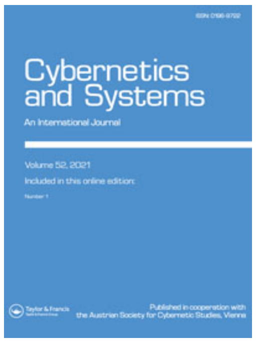

Cybernetics and Systems

An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucbs20

Investigation of Data Size Variability in Wind Speed Prediction Using AI Algorithms

M. A. Ehsan, Amir Shahirinia, Nian Zhang & Timothy Oladunni

To cite this article: M. A. Ehsan, Amir Shahirinia, Nian Zhang & Timothy Oladunni (2021) Investigation of Data Size Variability in Wind Speed Prediction Using AI Algorithms, Cybernetics and Systems, 52:1, 105-126, DOI: 10.1080/01969722.2020.1827796

To link to this article: https://doi.org/10.1080/01969722.2020.1827796

Published online: 06 Oct 2020.

CYBERNETICS AND SYSTEMS: AN INTERNATIONAL JOURNAL
2021, VOL. 52, NO. 1, 105–126
https://doi.org/10.1080/01969722.2020.1827796

Investigation of Data Size Variability in Wind Speed Prediction Using AI Algorithms

M. A. Ehsan[a], Amir Shahirinia[a], Nian Zhang[a], and Timothy Oladunni[b]

[a]Department of Electrical and Computer Engineering, University of the District of Columbia, Washington, DC, USA; [b]Department of Computer Science & Information Technology, University of the District of Columbia, Washington, DC, USA

ABSTRACT
Electricity generation from burning fossil fuel is one of the major contributors to global warming. Renewable energy sources are a viable alternative to produce electrical energy and to reduce the emission from power industry. They have unlocked opportunities for consumers to produce electricity locally and use it on-site that reduces dependency on centralized generation. Despite the widespread availability, one of the major challenges is to understand their characteristics in a more informative way. Wind energy is highly dependent on the intermittent wind speed profile. This paper proposes the prediction of wind speed that simplifies wind farm planning and feasibility study. Twelve artificial intelligence algorithms were used for wind speed prediction from collected meteorological parameters. The model performances were compared to determine the wind speed prediction accuracy and model comparison for different sizes of data set. The results show, the most effective algorithm varies based on the data size.

KEYWORDS
Convolutional neural networks; deep learning; long short-term memory (LSTM); wind speed prediction

Introduction

Climate change is one of the most challenging questions in the research and non-research communities. Global warming is marked as one of the prime reasons (Rivas and Gonzalo 2019). Electric power generation on the other hand is an integral part of current advancement while being responsible for greenhouse gas emission due to the combustion of fossil fuel. In 2018, 33% of total carbon dioxide emission came from electric power sector (U.S. EIA 2020). Therefore, dependency shift from fossil fuel to renewable energy centered power system is a key. However, among the renewable energy resources, wind has prospect that has not been harnessed in full due to its intermittent nature (variability of wind speed) and capital investment requirement (Office of Energy Efficiency & Renewable Energy 2018a, 2018b). Technological challenges are yet to overcome (Watson et al. 2019).

CONTACT Nian Zhang nzhang@udc.edu Department of Electrical and Computer Engineering, University of the District of Columbia, Washington, DC, USA.

Wind speed prediction is a regression problem where predictors, in this case, are the meteorological parameters, and the response variable is the wind speed at 80 m height. In general, regression is a classical problem both in statistics and machine learning. Usually, statistical methods are to find the inference while machine learning makes the prediction (Shmueli 2010). Then again, they intersect in some cases and do serve a similar purpose, for example- linear regressio (Matson and Huguenard 2007). However, statistical learning relies on distributions, while machine learning is an empirical process and requires data (Bzdok, Altman, and Krzywinski 2018). The statistical approach, thus, considers how data are collected or generated while machine learning may result in accurate prediction without knowing much about the underlying aspects of data. Another line of the boundary is the shape or volume of data. While the statistical approach is very robust about the number of samples (as it considers the distribution of the data), machine learning is more applicable when the dataset is wide (Bzdok, Altman, and Krzywinski 2018). However, sometimes they are used interchangeably, and statistics are the backbone of machine learning (Brownlee 2018). Besides, some machine learning algorithms use the same bootstrapping methods as statistical models. In addition, researchers are also leveraging deep learning for similar prediction problems (Yen et al. 2019). Artificial Neural network (ANN) based models usually yield benefits in prediction tasks than statistical models due to its robustness toward the nature of data, especially when there are missing values, or the dataset is not well preprocessed, raw, and large data (Chen et al. 2019). Thus, many machine learning regression algorithms use statistical techniques in innovative ways, while deep learning neural network approaches are efficient for analogous tasks. The reason, however, behind the growing popularity of machine learning or deep learning (artificial intelligence, in general) is the availability of computational resources (OpenAI 2018). Therefore, the larger dataset is not a critical issue to work with, which was challenging in the past. In this research, we have considered both approaches, machine learning, and deep learning for our wind speed prediction problem.

The placement of a wind turbine for wind power generation is often a challenging step due to the varying nature of wind speed from a location/height to another location/height (Cetinay, Kuipers, and Guven 2017). Measuring wind speed at the level of turbine hub height is both expensive and requires continuous maintenance. Meteorological parameters also play a vital role in the wind characteristics. Therefore, wind speed profiling with the variation of meteorological parameters has been a research problem that leads to the prediction of wind speed of a certain location based on those parameters (Hoolohan, Tomlin, and Cockerill 2018; Ehsan et al. 2019). Therefore, utilization of easy to access parameters in a low elevation to predict corresponding wind speed at a higher height is a practical approach. Literature show there have been statistical approaches for wind speed application, while artificial intelligence-deep learning and machine learning are being considered recently (Pearre and Swan 2018; Zheng et al. 2011). In addition to machine learning regression algorithms, neural network-based deep learning techniques are getting attention for alike problems due to higher accuracy.

Deep Neural Network (DNN) can map features from raw data to provide regularization, thus minimizes the variance in each layer (Liu et al. 2014). This capability makes DNN suitable for prediction problems. The prediction accuracy is greatly dependent on efficient feature extraction of time series data. Literature (Liu et al. 2018a) shows – Convolutional Neural Network (CNN) creates filters to automate such function that makes it widely applicable for prediction. However, it can represent short term dependence while wind speed comprises both short and long-term dependence fundamentally (Liu et al. 2018b). Therefore, long short-term memory (LSTM) seems more effective for wind speed prediction. LSTM is a form of Recurrent Neural Network (RNN) that is capable of learning long-term dependencies to make a prediction (Pei et al. 2019).

The novelty of this paper is the inclusion of a wider ranged meteorological variables to predict wind speed for a higher elevation of interest using existing regression models, which differentiates this work from previously done research works as described in the literature. The heights where the predictors are measured are also noteworthy. Average wind speed at 2 m is an easy-to-reach height in terms of sensor installation and maintenance compared to 80 m, same goes to temperature, wind direction, and turbulence measuring points' elevations. Therefore, this paper shows an approach to utilize the information already available or gathered by minimum effort in the prediction of a parameter that is complex and expensive to get.

