

Project Report

# Solar Irradiance Forecasting

---

Ankush Phulia

Manish Kumar

10th May, 2018

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
<b>Objective</b>	<b>2</b>
<b>Theoretical Background</b>	<b>3</b>
Time Series Models	3
Artificial Intelligence/Machine Learning Models	3
Artificial Neural Networks (ANNs)	3
Support Vector Machines (SVMs)	4
Ensembles	4
<b>Methodology</b>	<b>5</b>
Data Analysis	5
Studying Trends	5
Statistical Analysis	5
Data Preparation	6
Cleaning Data - Flags	6
Resampling Data - Timescale	6
Varying Input Parameters	7
Models Trained	7
Time Series - ARIMA	7
AI/ML Models	7
ML Model Ensembles	7
<b>Results</b>	<b>8</b>
Forecasting with ARIMA	8
Forecasting Daily DNI, Input = Past 30 days DNI	9
Forecasting Daily DNI, Input = Past 5 Years DNI on that Date	10
Forecasting Hourly DNI, Input = Past 5 Years DNI	11
Forecasting Daily DNI, Input = Past 5 Years DNI + GHI on that Date	12
Forecasting Hourly DNI, Input = Past 5 Years DNI + GHI	13
<b>Conclusion</b>	<b>14</b>
<b>Future Work</b>	<b>14</b>
<b>Sources</b>	<b>15</b>



## Introduction

Solar power production is totally dependent on weather condition, clear sky is the best condition for any solar power production plant because maximum possible direct normal radiation (has significant intensity for power production) is absorbed by solar panels on a clear sky day while during bad weather large drop in power production is experienced. This gives birth to the need of solar irradiance forecasting, big investments are done to get the reliable forecast.

The need for reliable forecasting arises from the variable nature of the solar resource, seasonal deviations in generation and load profiles, the high cost of energy storage, and industry requirements that must balance grid flexibility with reliability. The problem at hand is of great complexity but a number of promising approaches have been developed in the past few years. Some of those approaches have been used in an attempt of producing reliable forecast.

## Objective

To forecast Direct Normal Irradiance for different time scales using different stochastic methods.

## Theoretical Background

### Time Series Models

Time series modeling is a dynamic research area which has attracted attentions of researchers community over last few decades. The main aim of time series modeling is to carefully collect and rigorously study the past observations of a time series to develop an appropriate model which describes the inherent structure of the series. This model is then used to generate future values for the series, i.e. to make forecasts. Time series forecasting thus can be termed as the act of predicting the future by understanding the past.


Due to the indispensable importance of time series forecasting in numerous practical fields such as business, economics, finance, science and engineering, etc, proper care should be taken to fit an adequate model to the underlying time series. It is obvious that a successful time series forecasting depends on an appropriate model fitting.

A lot of efforts have been done by researchers over many years for the development of efficient models to improve the forecasting accuracy. As a result, various important time series forecasting models have been evolved in literature. One of the most popular and frequently used stochastic time series models is the Autoregressive Integrated Moving Average (ARIMA) model. The basic assumption made to implement this model is that the considered time series is linear and follows a particular known statistical distribution, such as the normal distribution. ARIMA has subclasses of other models, such as the Autoregressive (AR), Moving Average (MA) and Autoregressive Moving Average (ARMA) models.

### Artificial Intelligence/Machine Learning Models

#### Artificial Neural Networks (ANNs)

Recently, ANNs have attracted increasing attentions in the domain of time series forecasting. Although initially biologically inspired, but later on ANNs have been successfully applied in many different areas, especially for forecasting and classification purposes. The excellent feature of ANNs, when applied to time series forecasting problems is their inherent capability of non-linear modeling, without any presumption about the statistical distribution followed by the observations. The appropriate model is adaptively formed based on the given data. Due to this reason, ANNs are data-driven and self-adaptive by nature. During the past few years a substantial amount of research works have been carried out towards the application of neural networks for time series



modeling and forecasting. There are various ANN forecasting models in literature. The most common and popular among them are the multi-layer perceptrons (MLPs), which are characterized by a single hidden layer Feed-Forward Network (FNN).

