AI is the New Electricity

Kevin Hu, Shagun Goel
{huke, gshagun}@stanford.edu
Category: Physical Sciences

1 Introduction

The three major fossil fuels – petroleum, natural gas, and coal – combined accounted for about 77.6% of the U.S. primary energy production in 2017. In fact, at current usage levels, oil will run out in 53 years, natural gas in 54, and coal in 110. Clearly, there is a need to move towards a society that refrains from exploiting unreplenishable natural resources.

Consequently, the world is evolving in the direction of renewable energy sources. While solar power and wind energy are rapidly gaining acceptance across the globe, a limiting factor to these resources is their dependency on climate conditions. In order to evaluate the efficiency of the aforementioned renewable sources, we built a machine learning model to predict solar and wind energy outputs at a given location, using local weather data. This, in turn, allows us to determine the optimal renewable energy source for a given location.

The input to our algorithm is weather data for a particular location. We then use one of four supervised learning models - linear regression, an SVM, an artifical neural network, and a generalized additive model (GAM) - to output the predicted solar energy output per unit area of the solar cell and/or the predicted wind energy output per unit area swept by the wind turbine.

2 Related Work

In general, previous attempts at solar and wind energy output predictions can be largely grouped into three categories: those utilizing artificial neural networks, those using linear formulas, and those using nonlinear formulas. Khatib et al. [1] are a prime example of attempts to use ANNs to make energy predictions. Rather than directly making energy predictions, however, they calculate a measure of solar irradiance through their ANN, which serves as an indicator of the energy production. In our paper, we will directly be predicting energy output per unit area, but this prediction is for standard photovoltaic cells. By measuring the solar irradiance, Khatib et al. ensure that their output can be utilized by any solar power generator. Mellit et al. [2] also utilize ANNs in order to forecast solar irradiance, but for a period of 24 hours only. They utilize solar irradiance and air temperature data as inputs to the ANN, which seems to limit the information they are providing to their model. In contrast, we attempt to provide a representative set of features that could empower a supervised learning model to make more accurate predictions.

Cammarano et al. [3] make a unique attempt at energy forecasting using nonlinear methods. Their paper is different from other such attempts since rather than focusing on short-term predictions, they aim to provide energy predictions for dynamically chosen horizons. Moreover, they do so for networks served by any ambient energy source like wind or solar energy, as opposed to our model which currently works for only solar and wind energy. Flocas [4] also uses a non-linear formula for making predictions by deducing the value of certain constant empirically from graphs. Despite these papers, Sayigh [5] contends that formulas for estimating solar radiation are ineffective, while leads us to believe that relying on machine learning techniques is indeed the right path.

Finally, Yang et al. [6] introduce the notion of using adaptive neural networks that use techniques like sliding window to adapt to unexpected changes in input data and better respond to real-time fluctuations. This is definitely an approach to be kept in mind for the future.

3 Dataset and Features

Keeping the objective of predicting solar and wind energy outputs using weather conditions in mind, there are three datasets relevant to our project. The first is an online weather API [7] that provides weather data for a given region (defined by a set of coordinates and a radius) over a given period of time. The second is a solar power dataset [8] which contains energy production data of photovoltaic cells across approximately 40 states in the United States measured at 5 minute intervals over an entire year. Finally, there is a wind energy dataset [9] which contains the average output of over 40,000 wind turbines across the United States in the years 2014 and 2017.

Note that it is necessary to synchorize these datasets. In particular, to construct a solar training example that uses certain weather values as input features and the solar production as the output value, we used the coordinates and time component of the solar data to request the relevant weather data. Similarly we again used the coordinates and the time component of the wind data to harmonize the wind dataset with the weather API.

The weather API response is represented as a JSON object that includes weather data for zero or more stations situated in the geographical region defined by the request. Data for each stations is typically indexed with respect to a list of timestamps that fall in the time period defined by the request. Figure 1 gives a graphical representation of the variation in a few weather features returned by a sample request.

