Short-term solar forecasting performance of popular machine learning algorithms

ABSTRACT

A framework for assessing the performance of short-term solar forecasting is presented in conjunction with a range of numerical results using global horizontal irradiation (GHI) from the open-source SURFRAD data network. A suite of popular machine learning algorithms is compared according to a set of statistically distinct metrics and benchmarked against the persistence-of- cloudiness forecast and a cloud motion forecast. Results show significant improvement over the benchmarks with tradeoffs among the machine learning algorithms dependending on the desired error metric. Training inputs include time series observations of GHI for a history of years, historical weather and atmospheric measurements, and corresponding date and timestamps, such that training sensitivities may be inferred. Prediction outputs are GHI forecasts for one, two, three, and four hours ahead of the issue time, and are made for every month of the year for seven locations. Photovoltaic (PV) power and energy outputs can then be made using the solar forecasts to better understand power system impacts.

1. INTRODUCTION

The integration of high levels of solar power into the grid poses a significant challenge to grid operators due to the uncertainty and variability of solar genereration. Improving solar irradiance forecasting will ease the ramping demands of back up generators for the grid. One of the goals of the Power Systems Design & Studies (PSDS) group within the Energy Systems Integration Facility (ESIF) at the National Renewable Energy Laboratory (NREL) is to research various methods to improve solar forecasting. Machine learning is one of the newest approaches to this challenge and shows promise to make large improvements to short-term solar forecasting.

Accurate forecasting of solar energy production for unit commitment can reduce solar generation uncertainty, which translates to significant savings. A study by Lew et al. finds that $5 billion savings could be achieved on the Western Electricity Coordinating Council (WECC) per year by integrating solar and wind forecasts into unit commitments.1 Inman et al. provides a comprehensive review of state of the art methods in solar forecasting, which primarily focuses on averaged rather than instantaneous forecasts.2 Lorenz et al. show that for short-term forecasts, accuracy is greatly improved by applying model output statistics (MOS) to Numerical Weather Prediction (NWP) methods.3 A variety of regression approaches have been applied to improve short-term solar forecasting.4-8 For 15-minute to four-hour ahead forecasts, hybrid machine learning approaches have achieved significant improvements over the traditional NWP models.9 A study by Perez et al. finds that the use of inputs such as satellite data improves the accuracy of short-term forecasts at several surface radiation (SURFRAD) sites.10 Other studies such as Gordon et al. incorporate other exogenous observations such as relative humidity and cloud cover, which are utilized by the forecasting methodologies to improve forecasting accuracy.11

The method developed in this paper utilizes irradiance and exogenous weather time series data from seven publicly available weather stations in the surface radiation (SURFRAD) network, and uses different machine learning (ML) algorithms to predict solar irradiance forecasts one, two, three, and four hours ahead. This paper begins by describing the available data and the preprocessing techniques that were applied in this study. A brief overview of the ML forecasting methods is then given, followed by results and discussion comparing the performance of the ML models against the benchmarks and against each other. Finally, concluding remarks and suggestions for future research will be presented. The work presented in this paper was completed at the Department of Energy’s (DOE) National Renewable Energy Laboratory (NREL).

1. PROCESS
2. Preprocessing input data

The methodologies developed in this paper were trained and tested on data from the SURFRAD observation sites in Desert Rock, NV; Fort Peck, MT; Boulder, CO; Sioux Falls, SD; Bondville, IL; Goodwin Creek, MS; and Penn State, PA. Each site has 11 years of weather measurements, at one-minute resolution from 2009 to 2014, and at three-minute resolution from

2004 to 2008. This array of sites offers climatically different weather situations. Global horizontal irradiance (GHI) at the SURFRAD sites is best represented by the *global downwelling* solar

measurements. The clear sky GHI at time *t* is denoted by 𝑡 and represents the theoretical

𝑐𝑙𝑒𝑎𝑟

GHI at time *t* assuming zero cloud coverage, and is computed using the Bird model.12 Clear sky index is a metric of cloud cover that has been used extensively in forecasting literature.13-15 The

clear sky index at time *t* denoted by (𝑡) is the ratio between the instantaneous observed 𝑡

i

and the theoretical maximum 𝑡

𝑐𝑙𝑒𝑎𝑟

. Current time, temperature, relative humidity, wind speed,

wind direction, pressure, thermal infrared, 𝑡, 𝑡 , and (𝑡), are used as independent

𝑐𝑙𝑒𝑎𝑟

i

variables for the input training vectors.

Rather than training on the observed instantaneous GHI values at the one, two, three, or four-hour ahead forecast horizons (*f.h.),* which may not be representative of the *most probable*

ƒ.!., the ML models are trained on the average clear sky index for the hour ending at the *f.h..*

The averaged hourly clear sky index ending at time *f.h.* is denoted by (ƒ.!.), as in

𝑎

f.!

𝑠!f.!.!60

60

𝐾𝑡i 𝑠

Equation 1

(ƒ.!.) is used as the dependent variable when training each model, and the models are then used to predict (ƒ.!.) when given unseen test vectors. The forecasted (ƒ.!.) value is then multiplied

𝑎

ƒ.!.

by

𝑐𝑙𝑒𝑎𝑟

𝑎 𝑎

from the Bird model to predict ƒ.!., as in

ƒ.!. = (ƒ.!) ∙ ƒ.!.

Equation 2

𝑝𝑟𝑒𝑑i𝑐𝑡i𝑜𝑛 𝑎 𝑐𝑙𝑒𝑎𝑟

This ML forecast is finally compared to the testing input’s corresponding ƒ.!.

𝑜𝑏𝑠𝑒𝑟𝑣𝑒𝑑

from the

SURFRAD data to assess forecasting accuracy.

Data was partitioned by month and any entries with missing or misreported data were removed. All nighttime entries with current or future GHI readings below 20 W/m2, were removed to improve the performance of the ML algorithms. Each ML algorithm has many hyper- parameters that can be tuned and these internal parameters were set using a grid search method. Predictions were made for each forecasting situation at a frequency equal to the forecast horizon time span. For example, when forecasting GHI for three hours ahead, the ML models made predictions at three-hour intervals every day of the month for all daylight hours.