### Support Vector Machines (SVMs)

A major breakthrough in the area of time series forecasting occurred with the development of Vapnik's SVM concept. The initial aim of SVM was to solve pattern classification problems but afterwards they have been widely applied in many other fields such as function estimation, regression, signal processing and time series prediction problems. The remarkable characteristic of SVM is that it is not only destined for good classification but also intended for a better generalization of the training data. For this reason the SVM methodology has become one of the well-known techniques, especially for time series forecasting problems in recent years.

The objective of SVM is to use the structural risk minimization (SRM) principle to find a decision rule with good generalization capacity. In SVM, the solution to a particular problem only depends upon a subset of the training data points, which are termed as the support vectors. Another important feature of SVM is that here the training is equivalent to solving a linearly constrained quadratic optimization problem. So the solution obtained by applying SVM method is always unique and globally optimal, unlike the other traditional stochastic or neural network methods. Perhaps the most amazing property of SVM is that the quality and complexity of the solution can be independently controlled, irrespective of the dimension of the input space.

### Ensembles

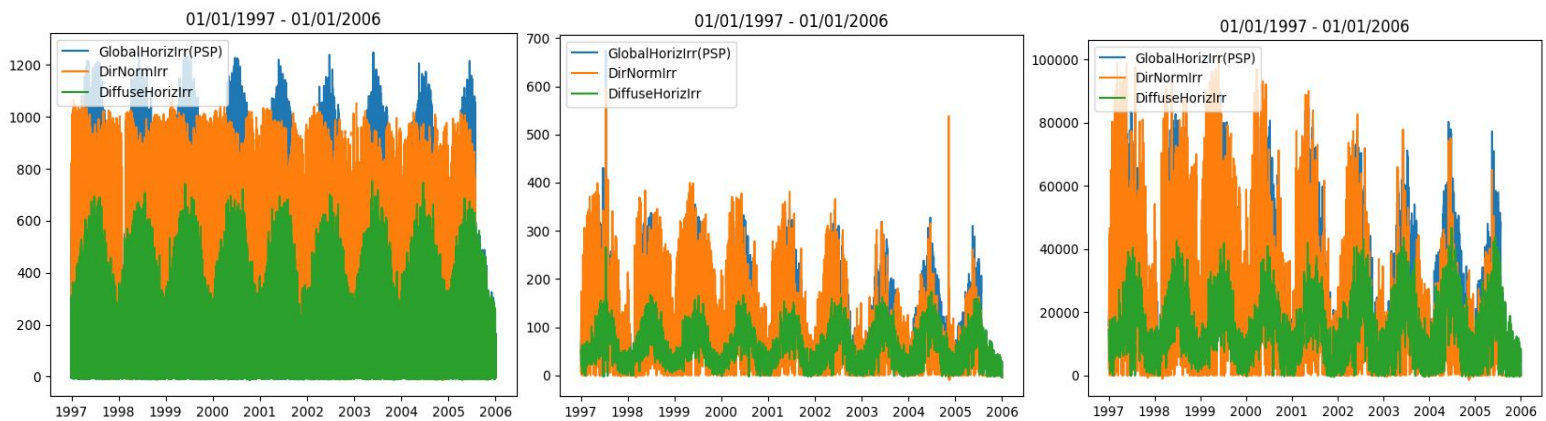
Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods

## Methodology

### Data Analysis

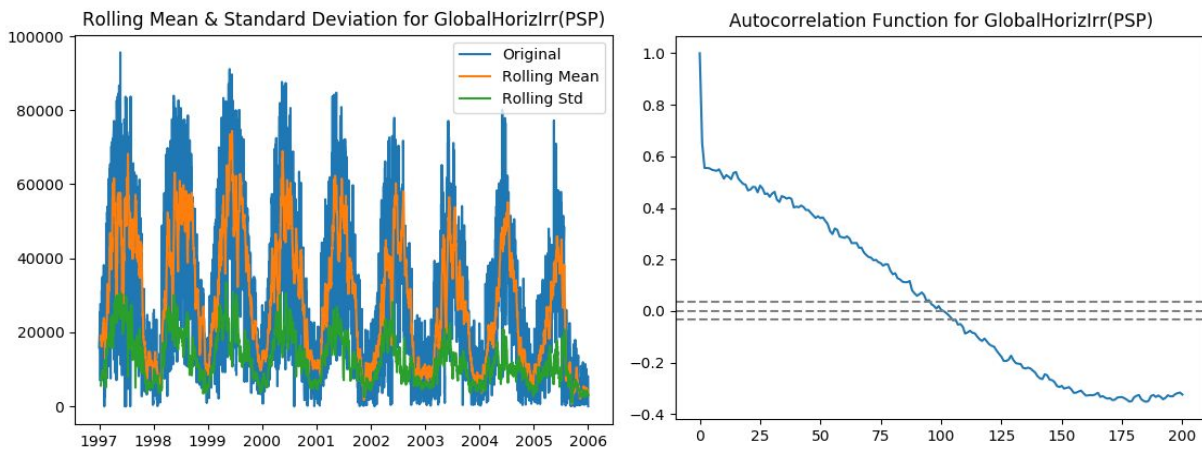
Initially the data is plotted. The plots are inspected visually and statistically for insight into it. The data for all the years is collated into one file

### Studying Trends



The plots above are raw data, daily average and daily sum for the years 1997-2006, the years where data passed visual inspection. Specifically, the data for about an year was missing for the chosen site(bluestate), after mid-2006, and the data following it had many extreme values. Thus data only upto 2006 was considered for the following experiments.

### Statistical Analysis



Statistics for daily summed GHI

Further, for each GHI, DNI (and DHI), statistical inspections like the rolling mean, rolling variance and autocorrelation plots can be seen, as well as tests like the Dickey-Fuller tests for stationarity of the time series. This kind of analysis gives some idea about the series and allows easier application of time series methods like ARIMA.

Daily Sum Correlation	GHI	DNI	DHI
GHI	1.0	0.828443	0.662883
DNI	0.828443	1.0	0.177654
DHI	0.662883	0.177654	1.0

Correlations between the quantities, Daily Summed

## Data Preparation

Perhaps the most important/time-consuming part, that needs the most code to be written for - preparing the data for input into the models

### Cleaning Data - Flags

The data provided has a flag associated with each measurement, which is an indicator of how good/reliable that measurement is, if at all. The flagging in the given data is by “SERI\_QC”, wherein a flag value of 1, 2 and 3 are the number of component tests passed by the point (n component test means that any n out of GHI, DNI and DHI are within their accepted bounds), and a flag value of 6 indicates that all component tests are passed and  $GHI = DNI + DHI$ , within 3%. All other values of flags indicate the issue with the error, and for this study, are simply ignored

Incidentally, there was not a single data point flagged 6. Speaks a lot about our data quality!

### Resampling Data - Timescale

The crucial step when solving a problem as open as solar resource forecasting is to decide the timescale of the forecasts, as well as the timescale of the data. In this study, various combinations have been tried - daily DNI forecasts on the basis of DNI of past 30 days, daily DNI forecasts on the basis of the same date in the past 5 years (e.g.. forecast DNI for May 5, 2006 based on May 5, 2005, 2004, ..., 2001), as well as forecasting Hourly DNI, based on same time(hour) in the past 5 years. It is a simple extension to do forecasts for Hourly DNI based on previous n days, or previous n years, or any combination of the two.

## Varying Input Parameters

It is DNI that is being forecast in this study always. But to put things into perspective, DNI and GHI both could be predicted, based on some past data for DNI, GHI or both. In this study, past DNI and past DNI + past GHI are tried. The intuition behind trying past GHI as an input was simply the observation that GHI data quality was better, so DNI forecasts might be improved using that. It helped that there is high correlation between DNI and GHI

## Models Trained

### Time Series - ARIMA

ARIMA models were trained on past DNI to forecast DNI. The parameters that worked best were  $(p, d, q) = (4, 1, 4)$ , i.e. 4 terms of auto-regression, 1 level of differencing(integration) and 4 terms for moving average, AR - I - MA. Initially ARIMA was tried out on smaller sets of data, to get an idea about its functioning

### AI/ML Models

Focus was on ANNs and SVMs for this study. Specifically Multi-Layer Perceptrons(MLP) were used in ANN, and SVMs with linear and cubic kernels were used in SVM. *scikit-learn* library available in python was used.