The rest of this paper is organized as follows. Section "Methodology" describes the methodology for feature selection and prediction application. In Section "Background of Prediction Algorithms," prediction algorithms, both machine and deep learning approaches are discussed. Performance evaluation metrices are presented in Section "Performance Evaluation." In Section "Simulations and experiments," the National Renewable Energy Laboratory (NREL) data set is utilized for simulations and experimental results are demonstrated. In Section "Conclusion," the conclusions are given.

Methodology

This section describes the theoretical approach taken for this research. Figure 1 illustrates a flow chart detailing every step. The steps include preprocessing of the data, short description of the candidate prediction

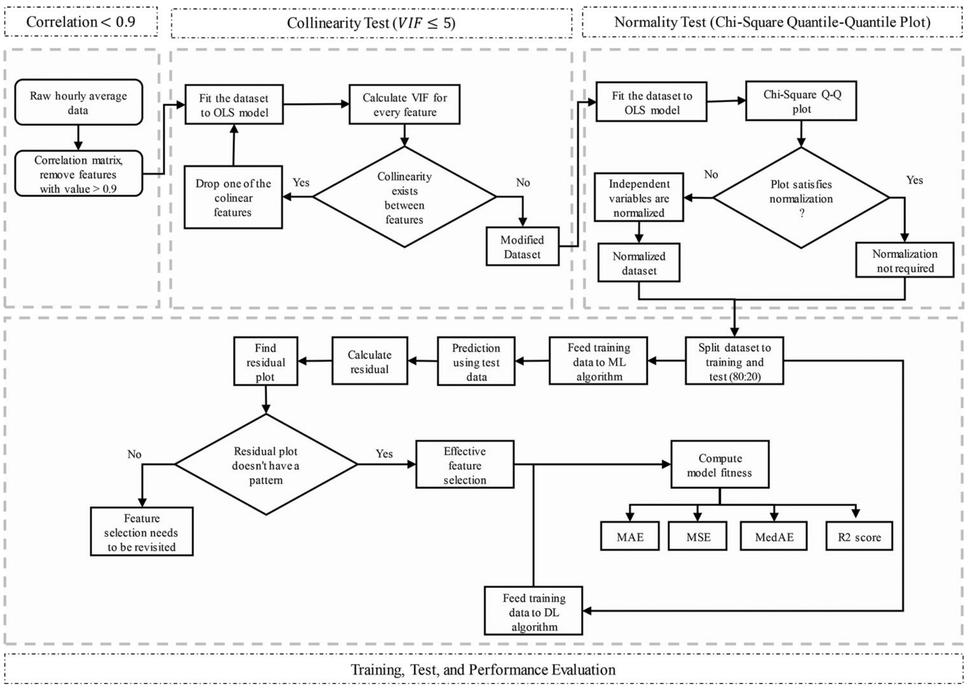
Correlation < 0.9
Collinearity Test (VIF ≤ 5)
Normality Test (Chi-Square Quantile-Quantile Plot)
Raw hourly average data
Correlation matrix, remove features with value > 0.9
Fit the dataset to OLS model
Calculate VIF for every feature
Collinearity exists between features
Drop one of the colinear features
Yes
No
Modified Dataset
Fit the dataset to OLS model
Chi-Square Q-Q plot
Plot satisfies normalization ?
Independent variables are normalized
Normalized dataset
Normalization not required
Split dataset to training and test (80:20)
Feed training data to ML algorithm
Prediction using test data
Calculate residual
Find residual plot
Residual plot doesn't have a pattern
Feature selection needs to be revisited
Effective feature selection
Compute model fitness
MAE
MSE
MedAE
R2 score
Feed training data to DL algorithm
Training, Test, and Performance Evaluation

Figure 1. Wind speed prediction approach.

algorithms, and performance evaluation metrices. These are the formal steps for any prediction modeling. Correlation and collinearity measures help to overcome overfitting in prediction. Then normality test is done to evaluate the data quality. The data is now ready for training the prediction algorithms. There are accuracy measures that are fundamental to evaluate the prediction accuracy. Following subsections contain detailed discussion of data pre-processing step. Then rest of the steps is described in Sections "Background of Prediction Algorithms" and "Performance Evaluation."

Correlation Matrix

Correlation is a measure to identify how strongly a pair of variables (X, Y) are related. This score ranges between $+1$ and -1 based on their degree of relation. *Pearson, Kendall,* and *Spearman* are common correlation techniques. We have used *Spearman* correlation (1) as it does not carry any assumption about the distribution of data (while *Pearson*'s correlation assumes data are normally distributed, which is not always the case) (Dodge 2008).

$$\rho = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{1}$$

Where, $\rho :$ correlation coefficient; $rg_X, rg_Y :$ ranks of X and Y; $cov :$ covariance; $\sigma_{rg_X}\sigma_{rg_Y}$: the standard deviation of rank variables.

The cross-correlation between the predictors (independent or explanatory variable, X) and the response (dependent variable, Y), is the initial step toward feature selection. If the absolute value of a correlation measure is greater than 0 between two parameters, then they are considered correlated. Perfect correlation is one that usually indicates self-correlation, or a problem called multicollinearity that leads to false predictions. Therefore, features with very high correlation (>0.9) are removed from the dataset as they might mislead the model performance (Vu, Muttaqi, and Agalgaonkar 2015; Alin 2010).

Collinearity Test

Collinearity is a condition when some independent variables are highly correlated. In a regression model, if there are highly correlated predictors, they cannot independently predict the value of the dependent variable. In other words, they explain some of the same variances in the dependent variable, which causes inflated values of coefficients or even reverses the sign. If more than two predictors are associated in that manner, it is called multicollinearity (Tintner 1975).

Variance Inflation Factor (VIF) is applied to investigate if collinearity exists among the features (Marcoulides and Raykov 2019). For a multiple linear regression model with n predictors, VIFs are the diagonal elements of the inverse of the correlation matrix of the predictors. VIF for the nth predictor variable is expressed as (2).

$$VIF_n = \frac{1}{1 - {R_n}^2} \quad n = 1, 2, 3, \ \ldots \tag{2}$$

The smallest possible VIF is one representing not collinear at all and higher the value greater the collinearity. However, higher values of VIF (>5) represent the presence of multicollinearity, causing redundancy (more than one predictor describing the same effect on the response variable) (Marcoulides and Raykov 2019). Therefore, one from each collinear subset of predictors is kept, and others are removed from data (James et al. 2013; Bruce and Bruce 2017).

Normality Test

Normality tests are applied to investigate if the dataset is well modeled (likelihood of data to be normally distributed). Some approaches to test for normalization are- graphical method, back-of-the-envelope test, frequentist test, and Bayesian test (Ghasemi and Zahediasl 2012). In this research, we use the graphical test. In this method, the Chi-Square Quantile-Quantile (Q-Q) plot of multivariate distribution is analyzed to see if the features are normally distributed (Liang, Pan, and Yang 2004). If normal, the plot should follow the 45-degree baseline. If not, then normalization is required before fitting the data to any model. Thus, prior normalization is required before feeding it to the models.