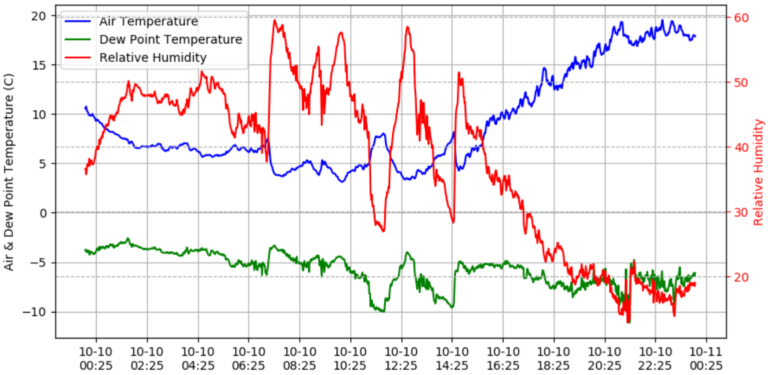

Figure 1: Sample weather timeseries data [7]

Note that weather data is quite inconsistent across the United States, i.e. often the field values are missing or invalid. In other words, there are very few features that have consistently valid values. This is why we decided to choose only 4 weather factors to serve as our input features for both solar and wind datasets: air temperature, relative humidity, wind speed, and pressure. However, we also do use an SVM implementation utilizes the Gaussian kernel. This is in order to extract hidden relationships between these input features, which is why projecting to infinite-dimensional space is a reasonable idea.

The solar data is also in the form of a timeseries discretized into 5-minute intervals over a full year. To process the solar dataset, we decided to average the energy production over a day and utilize any weather timestamps that fall in that range to serve as the input features. We also normalized these features by subtracting the mean and dividing by the standard deviation since they have widely differing ranges: pressure values of 10^5 Pascals are very common while temperature is generally within the range $[-50, 100]$ Celsius. In the end, we had 962,162 examples, which we shuffled and divided as 98%/1%/1% into the training, validation, and test sets.

Finally, the wind data is presented as the average energy output for wind turbines over an year, so we average the weather data over the same year to use as the representative input features. Once again normalization of inputs was key due to the same reasons as given above for the solar dataset. In this case we had 29,315 total examples, which we shuffled and divided as 70%/15%/15% into the training, validation, and test sets.

4 Methods

For each of solar and wind energy, we implemented four machine learning models - linear regression, SVM, artifical neural network, generalized additive model - that were trained, validated, and tested independently. In the next few sections we describe each of these models in greater detail.

4.1 Linear Regression

We used linear regression with mini-batch gradient descent as our baseline model to train a set of parameters for our four features discussed previously. At each iteration the update rule for the j^{th} parameter using mini-batch gradient descent is:

$$\theta_j := \theta_j + \frac{\alpha}{m} \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

where α is the learning rate and m is the batch size. Essentially, the linear regression algorithm finds a hyper-plane (in high dimensional space) that minimizes the root mean squared error of the predictions and the actual labels. At each step in the gradient descent, we update our parameter according to the computed gradient which further reduces the loss. This is repeated until convergence to give us the optimal parameters.

4.2 SVM

We also built an SVM model using the Gaussian kernel and L_1 regularization. SVM is a reasonable model because there can be many hidden, complex, and non-linear interaction between the weather features which cannot be captured by a simple linear model. In particular, because we only have four raw input features, the projection of our data into an infinite dimensional space using the Gaussian kernel is extremely powerful and can reveal valuable insights into the behaviour of the data.

In class, we learned about using SVM in a classification setting; but in our model, SVM is used to perform regression. As such, there is a slight change in the constraints of the optimization problem, which is shown below. In order to train the model parameters, we would like to

$$\min_{\gamma,w,b} \frac{1}{m}||w||^2 + C\sum_{i=1}^{m} \xi_i$$

subject to the constraints

$$\begin{cases} |y^{(i)} - w^T x^{(i)} - b| \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \quad \text{for } i = 1, \dots, m$$

The idea of SVM is to find a hyper-plane in high dimensional space that fit the data which maximizes the functional margin. As it is extremely improbable for that all the data points to lie exactly on the hyper-plane, we will ignore the error for points that lie within a small, fixed distance of their true values. And if the data is non-linear (cannot be modeled by hyper-planes), we project the data into a higher dimensional feature space using kernels, where the projected data becomes linear. In this case, we used the Gaussian Kernel to project the data into an infinite dimensional space.

