Optimizing Local Least Squares Regression for Short Term Wind Prediction

Erin S. Keith, Richard Kelley, Anna Panorska, Frederick C Harris, Jr.
University of Nevada, Reno
ErinKeith.unr@gmail.com, Fred.Harris@cse.unr.edu

Abstract

Highly variable wind velocities in many geographical areas make wind farm integration into the electrical grid difficult. Since a turbine's electricity output is directly related to wind speed, predicting wind speed will help grid operators predict wind farm electricity output. The goal of experimentation was to discover a way to combine machine learning techniques into an algorithm which is faster than traditional approaches, as accurate or even more so, and easy to implement, which would makes it ideal for industry use. Local Least Squares Regression satisfies these constraints by using a predetermined time window over which a model can be trained, then at each time step trains a new model to predict wind speed values which could subsequently be transmitted to utilities and grid operators. This algorithm can be optimized by finding parameters within the search space which create a model with the lowest root mean squared error.

keywords: machine learning, wind prediction, local least squares regression

1 Introduction

In light of recent and growing concerns regarding global climate change, an increasing focus has been directed to traditional fossil fuel combustion based electricity generation techniques and their negative environmental effects. On the other hand renewable energy sources, such as wind energy conversion devices, ultimately depend on energy converted from the sun, as wind is moving air caused by the uneven heating of the Earth's surface [1]. While sustainable over our foreseeable future and lacking any greenhouse emissions when utilized for electricity generation, sun and wind sources are intermittent and have been, until recent decades, unpredictable.

Of all of the electricity being used in this instant, two thirds of it is being produced at fossil fuel burning power plants, such as natural gas or coal [2]. Integrating power plants into the grid which are dependent on intermittent sources such as wind creates particular challenges for grid operators, whose job is to guarantee each utility customer is able to receive the amount of electricity required at the quality to which they have become accustomed. Although often required by state regulations, utilities are sometimes reluctant to incorporate renewable sources into their Renewable Portfolio Standards due to these constraints [3].

The difficulty of predicting both the long and short term availability of wind resources presents an obstacle for initial capital investors, as the schedule for their return on investment is not guaranteed. Additionally, this intermittency significantly affects the negotiation of a wind farm's power purchase agreement, a contract made between a power plant owner and the utility providing electricity to customers, when the projected revenues of the project would otherwise be uncertain and so some guarantee as to quantities purchased and price paid are required to make the project viable [4]. These practical and financial obstacles can prevent this beneficial technology from being more widely adopted.

In order to create a solution to the availability intermittency problem, a simple supervised machine learning regression technique which has been applied to wind speed forecasting was implemented. This method was preferred as it avoids the computational resources needed to process all data at once, provides a simplicity in processing that can be utilized on device microprocessors, and allows the algorithm to automatically adjust to seasonal or even shorter term weather pattern changes without any hand tuning. Optimization techniques were applied to determine the best parameters for the model, then whittle down the time necessary to determine these best parameters.

The following was undertaken as an endeavor to explore the unique challenges facing wind technology in particular and concepts related to the proposed solution. After the necessary background is expounded in Section 2, the solution's design and implementation is presented in Section 3, with the experimental results offered with visualizations in Section 4. Finally, in Section 5, current applications and ideas for future expansion of the project are discussed.

2 Background

So much of our modern lives are dependent on controlling and manipulating the flow of electrons. With the risks and negative consequences of current electricity generation techniques the question is, "at what cost?" Current, traditional, fossil fuel burning electricity generation techniques lead the pack in contributing to greenhouse gasses such as carbon dioxide, methane, and nitrous oxide [5].

Luckily, there are other resources such as sunlight and wind, which can be harnessed to help meet our electricity consumption. Much like water flows through the oceans, air flows through our planet's atmosphere, gases moving from high to low atmospheric pressure instead of liquid water. Invisible to the eye, this mysterious force carries a burgeoning potential in energy which humans have been straining to capture for millennia.

Modern utilization of wind energy for transformation into electric energy requires fine grain time dependencies. A combination of signal processing and machine learning techniques are valuable in predicting wind speed, and will subsequently prove to be vital in power generation from wind farms.

2.1 Our Electric Grid

Our electric grid is a highly complex, interconnected network of power generation stations, transmission lines, and customers. These customers can be industrial, commercial, and residential, and represent the load in the giant nationwide circuit.