Background of Prediction Algorithms

Machine Learning Approach

Machine learning regression algorithms are widely used for prediction applications while they vary in terms of behind the scene mathematics (Stulp and Sigaud 2015). The following algorithms are used in this research.

Multiple Linear Regression

Multiple linear regression relates two or more dependent variables to an independent variable by fitting a linear equation to the data (Hill and Gelman 2007; Uyanık and Güler 2013; Mehryar Mohri and Talwalka 2012). For n independent variables $(x_1, x_2, \ldots, x_n \in \mathbb{R})$ and m observations, the multiple linear regression model (3) is defined as-

$$\widehat{y_i} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_n x_{in} + \varepsilon_i \quad (i = 1, 2, \ldots, m) \tag{3}$$

Where, $y_i \in \mathbb{R}$: dependent variable, $\hat{y}_i$ is the estimate of y_i; ε : residual (deviation of $\hat{y}_i$ from its mean value); β : regressor coefficients estimated from least-square estimates (β_0 is known as intercept, and β_n is the slope of the regression line for simple linear regression); m : number of rows in the dataset.

The sum of squares of residuals ($SS_{residuals}$) of multiple linear regression is shown in (4), and we would like to minimize this value in the model. The equation for multiple linear regression can also be written as (5), where *argmin* stands for the argument of minimum. In other words, it finds the β that minimize $SS_{residuals}$. β is easy to find from (6), which is also known as the matrix formulation of linear regression, where X and Y represent independent and dependent variables, respectively.

$$SS_{residuals} = \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 \tag{4}$$

$$\hat{\beta}^{MLR} = \underset{\beta \epsilon \mathbb{R}}{argmin} \sum_{i=1}^{n} [y_i - \widehat{y_i}]^2 = \underset{\beta \epsilon \mathbb{R}}{argmin} \sum_{i=1}^{n} [y_i - (\beta_o + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in})]^2 \tag{5}$$

$$\hat{\beta}^{MLR} = (X^T X)^{-1} X^T Y \tag{6}$$

Ridge Regression

Ridge regression is a variant of linear regression that uses L2 regularization (Exterkate et al. 2016). The fundamental equation for ridge regression is the same as linear regression with a constraint on it, as described in (7), where C defines the boundaries of ridge regression. The regularization shrinks the parameters and reduces the model complexity by a coefficient of shrinkage. This coefficient is known as the penalty and denoted by λ (hyperparameter). Now we look at the equation for ridge regression and it is clear- first part of (8) is the same as linear regression, and the true difference between linear and ridge is the second term containing the constraint, B, shown in (9) and penalty. Due to the inclusion of the penalty, the residual error is minimized, therefore, ridge regression shall produce better accuracy.

$$\beta_0{}^2 + \beta_1{}^2 + \cdots + \beta_n{}^2 \leq C^2 \tag{7}$$

$$\hat{\beta}^{ridge} = \underset{\beta\epsilon\mathbb{R}}{argmin} \sum_{i=1}^{n} [y_i - \hat{y}_i]^2 = \underset{\beta\epsilon\mathbb{R}}{argmin}\, min(||y_i - XB||_2^2 + \lambda||B||_2^2) \tag{8}$$

$$||B||_2 = \sqrt{\beta_0{}^2 + \beta_1{}^2 + \cdots + \beta_n{}^2} \tag{9}$$

Least Absolute Shrinkage and Selection Operator (Lasso) Regression

Lasso regression uses L1 regularization. It has benefits over ridge regression when there are more features (Illarionov and Khudorozhkov 2018). The equation of lasso regression (10) is quite straight forward from the ridge regression, the only difference is the second term of the equation.

$$\hat{\beta}^{lasso} = \underset{\beta\epsilon\mathbb{R}}{argmin} \sum_{i=1}^{n} [y_i - \hat{y}_i]^2 = \underset{\beta\epsilon\mathbb{R}}{argmin}\, min(||y_i - XB||_2^2 + \lambda||\beta||_1) \tag{10}$$

Adding regularization is a very important technique in machine learning to prevent overfitting (Giarré and Argenti 2018). The difference between the L1 and L2: L2 is the sum of the square of the weights, while L1 is just the sum of the weights.

Bayesian Ridge Regression

Bayesian view of ridge regression is obtained by noting that the minimizer of (8) can be considered as the posterior mean of a model where $\beta_i \sim N\left(0, \frac{\sigma^2}{\lambda}\right)$, for $i = 1, 2, \ldots, n$ (Shi, Abdel-Aty, and Lee 2016).

Huber Regression

In the ridge and lasso, the penalty was a hyperparameter. Instead of considering an estimated constant, Huber function (11) is proposed. This method does not rely on SS$_{\text{residuals}}$ to minimize the error, rather sensitive to outliers $(|y_i - \widehat{y}_i| \leq \delta)$. Thus, the name, Huber regression (Sun, Zhou, and Fan 2020), came from the loss function is uses. δ is a hyperparameter, and the choice is critical. Residuals larger than δ are minimized with L1 (which is less sensitive to large outliers), while residuals smaller than δ are minimized appropriately with L2.

$$L_\delta\left(y_i, \widehat{y}_i\right) = \begin{cases} \frac{1}{2}(y_i - \widehat{y}_i)^2 \ for \ |y_i - \widehat{y}_i| \leq \delta \\ \delta|y_i - \widehat{y}_i| - \frac{1}{2}\delta^2 \ otherwise \end{cases} \tag{11}$$

Bagging Regression

Bootstrap aggregating (bagging) prediction models is a general method for fitting multiple versions of a prediction model and then combining them into an aggregated prediction (Breiman 1996; Sutton 2005). It is a straightforward algorithm in which b bootstrap copies of the original training data are created, the regression is applied to each bootstrap sample, and in the regression context, new predictions are made by averaging the predictions together from the individual base learners. Equation (12) thus demonstrates the formulation by letting x_i as the prior and $\hat{y}_{bag}$ as the bagged prediction. $\hat{y}_{i1}, \ \hat{y}_{i2}, \ldots, \hat{y}_{ib}$ are the the predictions from individual base learners for x_i.

$$\hat{y}_{bag} = \hat{y}_{i1} + \hat{y}_{i2} + \cdots + \hat{y}_{ib} \tag{12}$$

Random Forest Regression

Random forest regression is also a bootstrap aggregation (bagging) that involves training each decision tree on a different data sample where sampling is done with replacement (Sutton 2005; Pal 2017). Thus, it combines multiple decision trees in determining the model output. A visual illustration of such a model is in Figure 2.

