value or another criterion is reached (i.e., number of epochs or training time), the training phase is stopped. Therefore, the model's objective function is to minimize the error. During the training, a series of hyperparameters must be refined to achieve a performant model. First, taking into consideration the quantity of data that is passed through the network during the training, the batch size must be defined. Batch size may vary from 1 to n , where n is the total number of samples within the training set and represents the number of samples propagated through the network. With respect to the cost function and the selected optimization function, the model's synaptic weights are updated. Synaptic weights—or, in some cases, learning rates—are updated by the optimization algorithm in accordance with certain rules. Weights initialization represents one of the most important decisions that must be made. During the learning process, the model's synaptic weights acquire knowledge. For initial weights, if values are too small or too large, issues may arise in relation to vanishing or exploding gradients. For constant values, different knowledge acquisitions may appear across networks. However, both solutions negatively impact

the network's convergence to the minima and how the resources, such as training time or computational power, are spent.

To overcome these drawbacks, strategies for selecting weights from distributions are proposed. [LeCun, Bottou, Orr, and Müller \(2000\)](#) proposed that they be drawn from a distribution with a certain variance and zero mean. LeCun et al. normal initialization uses a truncated normal distribution centered in zero having a standard deviation:

$$\sigma_{\omega} = \sqrt{\frac{1}{\text{input_dim}}} \quad (4)$$

This strategy considers the number of inputs in the weight tensor, `input_dim`. LeCun uniform initialization proposes an interval from which the initial weights are selected. Upper and lower bounds are described as follows:

$$\text{lower bound} = -\sqrt{\frac{3}{\text{input_dim}}} \quad (5)$$

$$\text{upper bound} = \sqrt{\frac{3}{\text{input_dim}}} \quad (6)$$

Xavier initializations, normal, and uniform, take into account both the number of inputs in the weight tensor and the number of outputs in the weight tensor ([Glorot & Bengio, 2010](#)). The distribution is created in the same way as in [LeCun et al.'s \(2000\)](#) approach, but the additional variable related to the output is accommodated. Normal distribution is used for Xavier normal initialization:

$$\sigma_{\omega} = \sqrt{\frac{2}{\text{input_dim} + \text{output_dim}}} \quad (7)$$

and uniform distribution is used for Xavier uniform initialization:

$$\text{lower bound} = -\sqrt{\frac{6}{\text{input_dim} + \text{output_dim}}} \quad (8)$$

$$\text{upper bound} = \sqrt{\frac{6}{\text{input_dim} + \text{output_dim}}} \quad (9)$$

Rectified linear unit (RELU) is one of the most widely used activation functions among machine learning practitioners, being less computationally expensive than sigmoid or tanh functions since the mathematical operations deployed are simpler. Owing to its nature, only some neurons are activated. This behavior adds computational efficiency but simultaneously generates situations in which some neurons are excluded from training by no longer being activated ([Krizhevsky, Sutskever, & Hinton, 2012](#)).

$$f(z_i) = \max(0, z_i) \quad (10)$$

The number of hidden neurons and hidden layers is a matter of domain knowledge and testing. Some guidance may be found in the existing research in this area, but a clear answer about the best architecture will be found only through trial-and-error ([Hunter, Yu, Pukish, Kolbusz, & Wilamowski, 2012](#);

Madhjarasan & Deepa, 2016; Panchal, Ganatra, Kosta, & Panchal, 2011).

Adam is a method for stochastic optimization that computes for each parameter its own learning rate (Kingma & Ba, 2014). The name Adam is inspired by adaptive moment estimation and combines the advantages of both RMSProp (Hinton, Srivastava, & Swersky, 2020) and Adagrad (Lydia & Francis, 2019) into a better memory and computational solution. Adam updates the equation as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t \quad (11)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (12)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (13)$$

where t = time step, m_t = update-biased first moment estimate, v_t = update-biased second raw moment estimate, \hat{m}_t = compute bias-corrected first moment estimate, \hat{v}_t = compute bias-corrected first moment estimate, and β_1^t and β_2^t = exponential decay rates for the moment estimates at time step t .

3.2. Support Vector Regression

The support vector machines (SVM) principle is sophisticated yet simple to implement. SVM uses the structural risk minimization inductive principle to achieve a reasonable generalization on limited data (Smola & Schölkopf, 2004). SVM can solve both classification and regression problems, the theoretical foundations being common up to a point. SVR is an instance of SVM that deals with regressions. While linear regression or FFNN aim to minimize the error in the SVR, the aim is to fit the error within a fixed threshold (Basak, Pal, Ch, & Patranabis, 2007). In this way, SVR is associated with problems in selecting the right decision boundary. The best fit is achieved when the maximum number of data points is included between the boundaries. Effort is invested in deciding where the decision boundary should be placed and in setting the distances ε and $-\varepsilon$ from the hyperplane in such a way that the data records closest to the hyperplane are inside the boundaries, visually represented in Figure 3.