All the electricity we're using at this moment is being generated right now. At power plants across the country, in the control center, there is a smooth and continuous process of deciding when to ramp up electricity production to match demand. "Multiple sources and loads can be connected to the transmission system and they must be controlled to provide orderly transfer of power. In distributed power generation the generators are geographically distributed and the process to bring them online and offline must be carefully controlled." [6]

This intricate system requires control centers manned with competent operators making important decisions as to how grid operations are managed so that each customer receives their instantaneous supply of electricity. A grid operator's chief responsibilities include forecasting this electricity load, scheduling the cheapest possible generation methods, ensuring that the transmission systems are not overloaded or damaged, and reacting to unexpected changes. [7] Luckily, due to the routine behavioral habits of people, load tends to follow daily and seasonal trends making load forecasting relatively straightforward.

2.2 Current Methods

Because the problems presented by the integrating wind power into our grid infrastructure have been apparent for so long, there have been many attempts and approaches to predicting wind speed, especially over the past three decades. An efficient introduction to different methodologies is a literature review of some of the different methods available. A more detailed and constructive analysis of two additional approaches will help establish some of the aspects of this area which could still benefit from attention and effort at improvement.

One of the first services offered by A Literature Review of Wind Forecasting Methods is a pointed description of the different ranges of forecasting windows. Ultra-short-term forecasting, which spans from a few minutes to 1 hour, is helpful for real-time grid operations. Additionally, there is the short-term window range, 1 to several hours, which is best for economic load dispatch planning. There is also medium-term forecasting, which is several hours to 1 week and used for reserve requirement decisions. Finally, there is the long-term window spanning 1 week to 1 year, or more, for optimal operating cost determination or even a feasibility study for wind farm design. [8]

[8] also specifies a number of different methodology types for forecasting. The Persistence method, or Auto-Regressive Integrated Moving Average (ARIMA) is often the most accurate method when predicting values within the ultra-short term range. Therefore, it should be utilized to gauge any novel methods which may be developed.

Physical methods include numerical weather prediction (NWP) models developed by meteorologists. Even though these methods require huge amounts of supercomputing resources, they're still used most as the existing commercial wind power forecasting methods.

Many different variations of ARIMA, including AR and ARMA, are encompassed under the statistical methods umbrella. Additional approaches such as regression and Bayesian models are also counted within this collection.

Another pool of models, termed spatial correlation, utilize not one but multiple site measurements to predict the wind speed point at one of the sites. This technique is often combined with other techniques such as neural networks.

Neural networks, both back propagation and recurrent, are also used on their own for wind speed forecasting, as a member of the artificial intelligence category of methods. This category also includes fuzzy logic methods, support vector machines, some of which are even "found to be more accurate that traditional statistical time series analysis." [8]

The most important point made by the author, as it pertains directly to this work, is one of the recommendation for the future of wind forecasting. It is suggested to "do further research on the adaptive parameter estimation. The models have ability to automatically adapt to the changes of the farms and the surroundings." [8]

3 Optimizing Local Least Squares Regression for Short Term Wind Speed Prediction

When originally researching predicting changes in wind direction for use in controlling wind turbine yaw using neural networks, it was suspected that the model's integrity would degrade over time, especially as seasonal weather patterns changed, which has been supported by ARIMA's position as the benchmark in wind speed forecasting results. Rather than retrain the network periodically when the error escalates to a certain threshold a certain percentage of the time, it appears to be more practical to have a simpler model to be updated as new data comes in.

The contributing attribute of this work is the ability to quickly determine the values for these parameter which produce the best results, utilizing typical regression metrics (RMSE, MAE, percentage error, and R^2) as the evaluation criteria. The benefits of this method include increased accuracy over both choosing arbitrary parameters and utilizing the same model over longer periods of time. Necessary computing resources would be minimal and economic, and the data collection equipment necessary to make these wind speed predictions would consist solely of the system's own onboard sensors.

3.1 Features

While Least Squares Estimation (LSE) utilizes linear algebra to determine the best coefficients for a polynomial to describe the trend of wind speeds, there are important algorithmic parameters which affect the performance of the model. These significant parameters include lambda, which can be described as the regularization parameter, dimensions, which is also defined as the number of terms contained in the polynomial, and training window size, which here becomes the number of previous time-steps of training data included in the Least Squares Estimation kernel training.

The goal of this work was to create a simple and straightforward method for determining the best parameters to use in LSE of Wind Speed which would minimize the evaluation criteria for predictions made at certain time windows in an amount of time that would not be prohibitive in an industry setting.