4.3 Artificial Neural Network

We also implemented a fully connected neural network to model our data. We decided to use a fully connected neural network because the features are all highly interconnected and the connected layers allow us to model their unseen interactions with one another. Moreover, we decided to use ReLU as the activation function of the output layer because the energy output must be non-negative.

Deep learning and neural network is a powerful tool because it enables us to capture complex relationships between the features that we may not be able to reason without the aid of machines. Each of the links between two neurons models some hidden dependency between them. While fully connected layers may sacrifice run-time, it weighs the influence of all of the neurons in the previous layer when performing forward propagation. And in order to find the parameters (bias and weight matrices) of the neural network, we use a method called back propagation. Similar to gradient descent, at each iteration we update the parameters by taking the gradient of the loss with respect to the parameters.

4.4 Generalized Additive Model (GAM)

Finally, we conducted some research and found that a Generalized Additive Model may model our data well. Similar to General Linear Models (GLM), a GAM assumes that y follows some exponential family distribution. However, instead of trying to fit the data to some set of parameters θ, the goal of GAM is to model the output as a sum of functions of the inputs. That is, we want to find β_0 and smooth functions $f_1, \dots, f_M$ such that:

$$g(E[y|X]) = \beta_0 = f_1(X_1) + f_2(X_2, X_3) + \cdots + f_M(X_n)$$

GAM is a compelling model because it segregates the effects on the output from each of the input features. The use of arbitrary smooth functions f_i allows us to model arbitrary relationships between the input features and the output. Its additive nature also allows us to visualize individual relationships through dependency plots, as shown in the results section. In this particular case, our functions f_i's are all single variable, whose domains are a single input feature.

5 Experiments

5.1 Linear Regression

Due to the fact that we have more solar data than wind data, we implemented mini-batch gradient descent on the solar and wind data with batch sizes of 1000 and 100, respectively. In addition, after testing a range of learning rates, we found that its optimal value for both solar energy and wind energy data is $\alpha = 5 \times 10^{-10}$.

5.2 SVM

After testing a range of values for C and ξ, we found that the optimal values for the solar data are $C = 1$ and $\xi_i = 0.1$ for all i. Meanwhile, the optimal values for wind data are $C = 7$ and $\xi_i = 0.05$ for all i.

5.3 Artificial Neural Network

The architecture of our fully connected neural network is shown in the diagram below. The neural net has an input layer consisting of the four raw input features, three hidden layers both with ten neurons and sigmoid as their activation functions, as well as an output layer that uses ReLU as its activation function. The model is run with 1000 epochs, using a batch size of 10000 for solar data and a batch size of 100 for wind data. We initially implemented a neural net with two hidden layers, each with five neurons. However, the resulting models had abysmal accuracy, with training and test error that are approximately three orders of magnitude above the other models. As such, we deduced that our neural network is lacking depth and adjusted the architecture by adding an additional layer and increasing the number of neurons. We also tested different activation functions for the hidden layers, but sigmoid yielded the optimal performance so we left it as the activation function.

5.4 Generalized Additive Model

In GAM, the smoothness of the functions f_i can be adjusted using the number of splines (which is a way to estimate functions). Smoothing the functions allow us to capture the general trend while eliminating noise or fluctuations which can negatively effect our predictions. The details of the estimation will not be discussed, but the greater the number of splines, the smoother the functions will be. We played around with the spline numbers and found that num splines = (4, 20, 4, 100) for solar data and num splines = (5, 18, 10, 30) for wind data produces the highest accuracy. The four components corresponds to the features (wind speed, air temperature, relative humidity, pressure).