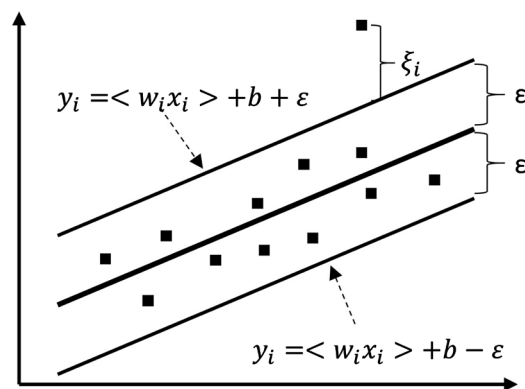


Figure 3. Support vector regression hyperplane and decision boundaries.

For a given data set split in training and testing sub-sets, the pairs of training points are $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_i, y_i) \in X \times R$, where X is the input space for the instance R^d and $i = 1, 2, \dots, n$ (Taylor, 2020). The function $f(x)$ must have the largest deviation ε from the hyperplane and must be as flat as possible. For a linear function:

$$f(x) = \langle w, x \rangle + b \tag{14}$$

where $w \in X$, $b \in R$ and $\langle \cdot, \cdot \rangle$ represents the dot product in X .

Seeking a flattened $f(X)$ means the value of w needs to be minimized. Hence:

$$\text{minimize } \frac{1}{2} \|w\|^2 \tag{15}$$

with the associated constraints

$$y_i - \langle w_i, x_i \rangle - b \leq \varepsilon \tag{16}$$

$$y_i - \langle w_i, x_i \rangle + b \leq \varepsilon \tag{17}$$

However, this is the ideal case. For errors that exceed the boundaries—that is, errors larger than ε —slack variables, ξ_i and ξ_i^* , are introduced in the optimization (15).

$$\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \tag{18}$$

with the new associated constraints:

$$y_i - \langle w_i, x_i \rangle - b \leq \varepsilon + \xi_i \tag{19}$$

$$y_i - \langle w_i, x_i \rangle + b \leq \varepsilon + \xi_i^* \tag{20}$$

where $\xi_i, \xi_i^* \geq 0$.

The new hyperparameter, C , can decrease or increase the tolerance on the data points that exceed the boundaries, impacting the flatness of f . The greater the value of C , the greater the tolerance on deviations larger than ε . For the features that are linearly non-separable, one option to make them linearly separable is to map them to a high-dimensional feature space, moving from a linear to a non-linear kernel and, based on (Smola & Schölkopf, 1998) and (Gani, Taleb, & Limam, 2010), introducing radial basis function (RBF) kernel defined as:

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \tag{21}$$

$$\Upsilon = \frac{1}{2\sigma^2} \tag{22}$$

where σ , length scale, is RBF kernel's parameter. If Υ is too small, the complexity of the studied problem cannot be covered by the model, resulting in a solution similar to the linear one but including all training examples. If Υ is too large, the overfit will appear even if the regularization parameter, C , appears to prevent it.

3.3. Regression Trees

Decision trees are part of the supervised learning branch, and based on the outcome of the model decision trees, regression trees may result when the outcome is continuous while classification trees ensue when the outcome is discrete. Tree-based models are efficient and simple when the use case implies the involvement of several variables.

The training set is split into smaller data sub-sets as in **Figure 4** using a greedy approach to divide the input space. This procedure is known as recursive binary splitting (Steorts, 2020). Beginning with all examples belonging to the same region and splitting them into multiple regions also makes this approach top-down. Node splitting is carried out according to $feature_1 \geq \text{threshold}$ or $feature_1 < \text{threshold}$. In the non-overlapping regions where leaves are placed, the predicted value represents the average over the training examples assigned to each region. In the left branch, another feature is added, again splitting the branch based on a threshold. In this way, regions R_1 , R_2 , and R_3 are built. The points at which the predictor space is split are called nodes, the segments that are connecting the nodes are called branches, and those at the bottom end are called leaves. The feature where the first split is made is the most important in predicting the dependent variable. Further, the second feature is important for the examples of only one branch and less so for the examples of the other branch. Tree performance depends on the extent to which the residual sum of squares (RSS) is minimized:

$$RSS = \sum_{n=1}^n \sum_{i \in R_n} (y_i - \widehat{y}_{R_n})^2 \quad (23)$$

where \widehat{y}_{R_n} corresponds to the n^{th} region training examples mean.

The splitting procedure must have a rule for stopping; otherwise, overfitting will occur (e.g., one leaf containing only one instance), leading to poor prediction performance. The splitting procedure must be constrained to avoid overfitting. To do this, tree parameters are set to refine the way in which splitting occurs. The minimum number of samples inside a node gives the threshold above which a split can happen. A tree's maximum depth need not be set if the minimum number of samples inside a node is fixed and vice versa. The minimum number of samples inside a leaf provides a threshold below which splitting cannot go. Node splitting decisions can be based on a selected metric or random.