### ML Model Ensembles

A variety of models, available out-of-box in *scikit-learn* library of python, that use decision trees as the base regressor were trained - GradientBoost Regressor, Random Forests and Extra Random Tree Forests. Additionally the AdaBoost Regressor was also tried out.

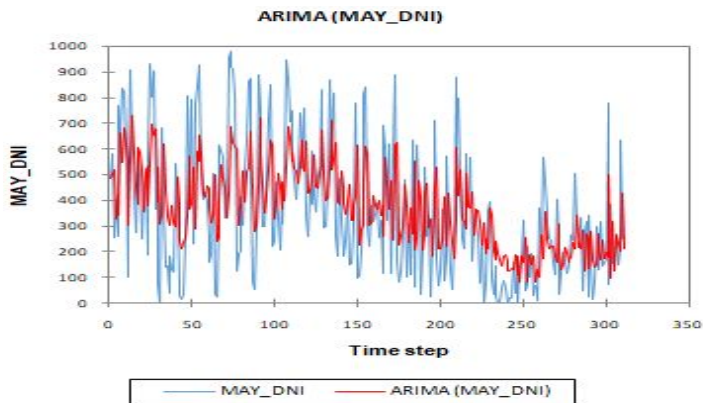
For AI/ML and Ensemble models, aside from trying out various input parameters, there was requirement to tune the models with appropriate parameter. This requires running the models multiple times, and to save some effort, *grid-search* was used. In grid search, given n hyperparameters for a model, an n-dimensional grid is created, with each point on the grid representing a set of parameters. Once the grid is formed, one can perform a search on this grid, and choose the point/set of parameters, for which the model does the best (usually on a validation set, separate from training and testing set). This entire process was automated to an extent, and coded up.



## Results

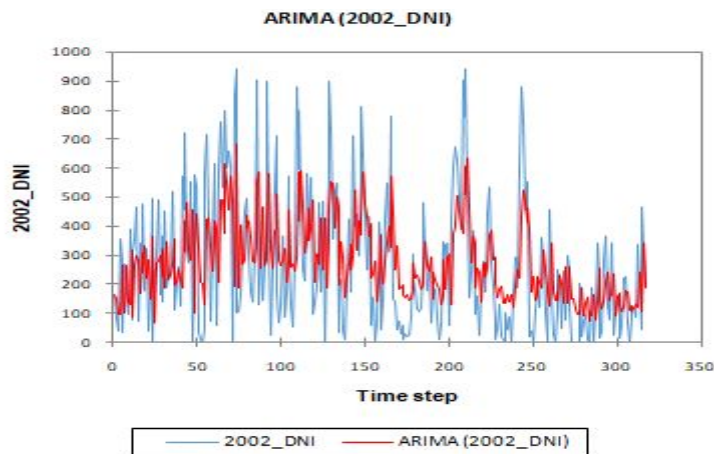
### Forecasting with ARIMA

Input data = past May months, Output = Daily DNI



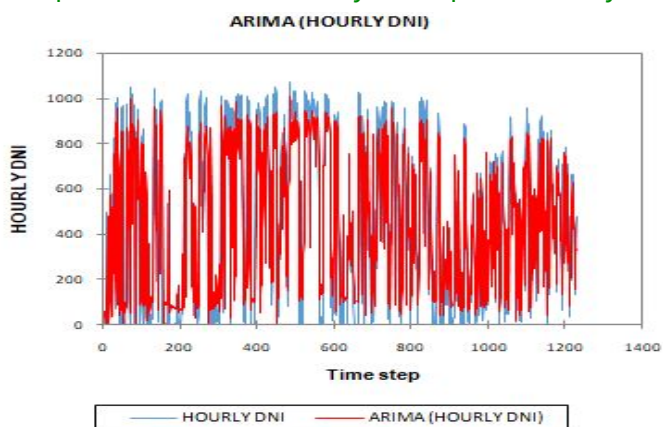
Observations	310
SSE	13624091
MSE	43948.68
RMSE	209.6394