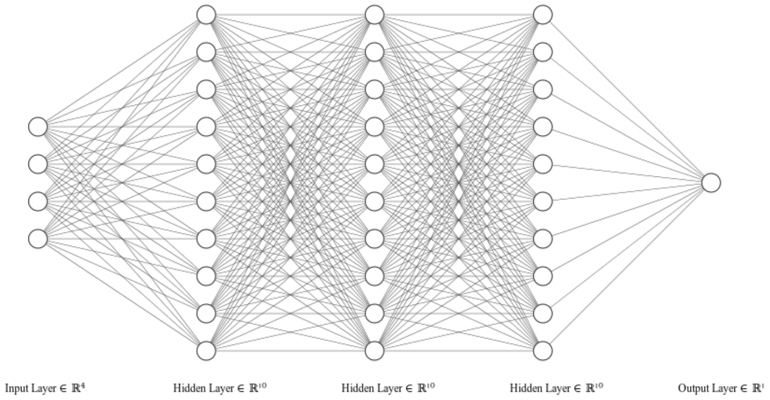

Figure 2: Neural Network Architecture

	Solar energy				Wind energy			
Model	Training error (RMSE)	Test error (RMSE)	Training set size	Test set size	Training error (RMSE)	Test error (RMSE)	Training set size	Test set size
Linear regression	0.129	0.395	940000	11081	0.11	0.163	25000	2000
SVM	0.218	0.385	940000	11081	0.049	0.074	25000	2000
Neural Network	0.121	0.371	940000	11081	0.082	0.533	25000	2000
GAM	0.121	0.369	940000	11081	0.047	0.0725	25000	2000

Table 1: RMSE errors

6 Results and Discussion

Since we have designed multiple regression models, the primary metric we use to measure the performance of a model is the average error. In particular, we utilize the root mean square error (RMSE) as our primary metric. Note that the RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2}$$

where m is the number of testing examples, $y^{(i)}$ is the true output of example i, and $\hat{y}^{(i)}$ is the predicted output of example i by the model.

Table 1 displays the RMSE values for each of the four aforementioned models for both solar and wind energy. Interestingly, a few key values have drastically changed over the past few days. The neural network initially looked to be quite ill-suited for energy predictions as it had significantly worse RMSE for both solar and wind energies. However, we hypothesized that this could be because the training set is quite sparse. In particular, notice that during a day, solar energy is only available from somewhere around 8 AM to perhaps 6 PM. But for the remaining 14 hours, i.e. the majority of the day, solar energy is unavailable, which is why the training set might have a large number of zero outputs. After getting rid of these zero entries, modifying the structure of the neural network to include a third hidden layer and have more neurons in each layer, and finally using the Adam optimization algorithm to utilize an adaptive learning rate, we were able to achieve much better performance.

It is, however, interesting to note that while the neural network achieves low training error for wind energy, it has a relatively high test error. This indicates that the neural network is overfitting to the training data in the case of wind energy. Similarly, note that while the training error is fairly low for all models in the case of solar energy, each of the models ends up having a much higher test error (relatively). This indicates that the models are once again overfitting in the case of solar energy. This could be so since the training set is so large while the validation and test sets are relatively small. For this purpose, we introduced regularization into our code, but it did not have a significant impact.

Most interesting, perhaps, is the performance of the generalized additive model (GAM). From Table 1 it is evident that both in terms of training and test error, GAMs are the optimal model out of those discussed so far for predicting renewable energy outputs. GAMs are unique since their additive nature allows us to explore and interpret individual features by holding others at their mean. Figures 3 and 4 show the original relationship between each individual feature and the energy output for both solar and wind energy.

Partial dependency plots allow us to gain a visual intuition on how features affect the output value. For instance, by looking at the solar output vs. relative humidity graph in Figure 3, one can easily deduce that, from the given data, an increase in relative humidity tends to lead to a downfall in solar energy output. Similarly the solar output vs. air temperature graph in Figure 3 seems to suggest, as one would expect, that an increase in air temperature leads to higher solar energy output.