1. Description of forecasting methods
   1. *Persistence of cloudiness*

Persistence forecasts use the current cloud cover to predict the future GHI. The forecast

horizon’s clear sky index is set to the current clear sky index at t and multiplied by GHIf.h.

cl#$r

. This

simple model is most effective for very short-term forecasts (e.g. minute ahead range), but can also be used to make one to four-hour ahead forecasts. Persistence forecasts were provided as a benchmark for the forecasts performed by ML methods in this study.

* 1. *Support vector machines*

Support Vector Machines (SVMs) have been used in many solar forecasting applications, and have been shown to work well in conjunction with other methods.16-18 SVM regression estimates a target function based on the training instances. The output observations are assumed to take the form of *yi = (wi • xi) + b*, where *yi* is the output observation for training instance *i, xi* is the input training vector for instance *i, wi* is a weight vector which defines the functional form, and *b* is the bias constant. SVMs operate by transforming a non-linearly separable feature space into a multi-dimensional space where variables can be seperated by a 3-D *hyperplane*. SVMs map the original data into this higher dimensional space using a technique known as the kernel trick, which allows for different perspectives on the data and makes linear relations more apparent. The final objective is to minimize the deviation errors between the output observation *yi*, and the linear functional form *= (wi • xi) + b*, while maximizing the margin of space on either side of the hyperplane.

* 1. *Artificial neural networks*

Artificial Neural Networks (ANNs) are one of the most popular machine learning methods used in solar forecasting.19-21 ANNs operate as computational models of the neural networks found in the human brain. They contain layers of nodes with connections between nodes in adjacent layers. The input layer consists of one node for each input signal such as current time, temperature, GHI, etc. The output layer has a single output node for the corresponding GHIf.h.. Between the input and output layers are one or more hidden layers, which contain a pre-determined number of nodes. Each node receives a weighted sum of input from the signals in the previous layer, and applies an activation function to the weighted sum. The weight of each connection is akin to the strength of neural connections in the brain. The output from the network is compared to the known training output value, and back propogation is performed to adjust the weights between the network’s nodes. This process is repeated until proper weights have been determined for the training data and the network can then be used to test unseen data.

* 1. *Random forests*

A Random Forest (RF) machine learning algorithm, is a forest made of an ensemble of decision trees. RFs have been used in solar forecasting in several studies.22-24 Each decision tree directs input through several classification and regression decision nodes. Each node splits into two possible branches, or outcomes, with each branch leading to another node. The process repeats until a terminal node is reached and an output value is connected to the given inputs.

Performance of a single decision tree can be improved by training multiple regression trees with different structures and then averaging their predictions. RFs add random feature selection at each node for greater diversity in the decision tree models. Individual decision trees may have a bias due to their specific feature selection and structure but a RF averages over all the decision trees, significantly minimizing the error bias for the final prediction.

* 1. *Gradient boosting*

Gradient Boosting (GB) is a less common ML approach to solar forecasting, and is used in this study to extend the RF approach.25 As with RF, GB uses an ensemble of decision trees to make a more accurate prediction. In GB, however, trees are added to the ensemble incrementally in the training phase to correct for any residuals occuring in existing decision trees. These residuals are the negative gradients that quantify the amount of variability between a prediction and the expected outcome of the predictive function based on the independent variables of a single training instance. Adding trees one at a time allows each new tree to be specifically trained to improve an already trained ensemble, as opposed to the RF process of randomly adding nodes to new trees to provide for a better average.

1. Situation dependent, multi-model blending

One, two, three, and four-hour ahead forecasts were generated for all 12 months at all seven SURFRAD sites. The developed code was run 336 times to model each unique forecasting situation. Each run trained all four ML algorithms on pre-processed data for the desired month from the years 2004-2008 and 2010-2014. After the models were built they were tested on unseen data from 2009 and forecasts were made for the desired forecast horizon.

1. Validation metrics

A suite of validation metrics is used to compare the forecast accuracy of different methodologies and situations in this study. A thorough discussion of different validation metrics is covered by Zhang et al. for comparing N observed GHI values with the N forecast values

.26 Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are commonly used metrics that measure the difference between the forecasted and actual GHI values. The RMSE metric is commonly used to evaluate the overall accuracy of forecasts and penalizes large forecasting errors with its square order. The MAE metric is also appropriate for evaluating errors through the entire forecasting period, and is widely used in regression problems and by the renewable energy industry. It does not penalize large forecast errors as much as the RMSE metric. Smaller values of these validation metrics indicate a higher forecasting accuracy.

1. RESULTS & DISCUSSION
2. ML forecasts vs. benchmark methods

To calibrate this study against existing literature, forecasts were made for the same one, two, three, and four-hour ahead forecast horizons as Perez et al. study.27 Their study tested the period from August 23, 2008 through August 31, 2009. Limitations arose in this study from the algorithms’ inabilities to predict on the years before 2009, those years consisting of three-minute data resolution, which caused this study to use a slightly different testing period. Instead, this study made predictions over the period from January 1, 2009 to December 31, 2009. Seasons were partitioned into four three-month periods beginning with January 1st through March 31st.

Results are compared to both Perez’s forecasts of a slightly different time period and to the persistence of cloudiness forecasts made for the same January 1, 2009 through December 31, 2009 time period.

Table 1 in Appendix B shows the RMSE values for ML predictions made in this study, Perez’s forecasts, and the persistence of cloudiness forecasts. The values in the yellow columns in Table 1 are composed by taking the best performing machine learning algorithm per month and compiling these into seasonal and yearly results. The ML models employed in this study outperformed the persistence of cloudiness forecasts in every situation with average RMSE values of 92.36 W/m2 and 122.12 W/m2, respectively. This study outperformed Perez’s forecasting methodology, which was tested on a different period, with average RMSE values of

92.36 W/m2 and 108.29 W/m2, respectively.

Comparing the performance of forecasting methods is also discussed using the relative frequency (rounded) that a given technique had in producing the lowest error. This study outperformed Perez’s method on a seasonal basis for one and four-hour ahead forecasts in 86% and 57% of tests, respectively, based on RMSE values. However, their results outperformed this study for two and three-hour ahead forecasts in 57% and 61% of forecasts, respectively. The Perez forecasts also outperformed this study in 68% of winter and 57% of spring seasonal forecasts, while the ML models outperformed the Perez forecasts in 75% and 79% of all situations for the summer and fall seasons, respectively. When broken down by geographic location, this study outperformed Perez et al. in Boulder, Fort Peck, Desert Rock, and Bondville with respective relative frequencies of 75%, 94%, 63%, and 53% across all tests. Their study outperformed this study when forecasting across all situations in Goodwin Creek, Penn State, and Sioux Falls 75%, 56%, and 56% of the time. These relative frequencies only take into account the number of times that one method outperformed the other and do not consider the margin of difference, in W/m2, between competing forecasts.