An additional conjecture included the assumption that training the model on large amounts of data, while improving accuracy, would become cumbersome to compute, and that computing resources would be strained without any significant improvement in accuracy. This was supported by the results list in Table 3.1. It is significant to consider that the point on which these models were test was temporally distant from the training data, which was demonstrated earlier to contribute to error. While there is an initial improvement in accuracy, a reflection point becomes evident around 9000 training points. Most importantly, at around 14,000 data points, the test machine runs out of memory resources entirely.

3.2 Design

Preliminary experimentation supported the assumption that the coefficients which determine the shape of polynomial increasingly lose their ability to fit the data. Even though the increase over the year is slight, errors would continue to accumulate and the model would lose accuracy over time.

These findings led to a design decision which would focus on the ability to quickly prepare accurate models utilizing a smaller training set located temporally near the desired prediction. This approach would take advantage of a continuously updating process to eliminate the eventuality of the model becoming obsolete.

The algorithm was designed to provide optimal parameters dependent on the specified prediction window. The optimization loop will determine which LSE parameters provide the smallest root mean square error. These LSE parameters are then utilized over the entire data set to predict wind speeds for the best prediction window.

The parameter **lambda**, which can be described as the regularization parameter, is used to control the fitting parameters and guard against over fitting. In a graphical sense, it smooths the ripples which would otherwise occur in a polynomial of higher degrees. The **dimension** parameter is what designate how many terms will be included in the polynomial which describes the trend of the data. In the graph, it determines how many inflection points are utilized to follow the data's curves. Finally, the training **window** parameter determines how far back in time the algorithm goes back to use training data.
Performance measures include RMSE calculated as

$$RMSE = \sqrt{\left(\frac{\sum_{i-1}^{n}(p_i - o_i)^2}{n}\right)}$$

where p is predicted wind speed, o is the observed wind speed, and n is the number of predictions in the set. The MAE value is simply

$$MAE = \left(\frac{\sum_{i-1}^{n}(p_i - o_i)}{n}\right)$$

Together these two metrics help determine the variance in the magnitude of errors.

3.3 Technologies

Due to its popularity in industry and research, and the extensive libraries available, especially for scientific computing and machine learning, the main technological choice was to utilize the Python programming language [9]. Python is also well supported and respected in the engineering community with advanced tools for testing and debugging.

The Python package **matplotlib** is a 2D plotting library which produces publication quality figures and was utilized for visualizing data, and plotting histograms and graphs. [10] This collection of modules greatly contributed to the ease with which we were able to generate high quality plots.

NumPy is a prominent package in the machine learning community. [9] This library of functions is comprised of Matlab-like processing features such as array object creation and linear algebra tools. In addition to utilizing the NumPy array objects, simple linear algebra functions were employed including the **dot** function to find dot products, the **identity** function for creating the identity matrix, and most importantly the **linalg.solve** function for solving systems of equations providing the **α** term.

The final Python module applied is **scikit-learn**, an esteemed machine learning package for data mining and data analysis. Although none of the machine learning specific functions were used, the provided **mean_squared_error** function was essential in computing the error used to compare the effectiveness of the derived parameter.

3.4 Implementation

The final design is broken down into three steps which include initial preprocessing of the input data to ensure it is properly screened to avoid misleading results. Next we conduct a parameter search for the necessary values. Finally, a test run of the algorithm is conducted over the entire data set.

Input data was downloaded from the Nevada Climate Change Portal [11]. Four features were available from the original data: wind speed, humidity, barometer, and temperature. It's evident by studying the measurements, that there's a wide discrepancy in their ranges. To avoid the problem of incommensurability, the decision was made to convert each feature to dimensionless variables through normalization.

A Python module was created to handle these data preprocessing steps. Data gets loaded from the csv file into Python lists, with any non-numerical values being replaced by zeros and any values less than one being rounded according to Python's **rint()** function. Those lists are then converted to NumPy arrays, normalized by dividing them by the maximum value in the array, and then saved as a multi-dimensional data array.

During the search for optimal algorithm parameters, a section of this data array, made up of all four features listed above, including wind speed, is used to train the model on the wind speed labels which correlate with a particular window of time into the future. Training occurs by using this section as the inputs for the kernel computation. Then alpha is computed by solving the system of linear equations involving the kernel, lambda, and the labels.