Adaptive Boosting (AdaBoost) Regression

AdaBoost is stagewise estimation procedures to fit an additive logistic regression model, which minimizes the exponential loss function (Cao et al. 2013). The operational steps of this algorithm start with initializing the observation weights β_{i} on the original sample (predictors). The classifier then adjusts the weights for residual minimization and re-iterate the process for a defined number of times. Finally, the model becomes an

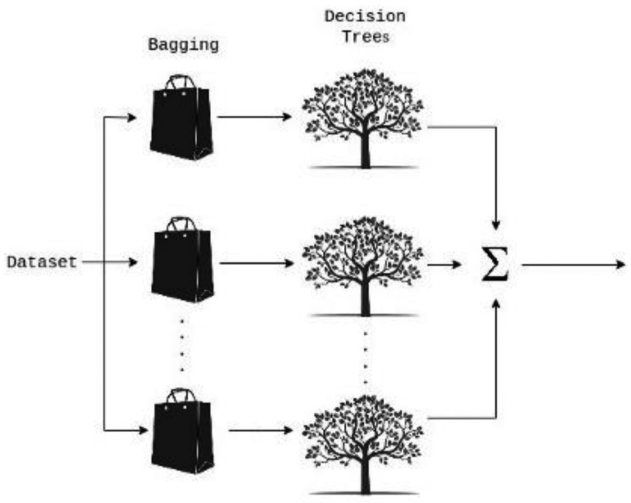

Figure 2. Visual aid operation of random forest regression.

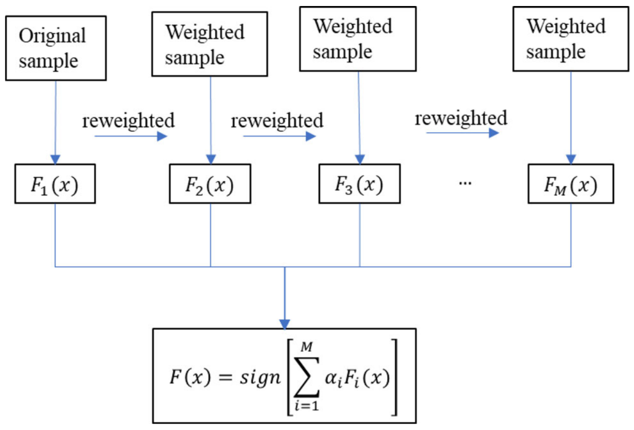

Figure 3. Learning steps in AdaBoost regression.

aggregation of classifier function, $F_i(x)$, and weight minimizer, α_i, as shown in Figure 3 (Varmuza 2003).

Support Vector Regression (SVR)

Support vector regression is a robust algorithm that can handle both linear and non-linear input regression problems (Smola and Schölkopf 2004). We consider a linear case, where predictors, x and response $\widehat{y}_i$. The formulation for a linear case is shown in (13) where $f_i(x)$ is the transfer function (also known as the kernel), and b is bias. A few of the common transfer functions are linear, non-linear, polynomial, and radial basis functions.

$$\hat{y}_i = \sum_{i=1}^{n} \beta_i f_i(x) + b \tag{13}$$

Deep Learning Approach

Deep learning algorithms are now applied to solve problems of a diverse nature, including prediction (Filik and Filik 2017). Therefore, we are considering deep learning algorithms for this research. Firstly, we would like to review a few basics of deep learning. The building blocks of deep learning or artificial neural networks are called perceptron, which mimics an equivalent functionality (in computation) as neuron (a biological cell of the nervous system that uniquely communicates with each other) (Goodfellow 2016).

Now, perceptron or artificial neurons receive input signals $(x_1, x_2, \ldots, x_m)$, multiply input by weight $(w_1, w_2, \ldots, w_m)$, add them together with a predetermined bias, and pass through the activation function, $f(x)$. The signal goes to output as *0* or *1* based on the activation function threshold value. A perceptron with inputs, weights, summation and bias, activation function, and output all together forms a single layer perceptron (Kanal 2001). However, in common neural network diagrams, only input and output layers are shown. In a practical neural network, hidden layers are added between the input and output layers. The number of hidden layers is a hyperparameter and usually determined by evaluating the model performance. If the neural network has a single hidden layer, the model is called a shallow neural network, while a DNN consists of several hidden layers (Goodfellow 2016). In this research, we have considered DNN, CNN, and RNN- in the form of LSTM, all of which will be discussed in the following sections.

Deep Neural Network (DNN)

DNN is composed of three neural network layers, namely- an input layer, hidden layer(s), and an output layer. The (number of hidden layers) is tuned through trial and error (Goodfellow 2016). Figure 4 illustrates such a model structure with two hidden layers consisting of three neurons each, five input neurons, and one output neuron. The number of neurons depends on the number of input and output.

In Figure 4, [x_1, x_2, x_3, x_4, x_5] is input; h represents hidden layer weights, and $\hat{y}$ is output.

A simplified DNN kernel is formulated in (14) that considers linear modeling. x, W, and c symbolize input, weights, and bias, respectively, while w and b are linear model parameters. The hidden layer parameter h is shown in (15), where g is the activation function. For DNN modeling,

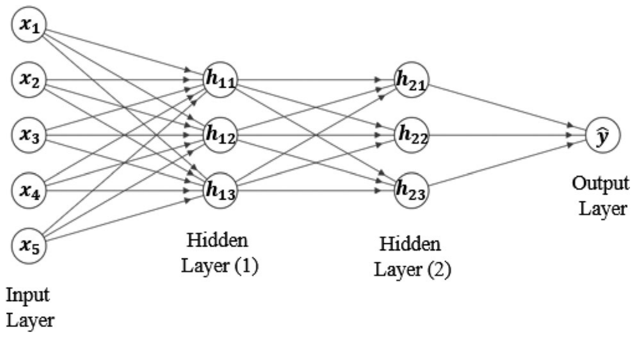

Figure 4. Simplified architecture of a DNN.

ReLu (16) is used as the hidden layer activation function.

$$f(x; W,\ c,\ w, b) = \ w^T max\{0,\ W^T + c\} + b \tag{14}$$

$$h = \ g(W^T x + c) \tag{15}$$

$$f(x) = max(0, x) \tag{16}$$

Convolutional Neural Network (CNN)

CNN, also known as ConvNet, is one way to solve the issue with DNN using convolution rather than matrix multiplication (Gulli 2017). In other words, CNN is the regularized version of DNN to ensure model robustness toward overfitting. CNN is very popular for image processing; however, in the prediction problem, it is also utilized (Kiranyaz 2019). In this research, we are using 1 D CNN for wind speed prediction. The characteristics and approaches are the same for all CNNs, regardless of dimensionality (Zhao et al. 2018). The architecture of CNN (Figure 5 shows for 1 D CNN) consists of a convolution layer, pooling layer, and a fully connected neural network layer, thus, incorporates local receptive fields to ponder the spatial information, shared weights, and pooling to consider the summary statistics in the output.

Recurrent Neural Network (RNN) – LSTM

Long Short-Term Memory Networks (LSTM), a form of gated RNN, is proposed to implement. LSTM introduces self-loops to produce paths where the gradient can flow for a long duration; thus, it is capable of learning long-term dependencies (Goodfellow 2016). LSTMs are explicitly designed to avoid the long-term dependency problem, as illustrated in Figure 6. The equations describing the operations are listed below.

$$f(t) = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{17}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{18}$$

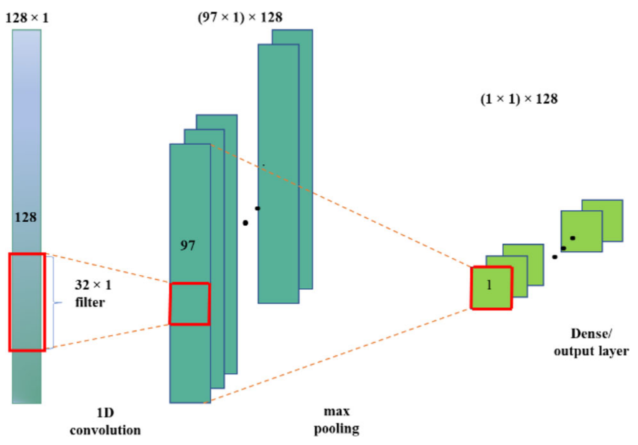

Figure 5. Architecture of 1 D convolution neural network.