Relative frequencies are useful to show which method works best in individual forecasting situations but are not ideal for assessing a method’s overall ability to minimize forecasting errors across all situations. Figure 1 in Appendix C compares the ML forecasts’ RMSE values with the two benchmark forecasts, Perez and persistence, for all SURFRAD sites. The graphs show how forecasting errors tend to increase as the forecast horizon extends in time. The relative strength of the Perez two and three-hour ahead forecasts over the ML models is especially apparent in the graphs for Bondville, Goodwin Creek, and Penn State. Comparisons are also made by showing the percentage improvement, defined as the difference between the RMSE of the ML forecast and the RMSE of the benchmark forecast divided by the RMSE of the benchmark. The greatest improvement across all situations occurs in Fort Peck where the suite of ML algorithms demonstrates a 28.8% improvement over the Perez forecasts’ average RMSE values, followed by a 25.7% improvement in Boulder. Improvements over Perez forecasts’s RMSE averages are made in Desert Rock, Sioux Falls, Penn State, Bondville, and Goodwin Creek by 21.5%, 10%, 6.4%, 4.8%, and 0.7%, respectively. It is interesting to note that this study shows the largest improvements in RMSE scores for Boulder, Fort Peck, and Desert Rock. These

three sites are located at the highest elevations and are the three westernmost locations in the SURFRAD network. ML forecasts outperform the RMSE results from the persistence of cloudiness forecasts for all sites as well. They show the greatest improvement in the four locations situated eastward of mountain ranges or other orographic features: Boulder (25.3%), Desert Rock (30.7%), Penn State (26.3%), and Fort Peck (27.2%).

1. Performance of ML algorithms against each other

Table 2 in Appendix B compares the four ML models used in this study against each other by showing the relative frequency (rounded) of each algorithm’s ability in producing the lowest RMSE values in the listed forecasting situations. There were 84 forecasting situations for each of the four forecast horizons (12 months per seven sites), 84 situations for the seasonal forecasts (three months per seven sites for four forecast horizons), and 48 situations for the geographic forecasts (12 months per four forecast horizons) in this study. The ANN algorithm was the top performer in each of these situational categories and produced the lowest error value in 41.1% of the 336 forecasting situations, based on the RMSE metric. The SVM algorithm performed equally well in Fort Peck, and when the seasons are broken down by month, SVM outperformed ANN during all April forecasts by 42% to 32%. The RF and SVM algorithms performed almost equally well when considering all forecast situations.

Table 3 in Appendix B is similar to Table 2 except that it shows the relative frequency of each model’s ability to produce the lowest MAE values in each forecasting situation. The SVM algorithm produced the lowest MAE values in all types of forecasting situations more often than any of the other ML models, though it tied ANN when making three-hour ahead forecasts. It was the top performer in more situations according to the MAE metric than the ANN was when considering the RMSE metric. The SVM performed best most often in one-hour forecasts and approached lower relative frequencies as the forecast horizon extended in time.

1. CONCLUSIONS & FUTURE WORK

This paper assessed the performance of machine learning techniques and their validity in improving short-term solar forecasting. The machine learning approach was compared to other forecasting methods, and individual machine learning algorithms were compared against each other. ML forecasts generated lower average RMSE values than a cloud motion forecasting method for all seven sites, with the biggest improvements for the three sites at the highest elevations and westernmost locations in the SURFRAD network. They also outperformed persistence of cloudiness forecasts at all seven sites, with the greatest improvements at the four locations situated downwind from large orographic features. The ML forecasts had the lowest RMSE more often than the cloud motion method across all summer and fall seasonal forecasts, as well as for one and four-hour ahead forecasts. Assessing the performance of the four algorithms against each other did not reveal any strong situation dependent sensitivities, as each algorithm was capable of making the best forecast in the various forecasting situations though some less than others. However, either SVMs or ANNs most often led to the lowest forecasting

errors depending on the error metric used. ANN was the preferred algorithm if minimizing the largest point forecast errors is the greatest concern, according to the RMSE metric. However, SVM was the best performer if minimizing the average absolute difference, MAE, is top priority.

This solar irradiance forecasting methodology can be extended by increasing the forecast horizon resolutions from hourly increments to five-minute increments, allowing for more dynamic time series information about upcoming ramping events. Further work is needed to fine- tune each ML algorithm, and future research should also look into optimizing ML hyper- parameters for each situation dependent forecast. Improved forecasts will help facilitate higher penetrations of solar energy into the grid by providing increased grid reliability and minimizing costs associated with ramping events.

1. ACKNOWLEDGEMENTS

I would like to thank Anthony Florita, Tarek Elgindy, and Bri-Mathias Hodge for their support and guidance throughout this project. I would also like to thank the workforce development team at NREL, the CCI program, and the Office of Science. This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Community College Internships Program (CCI).

1. APPENDICES Appendix A: Equations

RMSE is defined as

MAE is defined as

Equation 3

1 𝑁 − () Equation 4



𝑁



1

𝑁

𝑁

i!1

( − ( )2

i!1

for comparing N observed GHI values with the N forecast values .