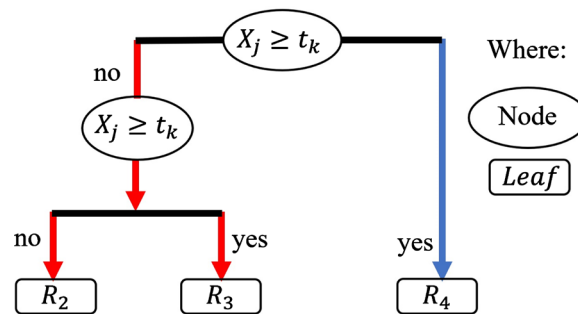


Figure 4. Decision tree logic.

Since the desired state implies having the simplest tree for the studied problem, pruning can be used to reduce the complexity, removing the branches with low importance. Complexity is given by the number of splits. Pruning can be performed manually by removing leaves step-by-step while reassessing the error on testing data using cross-validation. Manual pruning is time-consuming, requiring that every possible configuration be considered. Pruning can be performed automatically by deploying cost complexity pruning (Taylor, 2020). A learning parameter (α) is added, and a sequence of subtrees with the best performance is obtained based on it:

$$\min \sum_{n=1}^T \sum_{x_i \in R_n} (y_i - \widehat{y}_{R_n})^2 + \alpha |T| \quad (24)$$

where $|T|$ represents the number of terminal nodes of the tree T , R_n is the region in which the n^{th} terminal node is placed, and \widehat{y}_{R_n} is the prediction linked with R_n , constraining the value of $|T|$ and using Lagrange multipliers:

$$\Delta_g = \sum_i (y_i - \widehat{y}_{R_n})^2 + \lambda (|T| - c_\alpha) \quad (25)$$

This becomes a discrete optimization problem of minimizing T and λ . When a final leaf node is reached, the prediction is made. Regression trees are sensitive to the data used for training. If training data are replaced with another sample, the resulting regression tree solution may be different to the first. Simultaneously showing significant risk of overfitting, predilection for finding the local optima, and cost-intensiveness in terms of computational power, regression trees can be used in ensemble models like RF.

3.4. Random Forest

RF, an ensemble model, consists of a collection of decision tree models (Figure 5). Each tree that makes up the RF algorithm is trained on a randomly selected subset of data and makes its own prediction. RF prediction accuracy can be significantly improved compared with single decision trees (Breiman, 2001). The prediction of the entire RF model represents the average of the predictions provided by every tree. Bootstrap refers to the simple random samples from an original dataset selected with replacement and provides a better understanding of the bias and variance in the considered use case (Kotsiantis, 2011). Bagging or bootstrap aggregation is an ensemble method that combines the predictions made by different RTs over different bootstrap samples to make a more accurate prediction than any individual tree. For supervised learning, an important step is represented by variable selection. Taking the optimal subset of variables reduces the model's complexity, increases model generalization capability, and reduces the time and computational power needed for training (Ben Ishak, 2016). Comparing the parameters set for the model with those set for regression tree, a new one is added: the number of decision trees that constitute the RF.

Out-of-bag error (OOBE) is similar to cross-validation and represents the average of all predictions made on unseen data (i.e., sub-sets not used during the

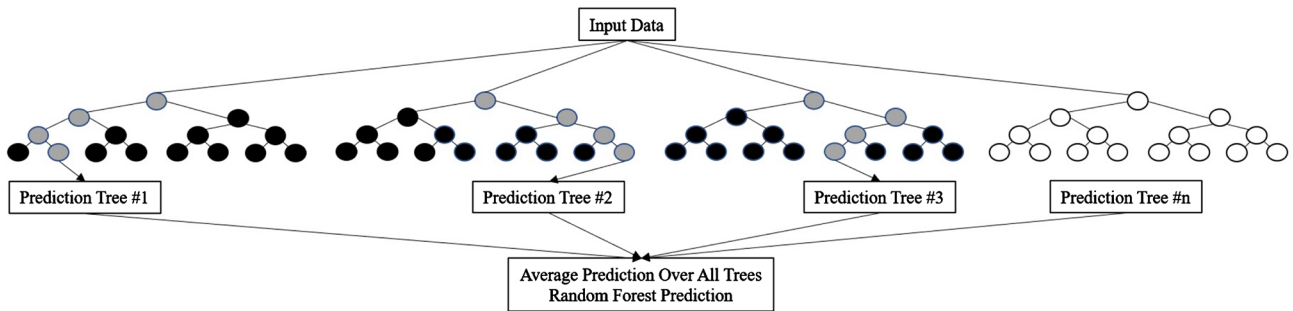


Figure 5. Random Forest logic.

training).

$$\text{OOBE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (26)$$

where \hat{y}_i is the predicted value.