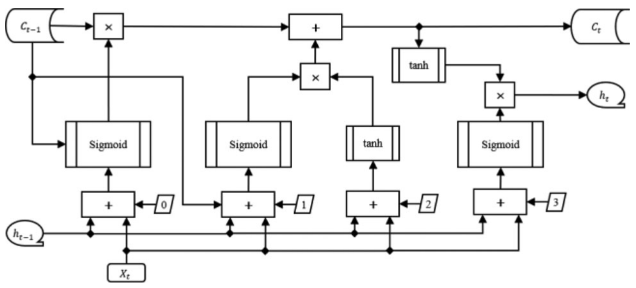

Figure 6. Block diagram of LSTM operations.

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{19}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{20}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{21}$$

Where, $x_t \epsilon \mathbb{R}^d$: Input vector to the LSTM unit; $f_t \epsilon \mathbb{R}^h$: Forget states activation vector; $i_t \epsilon \mathbb{R}^h$: Input/update gate's activation vector; $o_t \epsilon \mathbb{R}^h$: Output gate's activation vector; $h_t \epsilon \mathbb{R}^h$: Hidden state vector; $c_t \epsilon \mathbb{R}^h$: Cell state vector; $W \epsilon \mathbb{R}^{h \times d}$, $U \epsilon \mathbb{R}^{h \times h}$, $b \epsilon \mathbb{R}^h$: Weight matrices and bias vector parameters which need to be learned during the training; σ_g : Sigmoid function; σ_c, σ_g : hyperbolic tangent function.

Performance Evaluation

Some commonly used accuracy parameters are employed to evaluate how well a model is performing to predict the intended parameter (Joshi 2016). Mean absolute error (MAE), mean square error (MSE), median absolute error (MedAE), and R-square (R2) scores are considered to investigate the model performances on the test set.

MAE is the average of the absolute values of the error (the difference between actual response (y_i) and predicted response ($\widehat{y_i}$). As described by (22), n is the number of total input sets. The lower this value is, the better the model performance, while the desired is *0*.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \widehat{y_i}|}{n} \tag{22}$$

MSE is the mean of the square of error terms. Similarly, to MAE, it is desired to have *0* or close value for this term. The formula for this measure is in (23).

$$MSE = \frac{\sum_{i=1}^{n} \left(y_i - \widehat{y_i}\right)^2}{n} \tag{23}$$

MedAE is the median of all the error terms, defined in (24), thus effective to deal with outliers' effect in the model performance.

$$MedAE = median\ \left(|y_1 - \widehat{y_1}|,\ |y_2 - \widehat{y_2}|,\ |y_3 - \widehat{y_3}|,\ \ldots\ldots,\ |y_n - \widehat{y_n}|\right) \tag{24}$$

R2 score determines how well the model would perform in predicting the response variable as shown in (25) where $\overline{y_i}$ denotes the mean value of all predictions. This value is also known as the coefficients of determination. The best possible value is *1* for this case, and the closer to *1*, the better model prediction is.

$$R2\ score = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \widehat{y_i}\right)^2}{\sum_{i=1}^{n} \left(y_i - \overline{y_i}\right)^2} \tag{25}$$

Further model fitting is tested using the residual plot by graphing residual (the difference between prediction and actual value) vs. fitted instance.

Simulations and Experiments

Dataset

We collected data from the National Renewable Energy Laboratory (NREL) database available online (Jager and Andreas 1996). The dataset contains three-months hourly samples (May 1 to July 31, 2018) and is used as the reference of comparison here. This paper considers four additional data sets of different sizes: 10 days (March 1–10, 2019), 30 days

Table 1. Cross-correlation of model parameters.

	Parameters [unit]	1	2	3	4	5	6	7	8
1	Temperature (2 m) [°C]	1.0	−0.02	0.24	−0.22	0.12	0.42	0.05	0.16
2	Sea-level pressure (Est) [mBar]	−0.02	1.0	−0.13	−0.05	−0.09	0.03	−0.07	−0.13
3	Avg wind speed (2 m) [m/s]	0.24	−0.13	1.0	0.08	0.64	0.07	0.23	0.87
4	Avg wind direction (2 m) [°]	−0.22	−0.05	0.08	1.0	0.26	−0.52	0.19	0.25
5	Avg wind shear [1/s]	0.12	−0.09	0.64	0.26	1.0	−0.07	0.52	0.88
6	Turbulence intensity (2 m)	0.42	0.03	0.07	−0.52	−0.07	1.0	−0.01	−0.03
7	Friction velocity (u*) [m/s]	0.05	−0.07	0.23	0.19	0.52	−0.01	1.0	0.41
8	Avg wind speed (80 m) [m/s]	0.16	−0.13	0.87	0.25	0.88	−0.03	0.41	1.0

(January 1–30, 2019), 6 months (January 1 to June 30, 2010), and 1 year (January 1 to December 31, 2015). The raw data entail samples of each minute. It was converted to average hourly instances. Primarily, the dataset had 18 features, among which wind speed in 80 m height is our response variable, and other 17 are predictors- solar radiation [listed as global PSP (Precision Spectral Pyranometer)], temperature (2 m), estimated sea-level pressure, average wind speed (2 m), average wind direction (2 m), average wind shear, turbulence intensity, friction velocity, wind chill temperature, dew point temperature, relative humidity, specific humidity, station pressure, average wind speed (5 m), accumulated precipitation, atmospheric electric field, and estimated surface roughness. Instances inside "()" represents the height where the parameter was measured, "m" stands for meters.

After preprocessing the data following steps from Section "Methodology," two colinear pairs were found and treated: temperature (2 m) and wind chill temperature, and average wind speeds (2 m and 5 m). Then wind chill temperature and average wind speed (5 m) are removed from the dataset, and VIF applied again to make sure the absence of collinearity. After considering correlation, and collinearity, the selected best features are- temperature (2 m), estimated sea-level pressure, average wind speed (2 m), average wind direction (2 m), average wind shear, turbulence intensity, and friction velocity. The cross-correlations between each model parameters are listed in Table 1, where parameter numbered as "8" is our response, average wind speed at 80 m height. The cross-correlations are also illustrated as a heatmap in Figure 7, lighter being higher and darker being lower correlations. Average wind speed at 2 m height and average wind shear both are highly correlated (+ve) with the response variable. On the other hand, sea-level pressure and turbulence intensity are negatively correlated with the wind speed at 80 m.

Figure 8 shows the Q-Q plot that tells the dataset is not normalized. Thus, prior normalization is required before feeding it to the models.