Appendix B: Tables

Table 1: Yearly and Seasonal RMSE Metric Summary

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Forecast Horizon | Boulder | | | Bondville | | | Goodwin  Creek | | | Fort Peck | | | Desert Rock | | | Penn State | | | Sioux Falls | | |
| ML | PC | PZ | ML | PC | PZ | ML | PC | PZ | ML | PC | PZ | ML | PC | PZ | ML | PC | PZ | ML | PC | PZ |
| ALL YEAR | 1 hour | 74 | 104 | 120 | 62 | 83 | 85 | 71 | 96 | 80 | 56 | 79 | 94 | 52 | 76 | 80 | 67 | 96 | 86 | 52 | 74 | 68 |
| 2 hour | 108 | 142 | 139 | 98 | 118 | 98 | 103 | 130 | 101 | 81 | 110 | 106 | 72 | 103 | 88 | 97 | 132 | 99 | 81 | 106 | 84 |
| 3 hour | 123 | 161 | 154 | 116 | 135 | 112 | 125 | 146 | 114 | 94 | 126 | 123 | 83 | 116 | 96 | 114 | 151 | 113 | 96 | 126 | 102 |
| 4 hour | 125 | 169 | 166 | 121 | 143 | 122 | 120 | 152 | 127 | 93 | 130 | 132 | 82 | 122 | 104 | 117 | 157 | 124 | 103 | 136 | 115 |
| WINTER | 1 hour | 55 | 74 | 64 | 51 | 66 | 60 | 58 | 87 | 48 | 36 | 53 | 107 | 45 | 66 | 46 | 53 | 72 | 57 | 41 | 62 | 48 |
| 2 hour | 81 | 98 | 71 | 82 | 104 | 66 | 98 | 128 | 59 | 52 | 74 | 105 | 63 | 92 | 48 | 79 | 102 | 57 | 65 | 96 | 58 |
| 3 hour | 96 | 113 | 81 | 104 | 117 | 74 | 122 | 146 | 66 | 62 | 84 | 109 | 75 | 106 | 59 | 91 | 122 | 59 | 82 | 117 | 69 |
| 4 hour | 87 | 119 | 85 | 105 | 123 | 81 | 111 | 147 | 70 | 58 | 84 | 112 | 84 | 107 | 70 | 96 | 127 | 65 | 89 | 122 | 78 |
| SPRING | 1 hour | 97 | 143 | 125 | 84 | 114 | 93 | 94 | 127 | 92 | 75 | 108 | 110 | 71 | 108 | 86 | 84 | 117 | 83 | 66 | 94 | 69 |
| 2 hour | 137 | 195 | 141 | 133 | 154 | 109 | 125 | 171 | 122 | 110 | 149 | 124 | 106 | 147 | 95 | 119 | 161 | 99 | 103 | 133 | 90 |
| 3 hour | 170 | 218 | 157 | 147 | 178 | 123 | 159 | 190 | 144 | 129 | 174 | 141 | 120 | 155 | 111 | 143 | 183 | 118 | 124 | 156 | 107 |
| 4 hour | 162 | 228 | 170 | 159 | 189 | 137 | 145 | 202 | 164 | 134 | 186 | 148 | 115 | 171 | 115 | 145 | 190 | 137 | 131 | 171 | 126 |
| SUMMER | 1 hour | 96 | 136 | 143 | 76 | 97 | 100 | 88 | 119 | 92 | 81 | 101 | 91 | 48 | 71 | 99 | 88 | 125 | 112 | 67 | 90 | 80 |
| 2 hour | 137 | 185 | 175 | 111 | 134 | 115 | 121 | 151 | 113 | 110 | 143 | 109 | 64 | 85 | 110 | 122 | 170 | 127 | 99 | 129 | 98 |
| 3 hour | 144 | 211 | 189 | 135 | 153 | 129 | 135 | 168 | 120 | 125 | 164 | 129 | 70 | 105 | 111 | 140 | 194 | 142 | 112 | 155 | 120 |
| 4 hour | 175 | 222 | 204 | 138 | 169 | 138 | 139 | 175 | 129 | 122 | 173 | 142 | 74 | 118 | 124 | 138 | 208 | 152 | 118 | 168 | 129 |
| FALL | 1 hour | 46 | 63 | 85 | 35 | 56 | 58 | 44 | 50 | 55 | 34 | 52 | 59 | 43 | 60 | 55 | 45 | 71 | 60 | 35 | 48 | 49 |
| 2 hour | 78 | 92 | 97 | 67 | 80 | 68 | 67 | 70 | 66 | 52 | 73 | 67 | 57 | 87 | 62 | 69 | 96 | 71 | 57 | 67 | 54 |
| 3 hour | 81 | 103 | 110 | 76 | 90 | 84 | 83 | 81 | 81 | 59 | 81 | 83 | 65 | 97 | 69 | 80 | 104 | 76 | 65 | 76 | 64 |
| 4 hour | 75 | 107 | 120 | 81 | 143 | 89 | 87 | 81 | 94 | 58 | 78 | 88 | 56 | 94 | 72 | 89 | 102 | 83 | 74 | 81 | 80 |

ML (yellow columns): Forecasts made by machine learning methods.

PC (white columns): Benchmark forecasts made by persistence of cloudiness method. PZ (blue columns): Benchmark forecasts made by the Perez et al. cloud motion method.

Table 2: Relative Frequencies of ML Models per RMSE Metric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Forecast Situation | RF | SVM | ANN | GBM |
| 1-hour ahead | 8% | 31% | **42%** | 19% |
| 2-hour ahead | 22% | 20% | **38%** | 20% |
| 3-hour ahead | 25% | 13% | **45%** | 16% |
| 4-hour ahead | 29% | 20% | **38%** | 13% |
| Winter | 20% | 22% | **38%** | 20% |
| Spring | 15% | 25% | **40%** | 20% |
| Summer | 24% | 17% | **47%** | 12% |
| Fall | 25% | 23% | **38%** | 14% |
| Boulder | 27% | 15% | **43%** | 15% |
| Bondville | 21% | 21% | **35%** | 23% |
| Goodwin Creek | 21% | 23% | **39%** | 17% |
| Fort Peck | 23% | **31%** | **31%** | 15% |
| Desert Rock | 15% | 31% | **42%** | 12% |
| Penn State | 25% | 15% | **45%** | 15% |
| Sioux Falls | 15% | 12% | **50%** | 23% |
| All Situations | 20.8% | 21.1% | **41.1%** | 17.0% |