Variable importance is calculated by making random permutations of a feature across multiple trees, calculating the difference between the OOBE obtained after each permutation and the original OOBE. If the error increases compared with the original OOBE, then the feature is relevant for the analysis. Both regression trees and RFs are sensitive to the data on which they are trained.

To avoid overfitting and to better evaluate the models' generalization capabilities, a cross-validation technique is used (Fawcett & Provost, 2013). The first step is to split the entire data set into training and testing data. In this paper, the training data consist of 80% of the entire data set, while the remaining 20% is the holdout subset. The second step is to split the training data set into k -folds. For a given k , the training data set is split into k sub-sets. Training process is done using $k-1$ folds or $\frac{k-1}{k}$ of the training data subset, while the evaluation is made on $\frac{1}{k}$ remaining data. For the same k , a loop of k training and testing sequences are completed, with each fold being $k-1$ times part of the training set and 1-time part of the testing set. The metric computed at the end of testing for each k -fold combination is averaged, resulting in the model's performance. The third step is to test the model against the initial holdout data set.

4. Methodology and Use Case

The methodology used for data mining is Cross-Industry Standard Process for Data Mining (CRISP-DM). The main steps used in this methodology are business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The first step, business understanding, involves identifying and understanding the project objectives and crystallizing them into a complete definition of the problem under review. The details of this phase are covered in the Introduction. Data understanding covers the entire process of data collection and exploratory data analysis. At the end of this step, the researcher will know whether the available data are of the right quality and quantity to continue the

project. If necessary, multiple iterations are permitted to match the requirements. Data used in the analysis were made available by [Open Power System Data \(2020\)](#). Two data sources were aggregated ([Table 1](#)) to obtain the final data set containing information about the weather and energy output of the wind turbines under consideration. Open Power System Data is a portal aggregating multiple data sources across the European Union. The information sources include, among others, weather data and energy production and generation capacity by technology and country. The data is available at three resolutions: 15 minutes, 30 minutes, and 60 minutes.

Data preparation is the most time-consuming part of the data mining process. This typically represents around 80% of the effort. This step involved the transformation of the initial raw data into the data used for predictive modeling. Among the activities associated with this step are joining data from different sources, identifying extreme values, identifying missing values, identifying data that do not have the correct format, feature engineering, and data normalization. Once the potentially risky data are identified, techniques are applied to limit or remove the effects. At the end of this step, the data are in the optimal form. On the feature engineering side, out-of-time stamps were created for additional variables consisting of month, day of month, and hour. Once categorical encoding had been completed, the total number of dependent and independent variables became 77. One of the problems encountered was associated with multicollinearity. Highly correlated independent variables were dropped. Based on the current approach, measuring wind speed in different places does not provide additional information. Moreover, air density was also dropped since it is a function of air pressure and air temperature. The modeling step begins with

Table 1. Data set description.

	Mean	Std	Min	1 st quartile	Median	3 rd quartile	Max
Energy	8766.86	6846.41	135	3617	6944	11,800	33,626
v_1	2.94	1.19	0.75	2.07	2.72	3.58	7.93
v_2	4.3	1.61	1.35	3.14	3.95	5.11	11.16
v_{50m}	6.06	2.09	1.96	4.63	5.72	7.1	14.92
h_1	2.69	0.13	2.46	2.58	2.73	2.81	2.83
h_2	10.69	0.13	10.46	10.58	10.73	10.81	10.83
z_0	0.2	0.03	0.14	0.17	0.21	0.23	0.24
ρ	1.21	0.03	1.13	1.18	1.21	1.24	1.31
p	9.86E+04	8.58E+02	9.55E+04	9.81E+04	9.86E+04	9.91E+04	1.01E+05
T	9.22	7.58	-8.55	2.91	8.09	15.52	28.7

Notes: *energy* = actual wind energy production, v_1 and v_2 = wind speed 2 and 10 meters above displacement height, v_{50m} = wind speed 50 meters above ground, h_1 and h_2 = height above ground corresponding to v_1 and v_2 , z_0 = roughness length, ρ = air density at surface, p = air pressure at surface, T = temperature 2 meters above displacement height.

model selection. In the case of wind energy, more methods are available to achieve the prediction. Considering the problem statement and the previous publications of researchers dealing with similar use cases, SVR, RT, RF, and ANNs were selected for predicting wind energy production. An optimization for the number of folds, k , used for cross-validation was performed. Considering the performance and the training time, the selected value for k was equal to 3. Model evaluation facilitates comparison of the results and selection of the technical solution best suited to the problem at hand. It assesses how well the selected models succeed in generalizing using the testing data. At the end of the entire process, the best performing model is chosen for deployment in production. From among those available, coefficient of determination (R^2), mean absolute error (MAE) and root mean squared error (RMSE) were selected for the evaluation step.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)}{\sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (27)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (28)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (29)$$

where y = actual value, \hat{y} = predicted value, and n = number of samples.