Initial exploration of the algorithm parameters comes into play through a very straightforward exhaustive search. The program iterates over ranges of window sizes, dimensions, and lambdas while keeping track of the values which produce the minimum root mean squared error for use in the actual testing and running of the local least squared error algorithm.

The simulated annealing optimization was implemented. The temperature scheduler decreases the temperature by some cooling constant. A new set of parameters is chosen randomly, then evaluated to determine their relative errors. If the current parameter error is greater than the new parameter error, the new state is automatically saved. Otherwise the "dice are rolled" to see if the new state will be saved anyway.

Finally, we have optimal values for our algorithm parameters to pass to a function very similar to our training least squares estimator, but which will run over the entire data set, updating the model at each step and saving every root mean square error calculation to track the performance of the model, as presented in the Section 4.1.

4 Results

4.1 Polynomial Kernel

The kernel utilized in experimentation was the polynomial kernel. During preliminary investigation to determine the possible parameter combinations and associated evaluation metrics, an exhaustive enumeration methodology was implemented. Through iterations of all possible combinations of parameters, the root mean square error varies broadly. Displaying this error values as the algorithm iterates through the parameter ranges, as in Figure 4.1, illustrates the range of our search space and demonstrates the importance of heuristics in parameter selection.

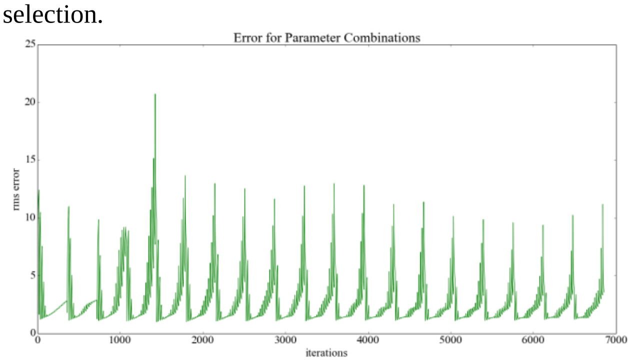

Figure 1: Graph of RMS Error for Different Parameters

The data deliver a couple of important insights. First, it appears that predicting wind speeds becomes more difficult as the prediction window expands, as evidenced by the increasing RMS values and decreasing R2 scores. Moreover, besides the consistent lambda term, there does not appear to be a trend in the parameter values or their combinations, which highlights the important of heuristics in parameter selection. It is also of significance to note that the exhaustive enumeration training times generally exceed thirteen minutes, with the greatest training time surpassing seventeen minutes. Finally, utilizing some of these combinations of parameters for wind speed predictions produce the following graphical results.

Predicting wind speeds at a five minute window produces a low root mean squared error; as expected, an examination of the graph in Figure 4.2 shows that the model does a good job of approximating the time series with the exception of a few sharp outliers. With the highest R2 value and lowest RMS error, this graph represents the best results for exhaustive enumeration parameter search.

The 35 minute results, in Figure 4.3, are worth examining, as the percent error is slightly larger, even though the RMSE is low. These predictions tend to be smoother and closer to the actual values. Another important feature is that the model does a fair job of handling abrupt changes in velocities.

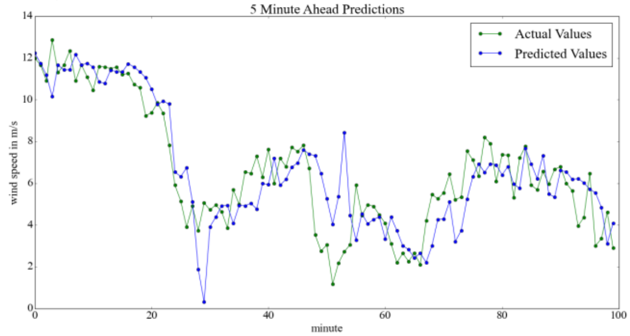

Figure 2: Exhaustive Enumeration Wind Speed Predictions for 5 Minute Window

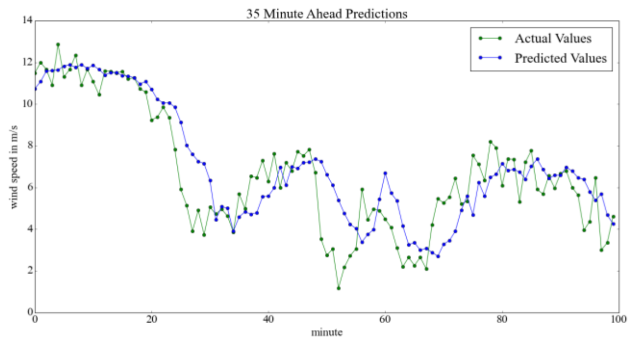

Figure 3: Exhaustive Enumeration Wind Speed Predictions for 35 Minute Window

The graph in Figure 4.4 captures the model generated for a 60 minute prediction window. With the lowest R2 score and highest RMSE, this model has the poorest performance. The model produced with these parameters appears to “shadow” the previous steps actual wind speed value, which would not be helpful to the intended user.