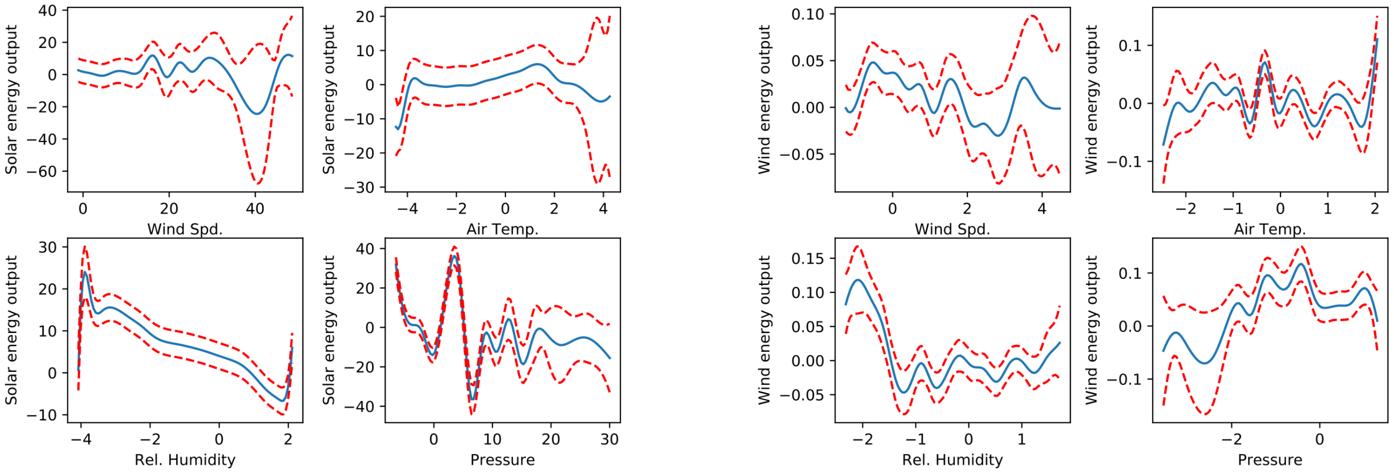

Figure 3: GAM original dependency plots for solar and wind energy

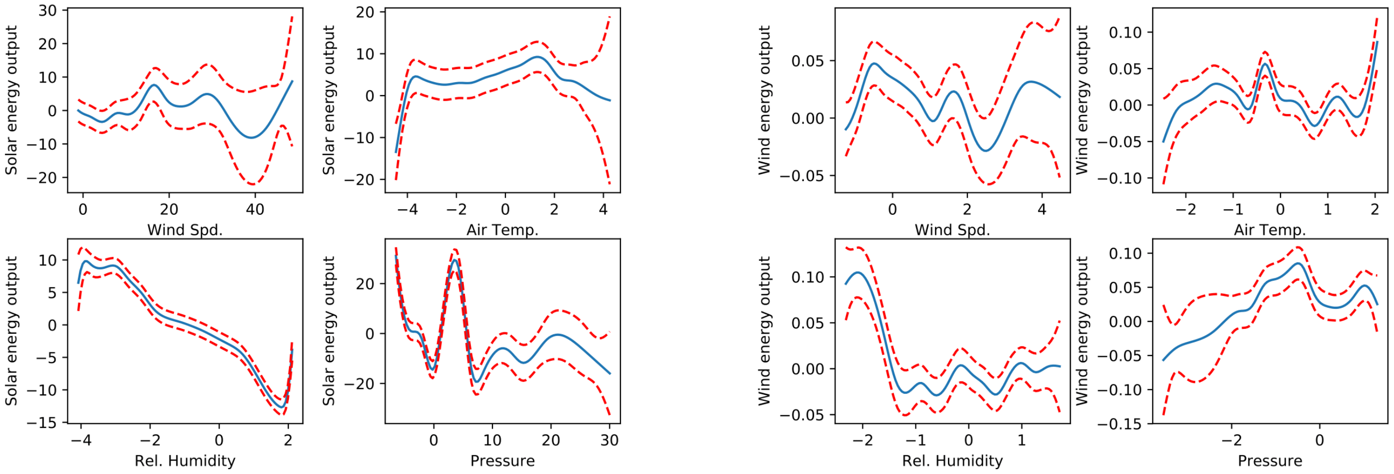

Figure 4: GAM smoothed over dependency plots for solar and wind energy

Another intriguing feature of GAMs is that since some of the graphs in Figure 3 seem to fluctuate a lot, we can actually ’smooth’ over these functions by increasing the number of splines to be used in order to evaluate these functions. As shown in Figure 4, this leads to a smoother and more consistent relationship between the input features and the output. Now even more relationships can be visually inferred! In particular, the wind output vs. pressure graph in Figure 4 seems to suggest that, at least initially, there’s a largely linear relationship between pressure and wind energy output.

7 Conclusion and Future Work

Over the course of this report, we have described the set up and utilization of four distinct supervised learning models in order to predict renewable energy outputs per unit area given the weather data for a particular location. During the training and testing process, GAMs produced the best performance for both solar and wind energy, albeit with close competition from neural networks in the former category and SVMs in the latter.

The success of GAMs might very well stem from the fact that they allow us to model arbitrarily complex relationships between individual features and the output. This might also serve as justification for the neural network’s performances, as it also allows us to glean and model increasingly complex interactions between the input features. Meanwhile, the fact that the SVM with its Gaussian kernel did not produce stellar results for solar energy suggests that the selected input space is reasonably representative of the features that affect solar energy output. At the same time, close competition from the SVM and its derived features in the case of wind energy leads us to believe that we may need to utilize additional features in the case of wind energy since the relationship between the chosen features and the output definitely does not seem linear.