Table 3: Relative Frequencies of ML Models per MAE Metric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Forecast Situation | RF | SVM | ANN | GBM |
| 1-hour ahead | 7% | **65%** | 17% | 11% |
| 2-hour ahead | 15% | **44%** | 26% | 15% |
| 3-hour ahead | 14% | **36%** | **36%** | 14% |
| 4-hour ahead | 23% | **32%** | 26% | 19% |
| Winter | 17% | **41%** | 25% | 17% |
| Spring | 6% | **60%** | 19% | 15% |
| Summer | 21% | **38%** | 25% | 16% |
| Fall | 12% | **40%** | 35% | 13% |
| Boulder | 10% | **48%** | 25% | 17% |
| Bondville | 8% | **52%** | 17% | 13% |
| Goodwin Creek | 23% | **35%** | 19% | 23% |
| Fort Peck | 21% | **42%** | 27% | 10% |
| Desert Rock | 21% | **48%** | 19% | 12% |
| Penn State | 8% | **42%** | 40% | 10% |
| Sioux Falls | 10% | **44%** | 27% | 19% |
| All Situations | 14.6% | **44.3%** | 26.2% | 14.9% |

Appendix C: Figures

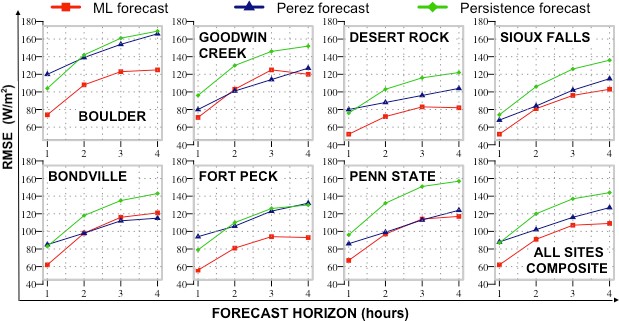


Figure 1: ML and benchmark methods’ performances forecasting different forecast horizons.

**DATA ANALYTIC**

Topic : Solar Pannel Forecasting

Team leader: M.Annapoorani

Team Members:

1.S.Abukalam azad

2.P.Prasanna

3.T.Sibiraj

4.Balakumaran

Assingnment1-The growth of supermarkets in most populated cities I increasing and market competitions are also high

Assignment2-The cognos HR scorecard:Measuring success in Talent

Management

Assignmen3-Autofotune:predicting automobile prices

Forecasting of total daily solar energy generation

using ARIMA: A case study

Sharif Atique

Dept. of Electrical and Computer Engineering

Texas Tech University

Lubbock, TX, USA

taufique.atique@ttu.edu

Subrina Noureen

Dept. of Electrical and Computer Engineering

Texas Tech University

Lubbock, TX, USA

subrina.noureen@ttu.edu

Vishwajit Roy

Dept. of Electrical and Computer Engineering

Texas Tech University

Lubbock, TX, USA

vishwajit.roy@ttu.edu

Vinitha Subburaj

Dept. of Computer Science

Texas Tech University

Canyon, TX, USA

vsubburaj@wtamu.edu

Stephen Bayne

Dept. of Electrical and Computer Engineering

Texas Tech University

Lubbock, TX, USA

stephen.bayne@ttu.edu

Joshua Macfie

Electrical Engineer

Group NIRE

Lubbock, TX, USA

joshua.macfie@groupnire.com

Abstract—In this paper, a well known statistical modeling

method named ARIMA has been used to forecast the total daily

solar energy generated by a solar panel located in a research

facility. The beauty of the ARIMA model lies in its simplicity and

it can only be applied to stationary time series. So our time series

data, which is seasonal and non-stationary, is transformed into

a stationary one for applying the ARIMA model. The model is

developed using sophisticated statistical techniques. The optimum

model is chosen and validated using Akaike information criterion

(AIC) and residual sum of squares (SSE). Error analysis is done to

demonstrate the efficiency of the proposed method. The accuracy

of the developed model can be further increased, which is subject

to future research.

Keywords—Forecasting, solar, ARIMA, time series, stationarity,

generation

I. INTRODUCTION

The electricity demand in the world is always increasing.

However, the traditional source of fossil fuel is limited and they

leave significant carbon footprint. These factors, along with the

technological advancements, have driven the increasing usage

of distributed renewable energy resources [?]. Penetration of

renewable energy resources are only going to increase as the

grid is becoming smarter. Among all the renewable resources,

photovoltaic (PV) based solar energy is the most promising

[?]. However, like any other renewable resources, solar energy

is inherently uncertain as it is heavily dependent on solar

irradiance and other environmental factors like humidity, temperature and geographic location [?]. As a result, forecasting

plays a significant role in PV based systems for operation

and planning purposes [?]. Accurate solar energy generation

forecast can help with mitigating the uncertainty and result in

better demand side management [?]. As the amount of solar

power generation is uncertain, it can be modeled as a stochastic

time series model [?].

Time series forecasting method is particularly useful when

there is little knowledge about the effects of explanatory

variable on the output. In a time series model, a dependent

variable or output is dependent only on its past values. After

a model has been established, it is then used to predict the

future values [?]. Time series methods are well studied in the

forecasting area and it is continually being improved. One of

the most well studied models of this arena is ARIMA, acronym

for Autoregressive Integrated Moving Average. The reasons

behind the popularity is simplicity of implementation and use

of the famous Box-Jenkins methodology [?]. This simplicity

is a result of the linear correlation assumption between time

series values of the past and present. This is a major drawback

of ARIMA model even though it can model various types of

time series data. As complex real world time series data is not

always linear, ARIMA models might not be the best solution.

In cases like this, there are other statistical models that can

be used to incorporate the non-linearity. In spite of all this,

ARIMA model is good benchmark for solar energy forecasting.

In this paper, ARIMA models, both the seasonal and nonseasonal variations, have been studied to predict the daily total

solar energy generation of a 10kW solar panel. This solar

panel is installed in the rooftop of Group Nire building in

the Reese Reseach Center located in Lubbock, TX. The time

series data is transformed to a stationary one, analyzed for

determining the model parameters and validated using various

criteria like Akaike Information Criterion (AIC) and sum of

square of residuals (SSE). Finally, the performance of the

model is judged by necessary error analysis.