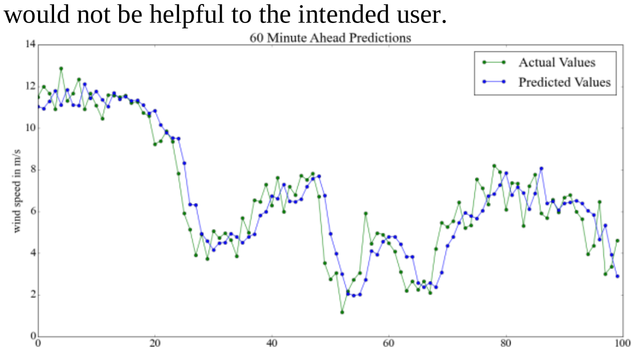

Figure 4: Exhaustive Enumeration Wind Speed Predictions for 60 Minute Window

One of the four principal assumptions which can justify the use of a linear regression model is a normal distribution of errors [12]. This assumption is clearly supported by the histogram presented in Figure 4.5, which gives confidence in our model and its parameters.

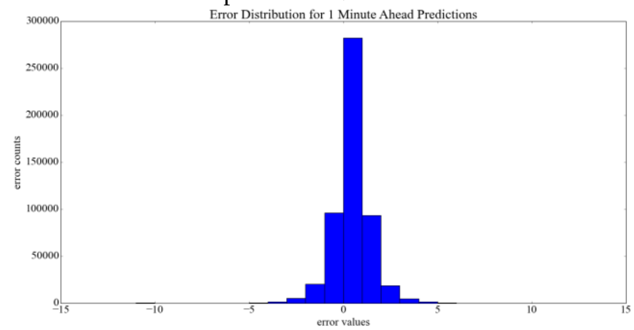

Figure 5: Histogram of Errors

4.2 Optimization with Simulated Annealing

While the polynomial models generated under the exhaustive enumeration approach provide good results, the training times for certain models can take over seventeen minutes, which would be prohibitive for use in an industrial setting. In comparison, less than one and a half minutes of optimized training time would likely fall within application constraints. The additional feature of faster training time supports Simulated Annealing as an optimization method. Moreover, even though the Simulated Annealing training times are at least 90% shorter, their RMSE values are not significantly higher.

Table 1 Comparison of Training Times and RMSE Values

Prediction Window	**Enum** (sec)	**SA** (sec)	**Enum RMSE**	**SA RMSE**
5 minute	1026.82	73.26	0.63	0.74
35 minute	810.46	68.19	0.72	0.73
60 minute	825.84	85.37	0.83	0.96

It is important to note that because of the randomized aspect to parameter generation, it is not guaranteed that the optimizing simulated annealing algorithm will produce the same parameters. Another helpful feature of utilizing the optimization is that, because the search times are so much shorter, a wider range of variables could be searched to provide the more expressive models.

Similarly to the model generated by the exhaustive enumeration parameters, the five minute ahead model illustrated in Figure 4.6 provides a good approximation for most points. Since the lambda value is significantly higher than the other model, the curve is much smoother and doesn’t provide the expressive predictions for the sudden change in velocities. Still, this model provides comparable results at a small fraction of the training time cost.

The 35 minutes simulated annealing results, included in Figure 4.7, are virtually indistinguishable from those generated by exhaustive search methods, which is to be expected as the R2 score and RMS value are almost equivalent to each other. In both this model and the previous one, it is interesting that such disparate parameter combinations manage to produce such analogous results, providing justification for utilizing the simulated annealing heuristics.