Input Data = all of 2002, Output = Daily DNI



Observations	316
SSE	12419301
MSE	39301.59
RMSE	198.2463

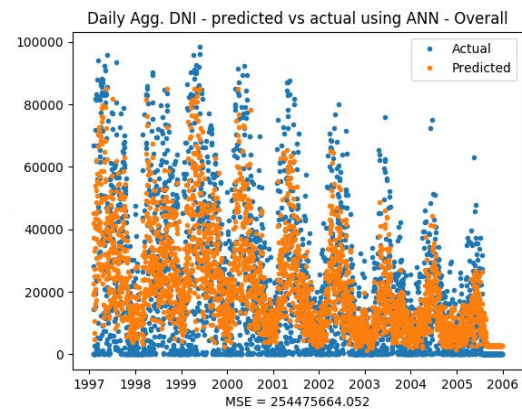
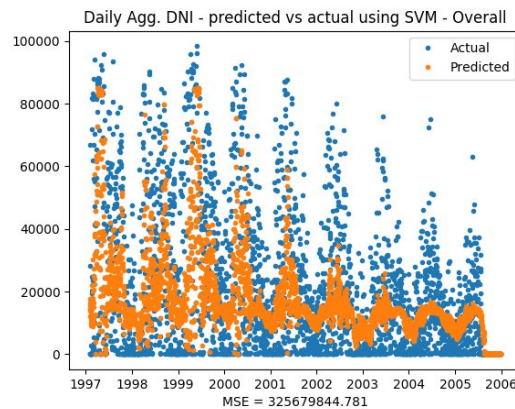
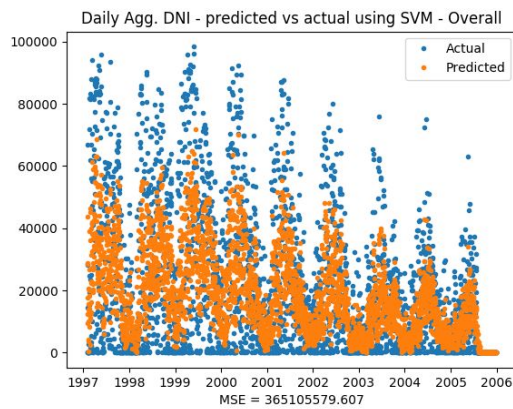
Input Data = Past 15 Days, Output = Hourly DNI