The deployment phase involves integrating the final solution into production and creating a mechanism that allows the consumption of new data generated in the system. In this paper, the solution will not be deployed in production, but theoretical considerations and the results of the study prepare the field for the development of a technical solution that can be deployed to support real operations in the wind turbine industry.

5. Results

The purpose of the study was to assess the capability of the selected models to predict wind energy production. First, the coefficient of determination is the percentage of variance in the dependent variable explained by the model. MAE and RMSE are scale-dependent measures of accuracy that enable comparisons between different models on the same data set. The fourth metric is represented by training time. In the overall context, delivering the prediction just-in-time is a desired feature. Based on the selected modeling paradigm, quantity of data, prediction horizon, available hardware and its limits, training time can become a decisive factor in selecting the final solution. All computations were performed using a Dell Precision 7350 equipped with Nvidia Quadro P2000, 2.5 GHz Intel Core i5-8400H CPU, 32 GB RAM, Python 3.6.10, Windows 10. **Table 2** indicates the optimized parameters and the selected values at the end of the optimization. Overall, ANN provides greater flexibility than the other three algorithms. However, SVR, RT, and RF parameters look like they are coming with a more com-

mune sense than ANN.

The flexibility in selecting the parameters has both positive and negative aspects. Regarding the capability of capturing the entire phenomenon under review, ANN showed the best performance, closely followed by RF and SVR. The negative aspect shown in **Figure 6** is represented by the training time and computational power invested in determining the best combination of parameters. At this stage, training time has no impact on whether or not the prediction is delivered in time, since the prediction horizon is 1 hour ahead and the maximum training time represents only up to 21% of this time frame.

Beyond R^2 it is important to understand the consequences of the trade-off between performance, complexity, and training time. Although the R^2 values of SVR, RF, and ANN may look similar to one other, assessing MAE and RMSE, another perspective is revealed. ANN delivers an MAE that is 20% smaller than RF, 35% smaller than SVR, and 59% smaller than RT (**Figure 7**). If the

Table 2. Parameter selection by model.

Algorithm	Optimized Parameters	Total algorithms tested	Best Parameters combination
SVR	C, ξ, γ	72	$C = 5, \xi = 0.00001, \gamma = \text{scale}$
Regression Tree	α , min samples split, min samples leaf, min weight fraction, splitter	900	$\alpha = 1,$ min samples leaf = 1, min samples split = 3, min weight fraction = 0, splitter = random
Random Forest	min samples leaf, min samples split, number of trees	1250	min samples leaf = 1, min samples split = 2, number of trees = 2375
ANN	batch size, epochs, no. of neurons, no. of hidden layers, weights initialization, activation, optimization algorithm	40,000	batch size = 8, epochs = 150, no. of neurons hidden layer = 141, no. of hidden layers = 2, weights initialization = Xavier Uniform, activation = RELU, optimization = Adam

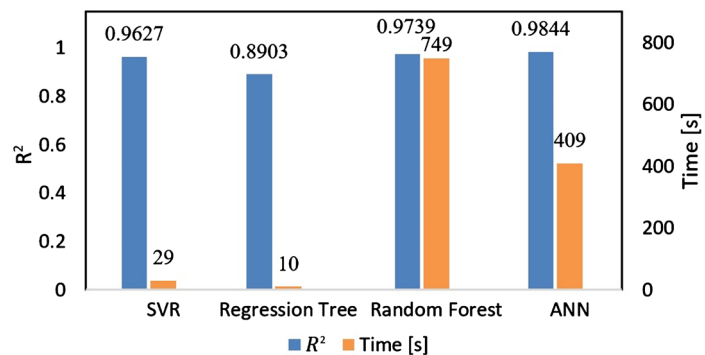


Figure 6. Comparison of R^2 and training time for the selected algorithms.

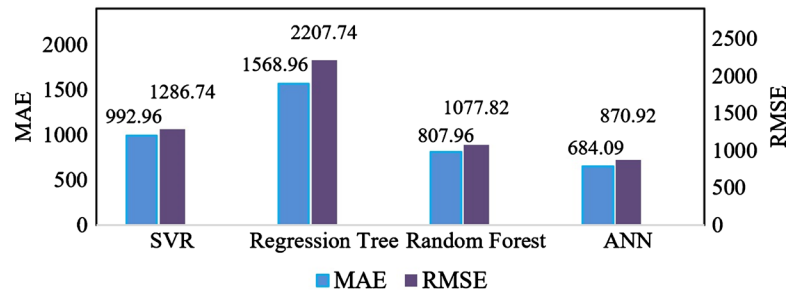


Figure 7. Comparison of MAE and RMSE for the selected algorithms.

performance is assessed considering the MW of electricity, the differences between the models will become clearer. ANNs are more complex than the SVR, RT, and RF, but if the requirements and limitations of the use case are understood, the model's performance will be prioritized over training time and complexity.