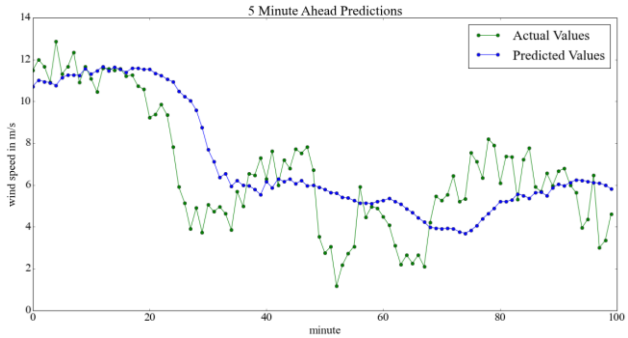

Figure 6: Simulated Annealing Wind Speed Predictions for 5 Minute Window

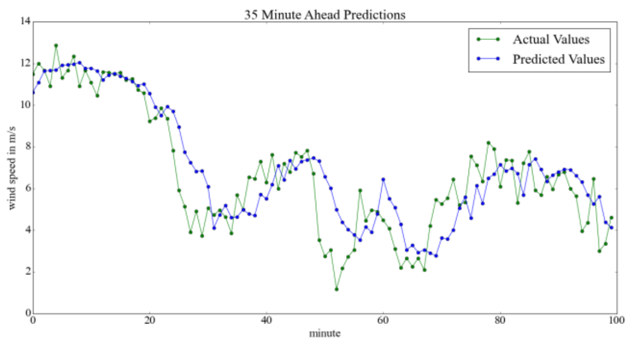

Figure 7: Simulated Annealing Wind Speed Predictions for 35 Minute Window

Finally, we have the 60 minute model in Figure 4.8 generated by the simulated annealing parameter search. The model overshoots the actual values is significant places, missing some of the important peaks and valleys. Root mean squared error is slightly higher and the R2 score is definitely low, but the model is still capable of describing general trends in wind speed behavior, which is still of benefit.

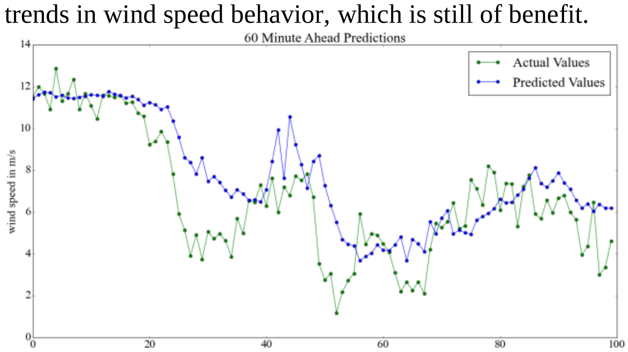

Figure 8: Simulated Annealing Wind Speed Predictions for 60 Minute Window

Based on the results collected and presented here, it's evident that nonlinear regression utilizing a polynomial kernel is a viable method for predicting wind speed. Utilizing a sliding window for model training prevents model deprecation over time. To continuously train these models, certain parameters must be fixed. Instead of enumerating over all possible combinations of parameters, it is more efficient to utilize the simulated annealing optimization technique, without any significant increase in RMSE values, as long as model consistency is not a problem constraint or user requirement.

5 Discussion

As humankind searches for ways to cope with and counteract the negative consequences of our rampant burning of fossil fuels, new technologies and political policies emerge. Applying an optimization algorithm for localized least squares regression techniques to predicting wind speeds is my contribution to our stalwart stand in support of new enterprises in electricity generation. Being able to accurately predict wind speeds may allow wind turbine farms to better predict their own electricity output, which will help make them more profitable and therefore more attractive to initial investors, and hopefully break down barriers to their entry into the electricity generation market.

5.1 Applications

In addition to the economic benefits to potential investors of reducing financial risk and encouraging the expansion of the wind electricity generation business, the main application for this work was envisioned as providing assistance to electricity grid operation, adding reliable resource forecasting to their current process which has historically been limited to load forecasting [13]. Relying on renewable resources, such as wind energy, provides challenges as their availability tends to be intermittent. If near future wind farm generation output is better understood then their integration into the existing grid network will be more effective and less detrimental to overall operation, as there won't be any sudden drop in supply which could a cascade of failures throughout the system.

Furthermore, more accurate power output predictions would reduce the necessity of continuing the practice of exploiting spinning reserves at traditional fossil fuel burning generation sites, such as coal and natural gas. To be ready to compensate for any sudden decrease in supply which might be caused by drops in renewable energy resources such as solar or wind, a number of these fossil fuel burning generators must be kept running and online but without any load, to offset ramp up times so they can be ready in time to meet instantaneous demand and compensate for sudden losses, sometimes in as little as ten minutes [14]. These spinning reserves contribute excess CO2 and are a terrible waste of fuel and water resources.