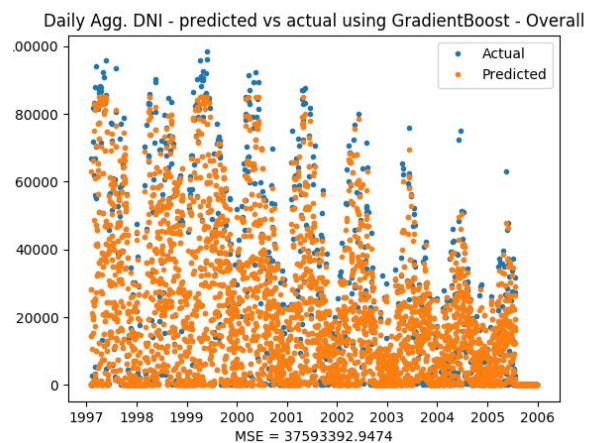
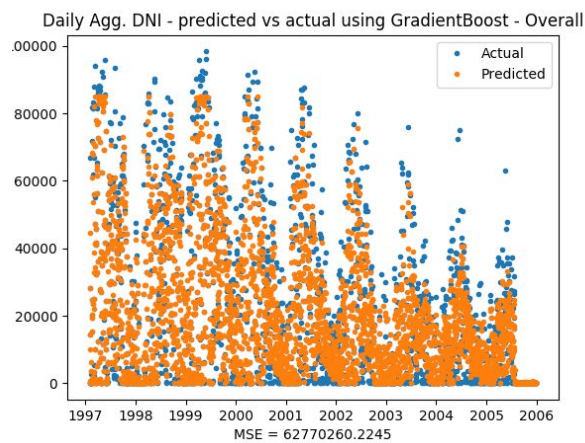
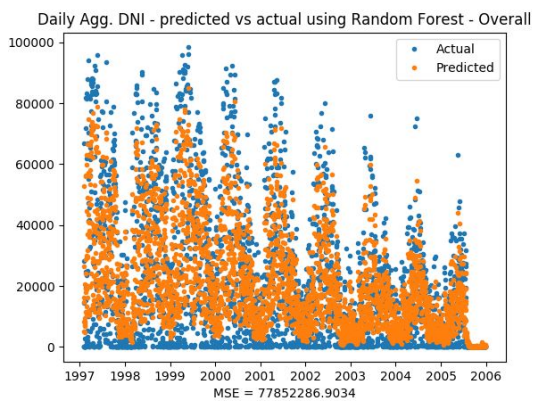
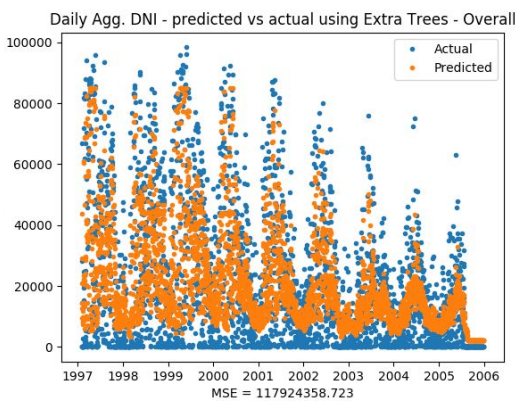
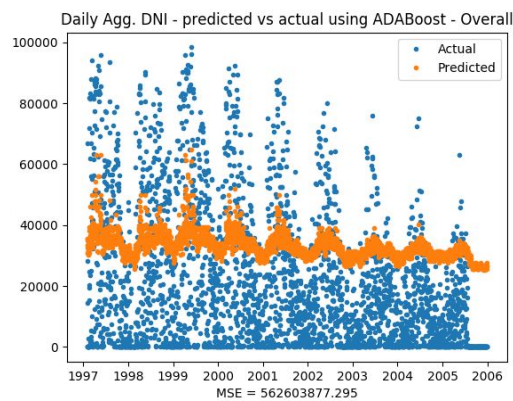
Observations	1227
SSE	50616224
MSE	41252.02
RMSE	203.1059