Considering performance alone and removing hardware constraints, ANN can deliver benchmark results for these types of real-world business cases. Practitioners working on these projects may encounter challenges related to data quality and availability. Hence, the data provided by the Open Power System Data is reliable. In the case of a smaller time horizon in which the prediction must be completed, training time may be a decisive factor in the model's selection. However, that would be another data mining project for which this study's findings could be considered but would not generally be accepted as valid.

6. Conclusion

This end-to-end approach based on the CRISP-DM framework offers some guidelines for future projects. Well-tuned ANNs can deliver accurate predictions for forecasting wind energy production but carry costs associated with the resources, time, and computational power required to find the optimum hyperparameter combination. Tree-based models provide transparency that is lacking in black box algorithms such as ANN. SVR may have been the best solution proposed if a single metric considering both performance and training time had been proposed. Wind power prediction remains an open topic despite the promising results. For the analyzed time frame, no special event or natural hazard occurred; the wind energy production as a dependent variable was consistently associated with the correct independent variables. A key feature of this research is the insights it offers into how data pre-processing and the model's optimization were performed.

Wind energy production is closely related to weather conditions. Bearing in mind the unpredictable nature of weather, the larger prediction time horizon, and the greater degree of uncertainty, even the metrics seem to be good. Wind energy prediction errors can be assumed to be within a certain limit and can be overcome through the addition of installations for storing energy when the production is greater than the demand, so that the grid does not become overloaded

and so the energy can be delivered when the pick-in demand is not fulfilled by the wind farms. A feasible solution could be to run multiple machine learning algorithms in parallel to short-, medium-, and long-term predictions to support both operational and strategic decisions. An accurate prediction of the amount of electricity produced from renewable resources is a step toward optimizing the entire ecosystem. Improvements in electricity generation capacity, stability, and predictability of operating mode are carefully pursued with the hope of achieving the most efficient integration and easing the transition from traditional to renewable energy resources.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- Barbounis, T. G., Theocharis, J. B., Alexiadis, M. C., & Dokopoulos, P. S. (2006). Long-Term Wind Speed and Power Forecasting Using Local Recurrent Neural Network Models. *IEEE Transactions on Energy Conversion*, *21*, 273-284. <https://doi.org/10.1109/TEC.2005.847954>
- Basak, D., Pal, S., Ch, D., & Patranabis, R. (2007). Support Vector Regression. *Neural Information Processing—Letters and Reviews*, *11*, 203-224.
- Ben Ishak, A. (2016). Variable Selection Using Support Vector Regression and Random Forests: A Comparative Study. *Intelligent Data Analysis*, *20*, 83-104. <https://doi.org/10.3233/IDA-150795>
- Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5-32. <https://doi.org/10.1023/A:1010933404324>
- Catalao, J. P. S., Pousinho, H. M. I., & Mendes, V. M. F. (2011). Short-Term Wind Power Forecasting in Portugal by Neural Networks and Wavelet Transform. *Renewable Energy*, *36*, 1245-1251. <https://doi.org/10.1016/j.renene.2010.09.016>
- Damousis, I. G., & Dokopoulos, P. (2001). A Fuzzy Expert System for the Forecasting of Wind Speed and Power Generation in Wind Farms. *PICA 2001. Innovative Computing for Power—Electric Energy Meets the Market. 22nd IEEE Power Engineering Society International Conference on Power Industry Computer Applications*, Sydney, 20-24 May 2001, 63-69.
- De Giorgi, M. G., Campilongo, S., Ficarella, A., & Congedo, P. (2014). Comparison between Wind Power Prediction Models Based on Wavelet Decomposition with Least-Squares Support Vector Machine (LS-SVM) and Artificial Neural Network (ANN). *Energies*, *7*, 5251-5272. <https://doi.org/10.3390/en7085251>
- Engel, H., Hensley, R., Knupfer, S., & Sahdev, S. (2018). *The Potential Impact of Electric Vehicles on Global Energy Systems*. <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-potential-impact-of-electric-vehicles-on-global-energy-systems>
- European Commission (2013). *Renewable Energy Directive: Cooperation Mechanisms*. <https://ec.europa.eu/energy/en/topics/renewable-energy/renewable-energy-directive/cooperation-mechanisms>
- European Commission (2020). *Climate Strategies and Targets: 2030 Climate & Energy*