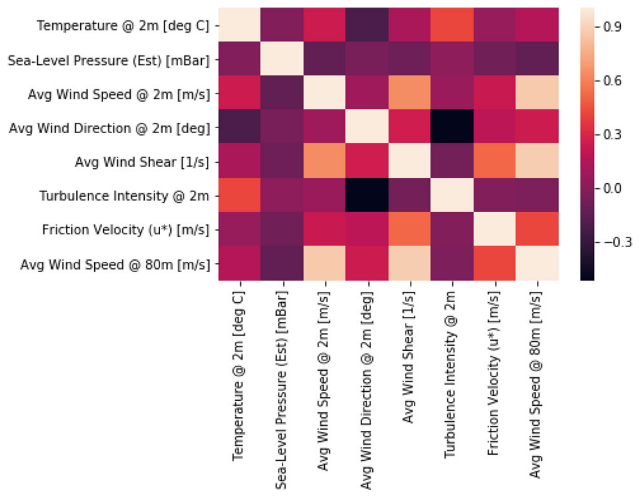

Figure 7. Heatmap showing model parameter cross-correlation.

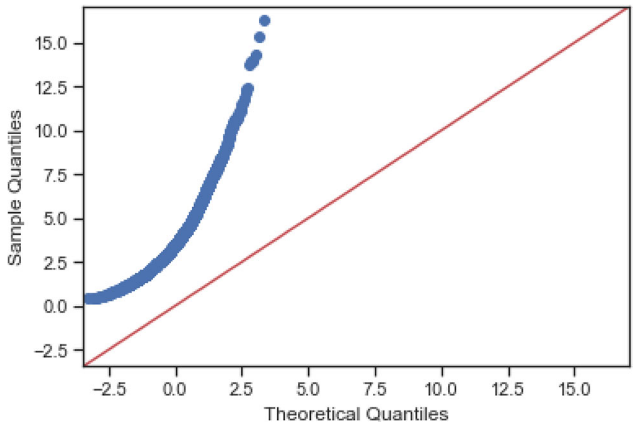

Figure 8. Chi-squared Q-Q plot.

Train-Test Split

The prediction algorithms are trained using training data. However, the performance of a model depends on how well it can predict the response variable when encounters unknown predictors (test data). Therefore, the dataset is usually divided into two sets: training and test sets. The training data set is then used to train the prediction algorithm while the test set is allocated to use them as an unknown predictor to analyze the model performance. The ratio of allocating data for training and test is randomly selected, but literature shows 70–80% for training, and 20–30% for the test is common practice (Illarionov and Khudorozhkov 2018; Ibrahim and Bennett 2014; Shobha and Rangaswamy 2018). In this research, we have separated 80% of the total data to train the models and rest 20% to test the model performance.

Table 2. Comparative model performances (duration: 3 months).

Model	Algorithm	Mean absolute error (MAE)	Mean squared error (MSE)	Median absolute error (MedAE)	R2 score
Model-1	Multiple linear regression	0.421	0.357	0.277	0.923
Model-2	Ridge regression (alpha = 15)	0.579	0.598	0.434	0.872
Model-3	Least absolute shrinkage and selection operator (Lasso) regression (alpha = 0.1)	0.823	1.156	0.704	0.752
Model-4	Bayesian ridge regression	0.428	0.361	0.285	0.922
Model-5	Hubber regression	0.422	0.38	0.259	0.919
Model-6	Bagging regression	0.274	0.171	0.185	0.963
Model-7	Random forest regression	0.275	0.179	0.192	0.962
Model-8	Adaptive boosting (AdaBoost) regression	0.385	0.272	0.297	0.942
Model-9	Support vector regression (SVR)	0.411	0.347	0.261	0.926
Model-10	Multilayer perceptron (MLP)/ DNN (hidden layer = 13, activation = relu)	0.31	0.178	0.234	0.962
Model-11	CNN (filters = 64, kernel size = 2, activation = relu, maxpooling size = 2)	0.634	0.831	0.45	0.82
Model-12	RNN – LSTM (kernel = normal, activation = linear)	0.226	0.107	0.145	0.978

Simulation Results

We will discuss the simulation and performances of the state-of-the-art prediction algorithms for wind speed prediction in 80 m height for the NREL dataset. We listed the algorithms as Model-1 to 12 in Table 2 and fitted them on the training data for learning. Once the training is done, we evaluated model performances according to the accuracy measures described in Section "Performance Evaluation" on test data. Following settings are chosen for the reference data set and then used in a generalized manner for the four different test cases.

For ridge regression, alpha was considered 15 after a few trial and errors. Similarly, for Lasso, the alpha parameter was set to 0.1. For SVR, the default kernel initializer was applied. Table 2 depicts the accuracy measures for each algorithm. Overall, the considered algorithms were able to predict the average wind speed properly for the refence data with an R2 value greater than 0.9 in most cases, as shown. Among the machine learning algorithms, MAE, MSE, and MedAE are minimum for bagging and random forest regression. Both algorithms show greater accuracy ($>96\%$). On the other hand, Models 3–5 show the lowest accuracy among the machine learning regression algorithms with an R2 Score ≈ 0.92.

Deep learning models- DNN, CNN, and LSTM, are denoted as Models 10–12 in Table 2. Both DNN and CNN use the ReLu activation function. DNN uses 13 hidden layers, while the neural network of CNN consists of

Table 3. Comparative model performances (duration: 10 days, 30 days, 6 months, 1 year).

	Mean absolute error (MAE)				Mean squared error (MSE)				Median absolute error (MedAE)				R2 score			
Models	10 days	30 days	6 months	1 year	10 days	30 days	6 months	1 year	10 days	30 days	6 months	1 year	10 days	30 days	6 months	1 year
1	0.419	0.475	0.637	1.913	0.332	0.411	0.721	7.021	0.296	0.37	0.489	1.509	0.976	0.979	0.922	0.296
2	1.365	1.152	0.835	1.931	3.917	2.596	1.128	7.243	1.055	0.848	0.696	1.533	0.717	0.87	0.878	0.273
3	0.681	0.72	0.996	2.153	0.753	0.86	1.53	8.817	0.566	0.609	0.895	1.736	0.946	0.957	0.835	0.115
4	0.419	0.475	0.637	1.914	0.332	0.411	0.721	7.031	0.298	0.371	0.489	1.509	0.976	0.979	0.922	0.295
5	0.411	0.476	0.629	1.355	0.343	0.418	0.731	12.302	0.288	0.353	0.462	0.922	0.975	0.979	0.921	−0.23
6	0.401	0.422	0.282	0.299	0.293	0.372	0.16	0.199	0.297	0.296	0.203	0.205	0.979	0.981	0.983	0.98
7	0.429	0.441	0.278	0.295	0.3	0.452	0.156	0.198	0.374	0.313	0.197	0.2	0.978	0.977	0.983	0.98
8	0.503	0.584	0.497	0.629	0.426	0.56	0.391	0.632	0.414	0.44	0.405	0.542	0.969	0.972	0.958	0.937
9	0.693	0.489	0.63	1.832	0.81	0.436	0.742	7.626	0.468	0.391	0.458	1.305	0.942	0.978	0.92	0.235
10	2.694	0.469	0.616	0.290	15.266	0.415	0.696	0.190	2.18	0.332	0.454	0.2	−0.10	0.979	0.925	0.98
11	0.349	0.414	0.617	0.620	0.196	0.304	0.677	0.63	0.311	0.311	o.45	0.54	0.986	0.985	0.927	0.93
12	0.481	0.396	0.572	1.830	0.344	0.314	0.614	0.720	0.413	0.305	0.427	1.3	0.975	0.984	0.934	0.23