II. ARIMA MODELING

A. General Formulation

The value of a dependent variable is expressed as a linear

relationship between past values of the dependent variable and

random errors in ARIMA model. However, a time series can

only be modeled as a ARIMA process if it is stationary. As

strong stationarity is somewhat complex to demonstrate [?],

in this paper we would assume stationarity if the time series

is weakly stationary. In general terms, a weakly stationary

time series has constant statistical properties, namely mean

and variance [?]. Transformation operations like differencing,

978-1-7281-0554-3/19/$31.00 ©2019 IEEE logging and deflating [?] are performed on a non-stationary

time series to make it stationary. There are two different

variations of ARIMA models: non-seasonal and seasonal. If

there is seasonality in the time series data, then seasonal

ARIMA model is used. Otherwise, the non-seasonal ARIMA

model is used for the general cases.

The non-seasonal ARIMA is modeled in the following way

[?]:

yˆt = µ+φ1yt−1 +· · ·+φpyt−p −θ1et−1 − · · · −θqet−q, (1)

where yˆt is the d

th difference of a non-stationary time series

Y. The order of autoregressive lag terms, differencing and

moving average lag terms are represented by p,d and q, respectively. The autoregressive and moving average parameters

are expressed with φ and θ terms, respectively. Finally, µ is a

constant.

Depending on the values of p,d and q, an ARIMA process

can undertake the form of purely moving average (MA),

purely autoregressive (AR) or autoregressive moving average

(ARMA) processes.

Presence of a periodic pattern in the time series is called

seasonality. Seasonality in a time series is expressed by its

span, S. For example, monthly solar energy generation has

higher values in summer months, so S = 12 in this case.

ARIMA models can be used to forecast seasonal time series

data, just like non-seasonal time series data.

A multiplicative model, including both the non-seasonal

and seasonal fluctuations, is used to represent seasonal ARIMA

model. Seasonal ARIMA is generally expressed in the following way [?]:

ARIMA(p, d, q) ∗ (P, D, Q)S, (2)

where

• P = seasonal AR order

• Q = seasonal MA order

• D = seasonal differencing order

• S = span of pattern in seasonality

A more formal representation of seasonal ARIMA model

is as follows:

(1 − φ1B − · · · − φpB

p

)(1 − Φ1B

S − · · · − ΦP B

P S)(1 − B)

d

(1−B

S

)

Dyt = (1+θ1B+· · ·+θqB

q

)(1+Θ1B

S+· · ·+ΘQB

QS)t,

(3)

where B is the backshift operator, whose operation is governed

by 4:

B

myt = yt−m (4)

B. Model Parameter Selection

The first step in the modeling process is checking for the

stationarity of the time series. A rough estimate of stationarity

can be graphically obtained by plotting the partial auto correlation function (PACF)and auto correlation function (ACF)

plots of the time series. ACF measures the correlation of a

time series value with other values of the same time series

at different lags. PACF also measures the correlation between

a value of a time series and another value at different lag.

However, PACF ignores the other values at different lags while

calculating the correlation for a particular lag value [?]. If the

ACF doesn’t display any significant value after a few lags or

the PACF contains a sharp cutoff after the initial value [?], then

we have a stationary time series on our hand. However, most

real life problems are not as straightforward and stationary.

After the initial estimation, a more methodical approach,

named Augmented Dickey Fuller (ADF) test, is executed to

confirm stationarity [?], [?], [?]. ADF is also known as unit

root test. If there is no unit root of the characteristic equation,

then the time series is stationary. Otherwise, the time series in

non-stationary.

The general equation for testing stationarity using the ADF

test is as follows:

∂Yr = µ + βt + ρYt−1 + ∂1Yt−1 + · · · + ∂pYt−p + et. (5)

Here, β represents the trend. Moreover, et represents a sequence of independent normal random variables of zero mean

and unit variance. Then hypothesis is formulated in the following way [?]:

NullHypothesis : H0 :| ρ |= 0(Non − stationarity)

AlternativeHypothesis : H1 :| ρ |6= 0(Stationarity)

Rejection or acceptance of the null hypothesis is dictated by the

p-value. A confidence level of 95% is assumed in this work. If

p ≥ 0.05, the time series is non-stationary (null hypothesis is

true). Otherwise, the time series is stationary (null hypothesis

is rejected)

C. Model Selection and Validation

After the initial checking of stationarity, differencing operation is performed in case the time series is non-stationary. If the

initial time series is stationary, then the order of differencing,

d = 0. Differencing would be performed as long as the time

series isn’t transformed into a stationary one. In this work,

other transformation techniques are not studied. After each

differencing operation, the stationarity can be checked using

the ACF and PACF plots or ADF test or both. Finally, the

PACF and ACF plots of the derived stationary time series

would determine the p and q parameters. p and q generally

correspond to significant terms in PACF and ACF plots, respectively. However, they might not always be the optimum model

parameters. The seasonal parameters can also be determined

from the ACF and PACF plots.

The final step before forecasting is selection of the optimum ARIMA model. The following criteria are commonly

used to estimate the goodness of fit for the developed models:

1) Akaike Information Criterion (AIC)

2) Corrected Akaike Information Criterion (AICc)

3) Bayesian Information Criterion (BIC)

4) Residual sum of squares (SSE)

1) AIC: The formulation of AIC ([?], [?]) is as follows:

AIC = −2log(maximumlikelihood) + 2k, (6)

where k is independently adjusted number of parameters.

2) AICc: The formulation of AICc [?] is as follows:

AICc = −2log(maximumlikelihood) + n + k

n − k − 2

, (7)

where n is total number of data points.

3) BIC: The formulation of BIC ([?]) is as follows:

BIC = −2log(maximumlikelihood) + klogn

n

, (8)

where k and n are the same as defined in AIC and AICc.

The preferred model is the one that minimizes all these

criteria. In this work, AIC and SSE have been used for

optimum model selection.