- Framework. https://ec.europa.eu/clima/policies/strategies/2030_en
- European Parliament (2018a). *Directive (EU) 2018/2001 of the European Parliament and of the Council of 11 December 2018 on the Promotion of the Use of Energy from Renewable Sources*. <https://eur-lex.europa.eu/eli/dir/2018/2001/oj>
- European Parliament (2018b). *Directive (EU) 2018/844 of the European Parliament and of the Council of 30 May 2018 Amending Directive 2010/31/EU on the Energy Performance of Buildings and Directive 2012/27/EU on Energy Efficiency*. <https://eur-lex.europa.eu/eli/dir/2018/844/oj>
- Eurostat (2020). *Renewable Energy Statistics*. https://ec.europa.eu/eurostat/statistics-explained/index.php/Renewable_energy_statistics
- Evans, R., & Gao, J. (2016). *DeepMind AI Reduces Google Data Centre Cooling Bill by 40%*. <https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/>
- Fawcett, T., & Provost, F. (2013). *Data Science for Business*. Newton, MA: O'Reilly Media.
- Flores, P., Tapia, A., & Tapia, G. (2005). Application of a Control Algorithm for Wind Speed Prediction and Active Power Generation. *Renewable Energy*, 30, 523-536. <https://doi.org/10.1016/j.renene.2004.07.015>
- Focken, U., Lange, M., & Waldl, H.-P. (2001). Previento—A Wind Power Prediction System with an Innovative Upscaling Algorithm. In *European Wind Energy Conference & Exhibition* (pp. 826-829). Copenhagen: European Wind Energy Association.
- Gani, W., Taleb, H., & Limam, M. (2010). Support Vector Regression Based Residual Control Charts. *Journal of Applied Statistics*, 37, 309-324. <https://doi.org/10.1080/02664760903002667>
- Gao, G., Li, J., & Wen, Y. (2019). *Energy-Efficient Thermal Comfort Control in Smart Buildings via Deep Reinforcement Learning*. arXiv:1901.04693. <https://arxiv.org/abs/1901.04693>
- Glorot, X., & Bengio, Y. (2010). Understanding the Difficulty of Training Deep Feed forward Neural Networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* (pp. 249-256). Sardinia: ML Research Press.
- He, D., & Liu, R. (2012). Ultra-Short-Term Wind Power Prediction Using ANN Ensemble Based on PCA. In *Proceedings of the 7th International Power Electronics and Motion Control Conference* (pp. 2108-2112). Harbin: IEEE.
- Hinton, G., Srivastava, N., & Swersky, K. (2020). *Lecture 6a: Overview of Mini-Batch Gradient Descent*. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
- Hunter, D. K., Yu, H., Pukish, M. S., Kolbusz, J., & Wilamowski, B. M. (2012). Selection of Proper Neural Network Sizes and Architectures—A Comparative Study. *IEEE Transactions on Industrial Informatics*, 8, 228-240. <https://doi.org/10.1109/TII.2012.2187914>
- International Energy Agency (2018). *World Energy Outlook 2018*. <https://www.iea.org/weo2018/electricity/>
- Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial Neural Networks: A Tutorial. *Computer*, 29, 31-44. <https://doi.org/10.1109/2.485891>
- Kariniotakis, G., Stavrakakis, G., & Nogaret, E. F. (1996). Wind Power Forecasting Using Advanced Neural Networks Models. *IEEE Transactions on Energy Conversion*, 11, 762-767. <https://doi.org/10.1109/60.556376>
- Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representations*.