50 neural network layers. Max pooling size for CNN is 2. On the other hand, LSTM uses a linear activation and consists of 50 hidden layers. In terms of accuracy and error parameters, CNN showed the worst performance, while both DNN and LSTM prediction accuracy were high (>96%). However, LSTM (Model-12) showed the best performance in terms of all metrics with the lowest error terms, while the exact accuracy was 97.8%. Overall, we can see from Table 2, plots, and discussion, LSTM performed best for our investigation. Therefore, LSTM is the efficient learning algorithm between 12 test models to predict the wind speed at 80 m height while the temperature at 2 m height, estimated sea-level pressure average wind speed at *2* m height, average wind direction at 2 m height, average wind shear, turbulence intensity at 2 m height, and friction velocity of a certain location are known. Now we shall discuss the model performances for other four test cases and compare them with the reference experiment.

In Table 3, for the 10 days dataset, Bagging regression was able to predict the wind speed with the highest accuracy of 97.9% and CNN with 98.6%. Similar trend is visible for 30 days data set, Bagging regression, CNN and LSTM all were able to predict wind speed with accuracies greater than 98%. For a larger dataset, 6 months, Bagging and random forest regressions outperformed others with >98% accuracy, while deep learning models are comparatively less effective (LSTM with 93.4%). For the largest dataset (1 year), a similar pattern is evident for both Bagging and random forest regressions with 98% accuracy while DNN performs similar.

In a nutshell, LSTM outperforms other models for three months wind speed data, while it does not hold ground truth for different data sizes. We have analyzed, Bagging regression outperforms other machine learning models for all cases with accuracy measures with accuracies >96%. However, the deep learning models perform differently. Overall, most of the algorithms (Model 1–12) were able to predict wind speed with accuracies greater than 90%.

Conclusion

Wind speed prediction is crucial for understanding the inherent nature of wind energy. Its variation is intricate and the influential weather factors are complex to infer. In this paper, we predicted wind speed at a height that is challenging to reach by using easy to access weather parameters- few of which haven't been considered before in analogous research. We investigated twelve artificial intelligence algorithms for four different data sizes and compared them to a reference case. This research will be useful for wind farm planning and feasibility study.

Funding

This work was supported by the National Science Foundation (NSF) grants #1900462, #1505509 and #2011927, DoD grants #W911NF1810475 and #W911NF2010274, and the NIH grant #1R25AG067896-01.

References

Alin, A. 2010. Multicollinearity. *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (3):370–4. doi:10.1002/wics.84.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24 (2):123–40. doi:10.1023/A:1018054314350.

Brownlee, J. 2018. *Statistical methods for machine learning: Discover how to transform data into knowledge with Python*. Machine Learning Mastery.

Bruce, P. C., and A. Bruce. 2017. *Practical statistics for data scientists: 50 Essential concepts*. 1st ed. Sebastopol, CA: O'Reilly Media, Inc.

Bzdok, D., N. Altman, and M. Krzywinski. 2018. Statistics versus machine learning. *Nature Methods* 15 (4):233–4. doi:10.1038/nmeth.4642.

Cao, Y., Q.-G. Miao, J.-C. Liu, and L. Gao. 2013. Advance and prospects of AdaBoost algorithm. *Acta Automatica Sinica* 39 (6):745–58. doi:10.1016/S1874-1029(13)60052-X.

Cetinay, H., F. A. Kuipers, and A. N. Guven. 2017. Optimal siting and sizing of wind farms. *Renewable Energy* 101:51–8. doi:10.1016/j.renene.2016.08.008.

Chen, D., S. Liu, P. Kingsbury, S. Sohn, C. B. Storlie, E. B. Habermann, J. M. Naessens, D. W. Larson, and H. Liu. 2019. Deep learning and alternative learning strategies for retrospective real-world clinical data. *NPJ Digital Medicine* 2 (1):43. doi:10.1038/s41746-019-0122-0.

Dodge, Y. 2008. Spearman rank correlation coefficient. In *The concise encyclopedia of statistics*, 502–5. New York, NY: Springer New York.

Ehsan, B. M. A., F. Begum, S. J. Ilham, and R. S. Khan. 2019. Advanced wind speed prediction using convective weather variables through machine learning application. *Applied Computing and Geosciences* 1:100002. DOI: 10.1016/J.ACAGS.2019.100002.

Exterkate, P., P. J. F. Groenen, C. Heij, and D. van Dijk. 2016. Nonlinear forecasting with many predictors using kernel ridge regression. *International Journal of Forecasting* 32 (3):736–53. doi:10.1016/j.ijforecast.2015.11.017.

Filik, Ü. B., and T. Filik. 2017. Wind speed prediction using artificial neural networks based on multiple local measurements in Eskisehir. *Energy Procedia* 107:264–9. doi:10.1016/j.egypro.2016.12.147.

Ghasemi, A., and S. Zahediasl. 2012. Normality tests for statistical analysis: A guide for non-statisticians. *International Journal of Endocrinology and Metabolism* 10 (2):486–9. doi:10.5812/ijem.3505.

Giarré, L., and F. Argenti. 2018. Mixed $\ell 2$ and $\ell 1$-norm regularization for adaptive detrending with ARMA modeling. *Journal of the Franklin Institute* 355 (3):1493–511. doi:10.1016/j.jfranklin.2017.12.009.

Goodfellow, I. 2016. *Deep learning*. Cambridge, MA: MIT Press.

Gulli, A. 2017. *Deep learning with Keras*. 1st ed. Birmingham: Packt Publishing.

Hill, J., and A. Gelman. 2007. *Data analysis usinf regression and multilevel/hierarchical models*. New York, NY: Cambridge University Press.

Hoolohan, V., A. S. Tomlin, and T. Cockerill. 2018. Improved near surface wind speed predictions using Gaussian process regression combined with numerical weather predictions and observed meteorological data. *Renewable Energy* 126:1043–54. doi:10.1016/j.renene.2018.04.019.

Ibrahim, A. M., and B. Bennett. 2014. The assessment of machine learning model performance for predicting alluvial deposits distribution. *Procedia Computer Science* 36:637–42. doi:10.1016/j.procs.2014.09.067.

Illarionov, E., and R. Khudorozhkov. 2018. Not quite unreasonable effectiveness of machine learning algorithms. https://arxiv.org/abs/1804.02543.

Jager, D., and A. Andreas. 1996. NREL National Wind Technology Center (NWTC): M2 Tower; Boulder, CO (Data); NREL Report No. DA-5500-56489. doi:10.5439/1052222.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. Statistical learning. In *An introduction to statistical learning: With applications*, ed. R. G. James, D. Witten, T. Hastie, and R. Tibshirani, 15–57. New York, NY: Springer New York.