III. DATA PREPARATION

We have used total daily solar energy generation (in kWh)

as our dependent variable. The data has been collected for a

complete year (6

th November, 2017 - 5

th November, 2018)

from the 10kW solar plant located in the rooftop of the Group

Nire building in Reese Research Center, Lubbock, TX. The

data was initially stored in .csv format. The data was read

and plotted as a time series (1) using the prominent statistical

software R.

As evident from 1, there are a few missing data points,

15 to be exact, as the solar panel wasn’t functional on those

days. So, this data was processed for filling up the missing

data using the tsclean() function in R and plotted in 2. This

time series is eventually utilized for the analysis in this paper.

IV. ANALYSIS

An initial assumption about stationarity of our data set can

be made just by looking at 2. Two trends, one upward and one

downward, can be guessed from this figure. However, decision

about stationarity is made after plotting of the necessary

autocorrelation functions and performing ADF test.

The ACF and PACF of the cleaned time series are plotted

in 3 and 4, respectively.

In 3, ACF of the time series data has been plotted. It is

evident from this figure that the ACF doesn’t become insignificant after a few lags. In fact, the ACF remains signifiation

even after 50 lags (5). There is also some periodicity in the

ACF plot, which implies seasonality. In the PACF plot in 4,

there is not sharp cutoff. So the graphical test confirm nonstationarity of the time series. However, the ADF test still need

to be performed to be certain about the conclusion on the nonstationarity of the time series. The result of the ADF test is

summarized in 6.

As we can see from 6, the p-value is 0.6639. So the null

hypothesis can’t be rejected and the time series can be declared

as non-stationary.

If a time series has inherent trend and seasonality, then the

time series is always non-stationary as they imply systemic

variation in mean and variance. The seasonal component, trendFig. 5: ACF plot of the original time series data upto 50 lags

component and the residuals of the time series is plotted in 7.

We can see from this figure that both the trend and seasonality are present in our data set. So necessary transformation

techniques need to be carried out in order to make our data a

stationary one. As previously mentioned, only the differencing

operation is covered in this paper.

Fig. 7: Trend and seasonality decomposition of the time series

A first order differencing operation is performed on the

time series and the differenced time series is plotted in 8. The

differenced time series looks stationary in the first look as no

systemic variation in mean and variance is readily evident. This

claim is further enhanced by both graphical and mathematical

analyses.

As we see from 9, ACF becomes insignificant after 2

lags, if we neglect the sparse significant ACFs at higher lags.

Partial autocorrelation functions of the differenced time series

mostly become insignificant after 5 lags (10), if the sparse

significant values at higher lags are ignored. These two plots

indicate possible stationarity of the differenced time series,

which is confirmed using ADF test subsequently. Moreover,

these significant lags from the ACF and PACF plots help us

with initial assumption of AR and MA orders in the ARIMA

model.

Finally, the ADF test is performed on the differenced time

series data and the result is summarized in 11. The calculated

p-value is 0.01, which confirms the rejection of null hypothesis

and subsequent stationarity. So our desired stationary time

series is produced with first order differencing operation of

the actual time series data that was collected and cleaned.

TABLE I: Performance comparison of seasonal and nonseasonal model using auto.arima() routine

TABLE II: Performance comparison of ARIMA models

p d q P D Q S AIC SSE p-value

0 1 1 1 0 1 30 2500.358 18002.47 0.004

0 1 2 0 0 1 30 2508.908 20450.01 0.99

0 1 2 1 0 1 30 2480.266 16983.22 0.99

1 1 1 1 0 1 30 2481.902 17067.95 0.89

1 1 2 1 0 1 30 2482.236 17076.8 0.99

V. MODEL VALIDATION

The span of seasonality is not apparent from figures 9 and 10. So, an initial auto.arima() routine is applied on the time series to obtain the optimum seasonal

periodicity. The auto.arima() routine yielded the model

ARIMA(0, 1, 2)(0, 0, 2)30. Other periodicity was randomly

chosen and tested against 30. However, periodicity of 30

performed better than the other values in terms of minimized

AIC. The auto.arima() routine is also performed without the

seasonality option. However, the seasonal model outperforms

the non-seasonal one in terms of all the information criteria,

which should be the case as our time series has seasonality in

it. So the further analyses in this work will be solely focused

on the seasonal model. The result is summarized in I.

The approximate non-seasonal AR and MA orders are

resembled by significant terms in PACF and ACF plots,

respectively. So, the AR and MA orders in this work should

be approximately 5 and 2, as evidenced by 10 and 9. However,

higher orders in the model bring increased cost and complexity.

In order to ensure simplicity and reduced cost, the nonseasonal AR and MA orders are limited to 2 and seasonal

AR and MA orders are limited to 1. So the constraints are:

0 ≤ p, q ≤ 2 (9)

0 ≤ P, Q ≤ 1 (10)

Arima models are simulated based on these constraints and

the results of the 5 best models are summarized in II. Results

from all the models are not presented due to space constraint.

The p-value, obtained by performing Ljung-Box test [?] on

the residuals, is needed to reject the hypothesis that there is no

autocorrelation among the model residuals. So p-value needs

to be more than 0.05 for a 95% significance level. From II, it is

obvious that ARIMA(0, 1, 2)(1, 0, 1)30 model outperforms all

the other models both in terms of AIC and SSE. The residual

analysis of this model is presented in 12. It is obvious that

there is no significant correlation between the residuals and

the residuals mostly follow the normal distribution, except for

the lower tail, which proves the rationale of our model.

So, our model equation has the form:

(1−Φ1B

30)(1−B)yt = µ+ (1+θ1B +θ2B

2

)(1+Θ1B

30)t.

(11)

The relevant parameter values are summarized in III. Only

Fig. 12: Residual analysis of ARIMA(0, 1, 2)(1, 0, 1)30 Model

TABLE III: Model parameter values

Parameter Estimated value p-value

Φ1 0.6543 0

θ1 -0.5319 0

θ2 -0.2578 0

Θ1 -1 0

µ 0.0201 0.6116

the constant µ has an associated p-value of greater than 0.05,

which makes it insignificant. So the constant value has been

disregarded in the final simplified forecasting equation in 12.

yt = yt−1 + Φ1yt−30 − Φ1yt−31 + t + θ1t−1

+ θ2t−2 + Θ1t−30 + θ2Θ1t−31 + θ2Θ1t−32. (12)

Finally, equation 12 is used to forecast the value of total daily

sonar energy generation for a particular day. In this work, the

forecasted values for the last 30 days in our dataset have been

compared with the actual values and demonstrated in 13.