Given more time and resource, we would like to develop more complicated models that perhaps combine the advantages of the models we have discussed in this paper. At a higher level, we would also like to extend our application to even more renewable energy sources. The end goal here is to determine the most efficient renewable energy source for any given location by utilizing not just weather data, but also information like proximity to resources (water bodies for hydro power generation for instance).

8 Contributions

Kevin Hu worked on the wind data collection, the linear regression model, and the SVM implementation. Shagun Goel worked on the solar data collection, the neural network, and the GAM implementation. The weather API was discovered and utilized jointly.

9 References

[1] Khatib, Tamer, et al. "Solar energy prediction for Malaysia using artificial neural networks." International Journal of Photoenergy 2012 (2012).

[2] Mellit, Adel, and Alessandro Massi Pavan. "A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy." Solar Energy 84.5 (2010): 807-821.

[3] Cammarano, Alessandro, Chiara Petrioli, and Dora Spenza. "Pro-Energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks." 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012). IEEE, 2012.

[4] Flocas, A. A. "Estimation and prediction of global solar radiation over Greece." Solar Energy 24.1 (1980): 63-70.

[5] Sayigh, A. A. M. "Solar energy availability prediction from climatological data." Space Environment and the Earth (1977): 61-82.

[6] Yang, Jin, Hugues Rivard, and Radu Zmeureanu. "On-line building energy prediction using adaptive artificial neural networks." Energy and buildings 37.12 (2005): 1250-1259.

[7] Synoptic Developers. “Mesonet API.” Mesonet API | Synoptic Data PBC, developers.synopticdata.com/mesonet/.

[8] NREL, National Renewable Energy Laboratory. “Solar Power Data for Integration Studies.” National Renewable Energy Laboratory (NREL) Home Page, www.nrel.gov/grid/solar-power-data.html.

[9] United States Geological Survey, Berkeley Lab. "United States Wind Turbine Database (USWTDB)." United States Geological Survey Home Page, https://eerscmap.usgs.gov/uswtdb/data/?fbclid=IwAR2kGpUd3d-os2SbmU4-nOIfhZTLPzEe8g6OyBksrqdFolbQmfI08z2Mq-g

Link for codes: https://drive.google.com/drive/folders/1_cANvcEUAxu9MPX-11jt0Wj_R6odrmpW?usp=sharing