Finally, since the local least squares regression techniques have been demonstrated to be effective, industrial-sized wind turbine manufacturers and wind farm engineers could easily incorporate them into their supervisory control and data acquisition systems and proprietary software. The algorithm is simple enough and machine learning libraries are readily available to aid non-data scientist, software engineers in incorporating this technique into their current software designs. Additionally, the necessary sensors are already incorporated onto the turbines which utilized in commercial wind generation farms.

5.2 Future Work

One of the first further experiments we would like to conduct would be to use more than one geographical location to see if that would help improve predictions for expanded prediction windows, as it does with statistical approaches [13]. Utilizing a cluster of sites' data may provide more features to make the predictions more accurate.

There are a number of ways to expand this technique to extend functionality. For instance, predicting wind speed is a preliminary step to predicting output power. There are power electronic components and aggregation factors which contribute to the actual electricity delivered to the electrical grid. [14] A more efficient model would map climate features of a geographical area directly to the total power output of an entire wind farm. This method of optimization can also be applied to research models which incorporate local regression into their prediction approach.

As stated, the algorithm is simple and not computationally expensive. As such, it would be interesting to prototype a full system utilizing the Raspberry Pi platform. Necessary equipment would include an anemometer and other weather sensors, the Raspberry Pi computer or Netduino micro-processing board, and a small wind generator. This prototype platform could facilitate direct prediction of electricity output instead of prediction of wind speed and subsequent calculation of the electricity output. Additionally, an interface allowing a user to enter a prediction window would make the prototyped system complete.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant numbers IIA-1329469 and IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Coriolis Energy Ltd, "Wind and its origins," 2014. [Online]. Available: http://www.coriolis-energy.com/wind_energy/wind.html.

[2] U.S. Energy Information Administration, "What is U.S. electricity generation by energy source?," [Online]. Available: http://www.eia.gov/tools/faqs/faq.cfm?id=427&t=3. [Accessed December 2014].

[3] U.S. Energy Information Administration, "Most states have Renewable Portfolio Standards," [Online]. Available: http://www.eia.gov/todayinenergy/detail.cfm?id=4850. [Accessed December 2014].

[4] R. Park, "The Power Purchase Agreement (PC for Solar)," in Energy Project Financing: Resources and Strategies for Success, The Fairmont Press, Inc., 2009, p. 93.

[5] United States Environmental Protection Agency, "Sources of Greenhouse Gas Emissions," [Online]. Available: http://www.epa.gov/climatechange/ghgemissions/sources/electricity.html. [Accessed 7 2015].

[6] K. Roebuck, Wireless Power: High-impact Emerging Technology - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors, Emereo Pty Limited, 2011.

[7] D. Ulmer, "Fundamentals of the Power Grid and Electricity Pricing," in Energy Leader Webinar Series, 2014.

[8] W.-Y. Chang, "A Literature Review of Wind Forecasting Methods," Journal of Power and Energy Engineering, vol. 2, pp. 161-168, 2014.

[9] M. A. T. Lira, M. D. S. Emerson, M. B. A. Jose and V. O. V. Gielson, "Estimation of wind resources in the coast of Ceara, Brazil, using the linear regression theory," Renewable and Sustainable Energy Reviews, vol. 39, pp. 509-529, 2012.

[10] Choosing R or Python for data analysis? An infographic, The DataCamp Blog, 12 May 2015. [Online]. Available: http://blog.datacamp.com/r-or-python-for-data-analysis/?imm_mid=0d2357&cmp=em-data-na-na-newsltr_20150520. [Accessed 28 July 2015].

[11] J. Hunter, "Matplotlib: A 2D graphics environment," Computing In Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[12] Nevada Climate Change Portal, "Sensor Data," [Online]. Available: http://sensor.nevada.edu/SENSORDataSearch/. [Accessed 01 November 2014].

[13] Duke University, "Regression diagnostics: testing the assumptions of linear regression," [Online]. Available: http://people.duke.edu/~rnau/testing.htm. [Accessed 17 July 2015].

[14] Caiso, "Spinning Reserve and Non-Spinning Reserve," 2006. [Online]. Available: http://www.caiso.com/Documents/SpinningReserveandNonSpinningReserve.pdf.