Fig. 13: Forecasted vs actual values for the last 30 days

After forecasting the values, the accuracy of the model is

tested using R. The mean absolute percentage error (MAPE)

for our model is 17.70%.

VI. CONCLUSION AND FUTURE WORK

In this work, a model has been established and tested

for predicting the total daily solar energy generation of a

research facility using the popular and simple statistical time

series method called ARIMA. The model is developed using

techniques like differencing, ACF, PACF and ADF test. The

model is validated using AIC and SSE. The MAPE is slightly

high, however this doesn’t necessarily indicate an issue in

the modeling process. There might be certain factors affecting

the accuracy of the developed model which should be further

investigated. Upon inspecting the original time series, visible

fluctuations in the last 30 day period can be spotted. This

volatility might be tackled better using something like moving

average of solar outputs, instead of daily data. Better smoothing techniques would be studied in future research. Although

a stationary time series was obtained using a differencing

operation. The variance still displayed possible heteroscedasticity. So forecasting of the time series data using models

like generalized autoregressive conditional heteroscedasticity

(GARCH) and autoregressive conditional heteroscedasticity

(ARCH) would also be studied in future. Finally, this work

can be used as a good building block for further research into

forecasting of renewable energy generation.

REFERENCES

[1] I. Khan, H. Zhu, J. Yao, and D. Khan, “Photovoltaic power forecasting

based on elman neural network software engineering method,” in

2017 8th IEEE International Conference on Software Engineering and

Service Science (ICSESS), Nov 2017, pp. 747–750.

[2] I. Majumder, M. K. Behera, and N. Nayak, “Solar power forecasting

using a hybrid emd-elm method,” in 2017 International Conference on

Circuit ,Power and Computing Technologies (ICCPCT), April 2017, pp.

1–6.

[3] M. Z. Hassan, M. E. K. Ali, A. B. M. S. Ali, and J. Kumar, “Forecasting

day-ahead solar radiation using machine learning approach,” in 2017

4th Asia-Pacific World Congress on Computer Science and Engineering

(APWC on CSE), Dec 2017, pp. 252–258.

[4] V. P. Singh, V. Vijay, M. S. Bhatt, and D. K. Chaturvedi, “Generalized

neural network methodology for short term solar power forecasting,”

in 2013 13th International Conference on Environment and Electrical

Engineering (EEEIC), Nov 2013, pp. 58–62.

[5] J. Wu and C. K. Chan, “The prediction of monthly average solar

radiation with tdnn and arima,” in 2012 11th International Conference

on Machine Learning and Applications, vol. 2, Dec 2012, pp. 469–474.

[6] G. Zhang, “Time series forecasting using a hybrid arima and neural network model,” Neurocomputing,

vol. 50, pp. 159 – 175, 2003. [Online]. Available:

http://www.sciencedirect.com/science/article/pii/S0925231201007020

[7] G. E. P. Box and G. Jenkins, Time Series Analysis, Forecasting and

Control. Holden-Day, San Francisco, CA, 1970.

[8] D. E. Myers, “To be or not to be... stationary? that is the question,”

Mathematical Geology, vol. 21, no. 3, pp. 347–362, Apr 1989.

[Online]. Available: https://doi.org/10.1007/BF00893695

[9] M. Poulos and S. Papavlasopoulos, “Automatic stationary detection

of time series using auto-correlation coefficients and lvq — neural

network,” in IISA 2013, July 2013, pp. 1–4.

[10] “Introduction to arima: nonseasonal models,”

https://people.duke.edu/ rnau/411arim.htm, accessed: 2018-10-08.

[11] R. Shumway and D. Stoffer, Time Series Analysis and

Its Applications: With R Examples, ser. Springer Texts in

Statistics. Springer New York, 2010. [Online]. Available:

https://books.google.com/books?id=dbS5IQ8P5gYC

[12] J. H. F. Flores, P. M. Engel, and R. C. Pinto, “Autocorrelation and

partial autocorrelation functions to improve neural networks models

on univariate time series forecasting,” in The 2012 International Joint

Conference on Neural Networks (IJCNN), June 2012, pp. 1–8.

[13] R. Hyndman and G. Athanasopoulos, Forecasting: Principles and

Practice, 2nd ed. Australia: OTexts, 2018.

[14] D. Dickey and W. A. Fuller, “Distribution of the estimators for autoregressive time series with a unit root,” Journal of the Americal Statistical

Association, vol. 74, no. 366, pp. 427–431, 1979.

[15] S. Halim, I. N. Bisono, Melissa, and C. Thia, “Automatic seasonal

auto regressive moving average models and unit root test detection,”

in 2007 IEEE International Conference on Industrial Engineering and

Engineering Management, Dec 2007, pp. 1129–1133.

[16] J. Wu and C. K. Chan, “The prediction of monthly average solar

radiation with tdnn and arima,” in 2012 11th International Conference

on Machine Learning and Applications, vol. 2, Dec 2012, pp. 469–474.

[17] H. Akaike, “A new look at the statistical model identification,” IEEE

Transactions on Automatic Control, vol. 19, no. 6, pp. 716–723,

December 1974.

[18] S. Halim, I. N. Bisono, Melissa, and C. Thia, “Automatic seasonal

auto regressive moving average models and unit root test detection,”

in 2007 IEEE International Conference on Industrial Engineering and

Engineering Management, Dec 2007, pp. 1129–1133.

[19] G. Schwarz, “Estimating the dimension of a model,” Ann. Statist.,

vol. 6, no. 2, pp. 461–464, 03 1978. [Online]. Available:

https://doi.org/10.1214/aos/1176344136

[20] G. E. P. Box and D. A. Pierce, “Distribution of residual autocorrelations in autoregressive-integrated moving average time series models,”

Journal of the American Statistical Association, vol. 65, no. 332, pp.

1509–1526, 1970.