- Kolhe, M., Lin, T. C., & Maunuksela, J. (2011). GA-ANN for Short-Term Wind Energy Prediction. *Asia-Pacific Power and Energy Engineering Conference*, Wuhan, 25-28 March 2011, 1-6. <https://doi.org/10.1109/APPEEC.2011.5749029>
- Kotsiantis, S. (2011). Combining Bagging, Boosting, Rotation Forest and Random Subspace Methods. *Artificial Intelligence Review*, 35, 223-240. <https://doi.org/10.1007/s10462-010-9192-8>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*. <http://www.cs.toronto.edu/~hinton/absps/imagenet.pdf>
- Kusiak, A., & Xu, G. L. (2012). Modeling and Optimization of HVAC Systems Using Dynamic Neural Network. *Energy*, 42, 241-250. <https://doi.org/10.1016/j.energy.2012.03.063>
- Lahouar, A., & Ben Hadj Slama, J. (2017). Hour-Ahead Wind Power Forecast Based on Random Forests. *Renewable Energy*, 109, 529-541. <https://doi.org/10.1016/j.renene.2017.03.064>
- Landberg, L., & Watson, S. J. (1994). Short-Term Prediction of Local Wind Conditions. *Boundary-Layer Meteorology*, 70, 171-195. <https://doi.org/10.1007/BF00712528>
- LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (2000). *Efficient BackProp*. <http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf>
- Liu, Y., Sun, Y., Infield, D., Zhao, Y., Han, S., & Yan, J. (2016). A Hybrid Forecasting Method for Wind Power Ramp Based on Orthogonal Test and Support Vector Machine (OT-SVM). *IEEE Transactions on Sustainable Energy*, 8, 451-457. <https://doi.org/10.1109/TSTE.2016.2604852>
- Lydia, A. A., & Francis, F. S. (2019). Adagrad—An Optimizer for Stochastic Gradient Descent. *International Journal of Information and Computing Science*, 6, 566-568.
- Madhiarasan, M., & Deepa, S. N. (2016). Comparative Analysis on Hidden Neurons Estimation in Multi Layer Perceptron Neural Networks for Wind Speed Forecasting. *Artificial Intelligence Review*, 48, 449-471. <https://doi.org/10.1007/s10462-016-9506-6>
- Marsh, J. (2019). *Five Types of Renewable Energy Sources: Best Alternatives to Fossil Fuels*. <https://news.energysage.com/five-types-of-renewable-energy-sources/>
- Niayifar, A., & Porté-Agel, F. (2015). A New Analytical Model for Wind Farm Power Prediction. *Journal of Physics: Conference Series*, 625, 012039. <https://doi.org/10.1088/1742-6596/625/1/012039>
- Open Power System Data (2020). *Open Power System Data: A Free and Open Data Platform for Power System Modelling*. <https://open-power-system-data.org/>
- Panchal, G., Ganatra, A., Kosta, Y. P., & Panchal, D. (2011). Behaviour Analysis of Multi-layer Perceptrons with Multiple Hidden Neurons and Hidden Layers. *International Journal of Computer Theory and Engineering*, 3, 332-337. <https://doi.org/10.7763/IJCTE.2011.V3.328>
- Perez, M. (2002). Wind Forecasting Activities. In *Proceedings of the First IEA Joint Action Symposium* (pp. 197-214). Norrköping: FOI-Swedish Defence Research Agency.
- Prada, J., & Dorronsoro, J. R. (2015). SVRs and Uncertainty Estimates in Wind Energy Prediction. In I. Rojas, G. Joya, & A. Catala (Eds.), *International Work-Conference on Artificial Neural Networks 2015: Advances in Computational Intelligence* (pp. 564-577). Palma de Mallorca: Springer. https://doi.org/10.1007/978-3-319-19222-2_47
- Ritchie, H., & Roser, M. (2017). *Fossil Fuels*. <https://ourworldindata.org/fossil-fuels>
- Shetty, B. (2019). *Curse of Dimensionality*. <https://builtin.com/data-science/curse-dimensionality>

- Singh, V. (2016). Application of Artificial Neural Networks for Predicting Generated Wind Power. *International Journal of Advanced Computer Science and Applications*, 7, 250-253. <https://doi.org/10.14569/IJACSA.2016.070336>
- Smola, A. J., & Schölkopf, B. (1998). On a Kernel-Based Method for Pattern Recognition, Regression, Approximation and Operator Inversion. *Algorithmica*, 22, 211-231. <https://doi.org/10.1007/PL00013831>
- Smola, A. J., & Schölkopf, B. (2004). A Tutorial on Support Vector Regression. *Statistics and Computing*, 14, 199-222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>
- Steorts, R. C. (2020). *Tree Based Methods: Regression Trees*. http://www2.stat.duke.edu/~rcs46/lectures_2017/08-trees/08-tree-regression.pdf
- Taylor, J. (2020). *Regression Trees*. <https://web.stanford.edu/class/stats202/notes/Tree/Regression-trees.html>
- Torres-Barrán, A., Alonso, Á., & Dorransoro, J. R. (2017). Regression Tree Ensembles for Wind Energy and Solar Radiation Prediction. *Neurocomputing*, 326-327, 151-160. <https://doi.org/10.1016/j.neucom.2017.05.104>
- U.S. Department of Energy (2011). *2011 Buildings Energy Data Book*. <https://ieer.org/wp/wp-content/uploads/2012/03/DOE-2011-Buildings-Energy-DataBook-BEDB.pdf>
- Wang, H., Sun, J., Sun, J., & Wang, J. (2017). Using Random Forests to Select Optimal Input Variables for Short-Term Wind Speed Forecasting Models. *Energies*, 10, 1522. <https://doi.org/10.3390/en10101522>
- Wei, T., Wang, Y., & Zhu, Q. (2017). Deep Reinforcement Learning for Building HVAC Control. In *Proceedings of the 54th Annual Design Automation Conference* (Article No. 22). New York: Association for Computing Machinery. <https://doi.org/10.1145/3061639.3062224>
- Willuhn, M. (2019). *Renewable Energy Investment to Increase by \$210 Billion over Five Years*. <https://www.pv-magazine.com/2019/02/12/renewable-energy-investment-to-increase-by-210-billion-over-five-years/>
- Zhu, Z., Zhou, D., & Fan, Z. (2016). Short Term Forecast of Wind Power Generation Based on SVM with Pattern Matching. In *2016 IEEE International Energy Conference* (pp. 1-6). Leuven: IEEE.