Joshi, P. 2016. *Python: Real world machine learning*. Birmingham: Packt Publishing.

Kanal, L. N. 2001. Perceptrons. In *International encyclopedia of the social & behavioral sciences*, ed. N. J. Smelser and P. B. Baltes, 11218–21. Oxford: Pergamon.

Kiranyaz, S. 2019. 1D convolutional neural networks and applications: A survey. https://arxiv.org/abs/1905.03554.

Liang, J., W. S. Y. Pan, and Z.-H. Yang. 2004. Characterization-based Q–Q plots for testing multinormality. *Statistics & Probability Letters* 70 (3):183–90. doi:10.1016/J.SPL.2004.10.002.

Liu, H., X. Mi, and Y. Li. 2018a. Smart deep learning-based wind speed prediction model using wavelet packet decomposition, convolutional neural network, and convolutional long short term memory network. *Energy Conversion and Management* 166:120–31. doi:10.1016/j.enconman.2018.04.021.

Liu, H., X. Mi, and Y. Li. 2018b. Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. *Energy Conversion and Management* 159:54–64. doi:10.1016/j.enconman.2018.01.010.

Liu, J. N. K., Y. Hu, J. J. You, and P. W. Chan. 2014. Deep neural network-based feature representation for weather forecasting. In Proceedings of International Conference on Artificial Intelligence (ICAI), Las Vegas.

Marcoulides, K. M., and T. Raykov. 2019. Evaluation of variance inflation factors in regression models using latent variable modeling methods. *Educational and Psychological Measurement* 79 (5):874–82. doi:10.1177/0013164418817803.

Matson, J. E., and B. R. Huguenard. 2007. Evaluating aptness of a regression model. *Journal of Statistics Education* 15 (2). doi:10.1080/10691898.2007.11889469.

Mehryar Mohri, A. R., and A. Talwalka. 2012. *Foundations of machine learning*. Cambridge, MA: The MIT Press.

Office of Energy Efficiency & Renewable Energy. 2018a. Wind Technologies Market Report. Washington, DC: U.S. Department of Energy.

Office of Energy Efficiency & Renewable Energy. 2018b. Offshore Wind Technologies Market Report. Washington, DC: U.S. Department of Energy.

OpenAI. 2018. AI and compute. *OpenAI*. Accessed December 14, 2019. https://openai.com/blog/ai-and-compute/.

Pal, R. 2017. Chapter 6 - Overview of predictive modeling based on genomic characterizations. In *Predictive modeling of drug sensitivity*, ed. R. Pal, 121–48. London: Academic Press.

Pearre, N. S., and L. G. Swan. 2018. Statistical approach for improved wind speed forecasting for wind power production. *Sustainable Energy Technologies and Assessments* 27: 180–91. doi:10.1016/j.seta.2018.04.010.

Pei, S., H. Qin, Z. Zhang, L. Yao, Y. Wang, C. Wang, Y. Liu, Z. Jiang, J. Zhou, T. Yi, et al. 2019. Wind speed prediction method based on empirical wavelet transform and new cell update long short-term memory network. *Energy Conversion and Management* 196: 779–92. doi:10.1016/j.enconman.2019.06.041.

Rivas, M. D. G., and J. Gonzalo. 2019. Trends in distributional characteristics: Existence of global warming. *Journal of Econometrics* 214 (1):153–74.

Shi, Q., M. Abdel-Aty, and J. Lee. 2016. A Bayesian ridge regression analysis of congestion's impact on urban expressway safety. *Accident; Analysis and Prevention* 88:124–37. doi:10.1016/j.aap.2015.12.001.

Shmueli, G. 2010. To explain or to predict? *Statistical Science* 25 (3):289–310. doi:10.1214/10-STS330.

Shobha, G., and S. Rangaswamy. 2018. Chapter 8 - Machine learning. In *Handbook of statistics*, eds. Venkat N. Gudivada and C. R. Rao, vol. 38, 197–228. Amsterdam: ScienceDirect. doi:10.1016/bs.host.2018.07.004.

Smola, A. J., and B. Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing* 14 (3):199–222. doi:10.1023/B:STCO.0000035301.49549.88.

Stulp, F., and O. Sigaud. 2015. Many regression algorithms, one unified model: A review. *Neural Networks: The Official Journal of the International Neural Network Society* 69: 60–79. doi:10.1016/j.neunet.2015.05.005.

Sun, Q., W.-X. Zhou, and J. Fan. 2020. Adaptive Huber regression. *Journal of the American Statistical Association* 115 (529):254–24. doi:10.1080/01621459.2018.1543124.

Sutton, C. D. 2005. 11 - Classification and regression trees, bagging, and boosting. In *Handbook of statistics*, ed. C. R. Rao, E. J. Wegman, and J. L. Solka, Vol. 24, 303–29. Amsterdam: Elsevier.

Tintner, G. 1975. Multicollinearity. *Applied Mathematics and Computation* 1 (3):233–42. doi:10.1016/0096-3003(75)90035-1.

U.S. EIA. 2020. How much of U.S. carbon dioxide emissions are associated with electricity generation? Accessed October 30, 2019. https://www.eia.gov/tools/faqs/faq.php?id=77&t=11.

Uyanık, G. K., and N. Güler. 2013. A study on multiple linear regression analysis. *Procedia - Social and Behavioral Sciences* 106:234–40. doi:10.1016/j.sbspro.2013.12.027.

Varmuza, K. 2003. Boosting applied to classification of mass spectral data. *Journal of Data Science* 1:391–404.

Vu, D. H., K. M. Muttaqi, and A. P. Agalgaonkar. 2015. A variance inflation factor and backward elimination based robust regression model for forecasting monthly electricity demand using climatic variables. *Applied Energy* 140:385–94. doi:10.1016/j.apenergy.2014.12.011.

Watson, S., A. Moro, V. Reis, C. Baniotopoulos, S. Barth, G. Bartoli, F. Bauer, E. Boelman, D. Bosse, A. Cherubini, et al. 2019. Future emerging technologies in the wind power sector: A European perspective. *Renewable and Sustainable Energy Reviews* 113:109270. doi:10.1016/j.rser.2019.109270.

Yen, M. H., D. W. Liu, Y. C. Hsin, C. E. Lin, and C. C. Chen. 2019. Application of the deep learning for the prediction of rainfall in Southern Taiwan. *Scientific Reports* 9 (1): 12774. doi:10.1038/s41598-019-49242-6.

Zhao, Q., Q. Mao, Z. Zhao, T. Dou, Z. Wang, X. Cui, Y. Liu, and X. Fan. 2018. Prediction of plant-derived xenomiRs from plant miRNA sequences using random forest and onedimensional convolutional neural network models. *BMC Genomics* 19 (1). doi:10.1186/s12864-018-5227-3.

Zheng, Z. W., Y. Y. Chen, M. M. Huo, and B. Zhao. 2011. An overview: The development of prediction technology of wind and photovoltaic power generation. *Energy Procedia* 12: 601–8. doi:10.1016/j.egypro.2011.10.081.