## Forecasting Daily DNI, Input = Past 30 days DNI

### SVMs/ANNs



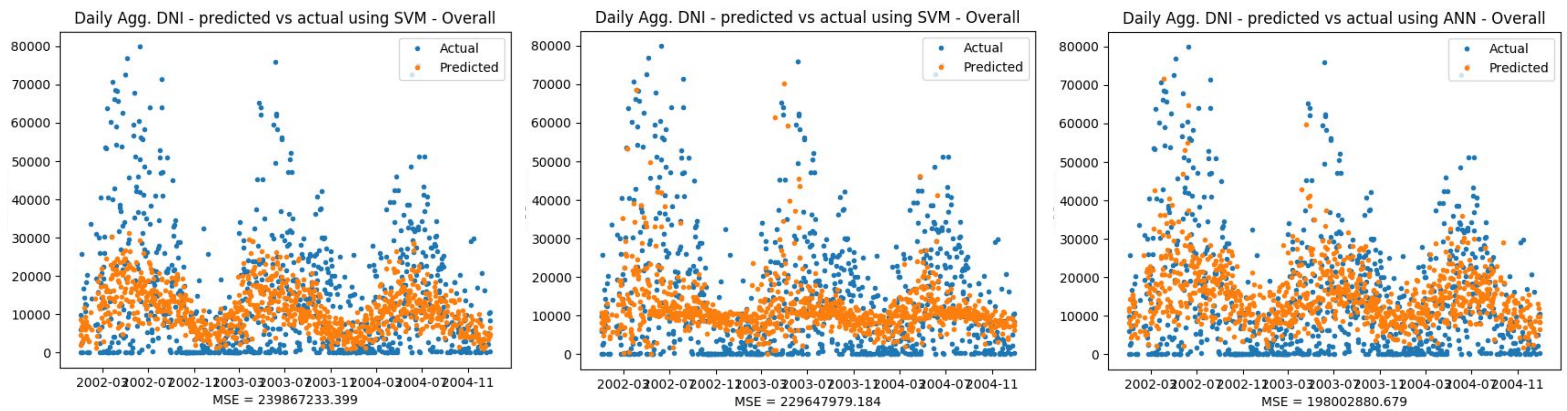
### Ensembles



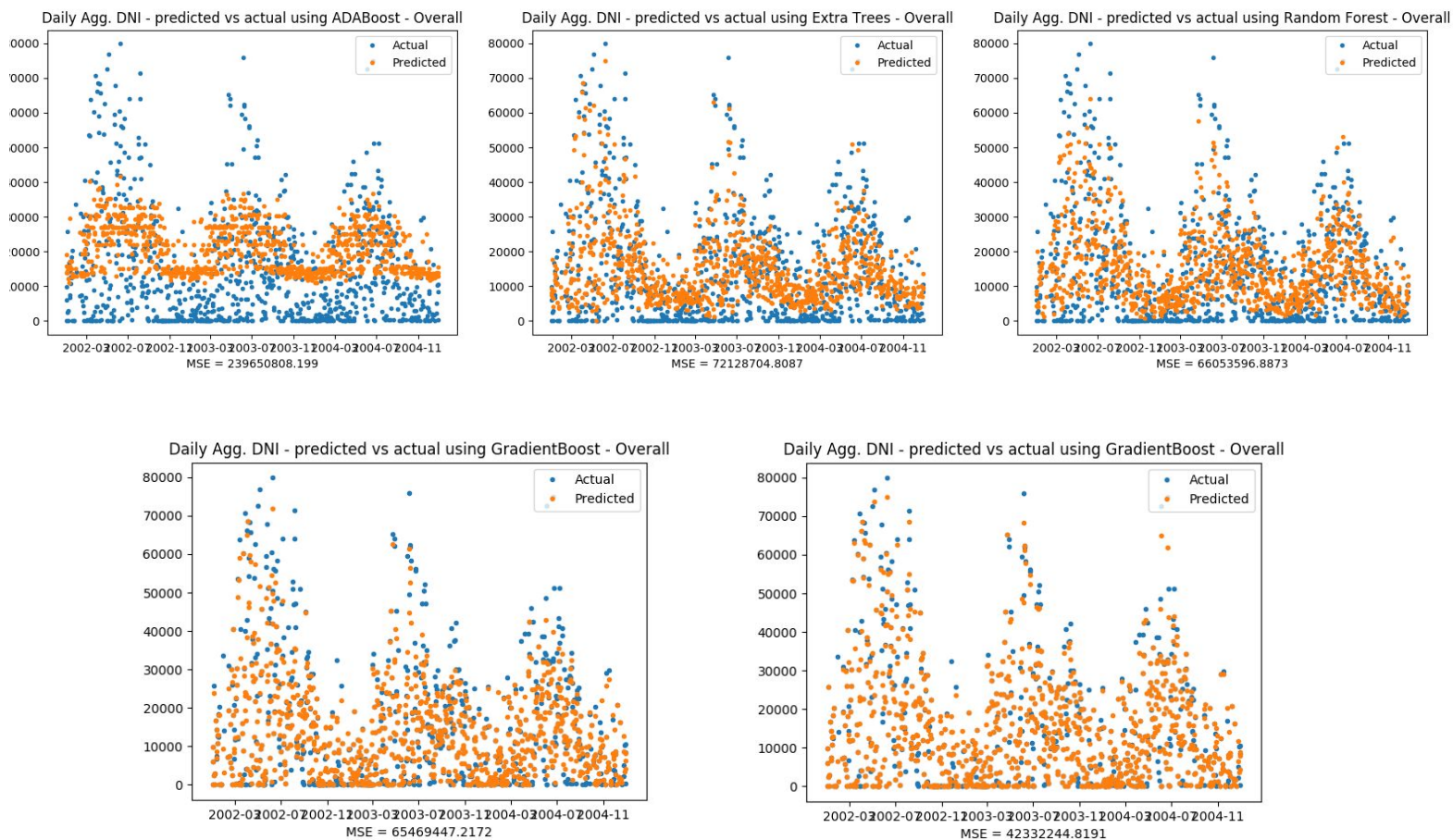


## Forecasting Daily DNI, Input = Past 5 Years DNI on that Date

### SVMs/ANNs

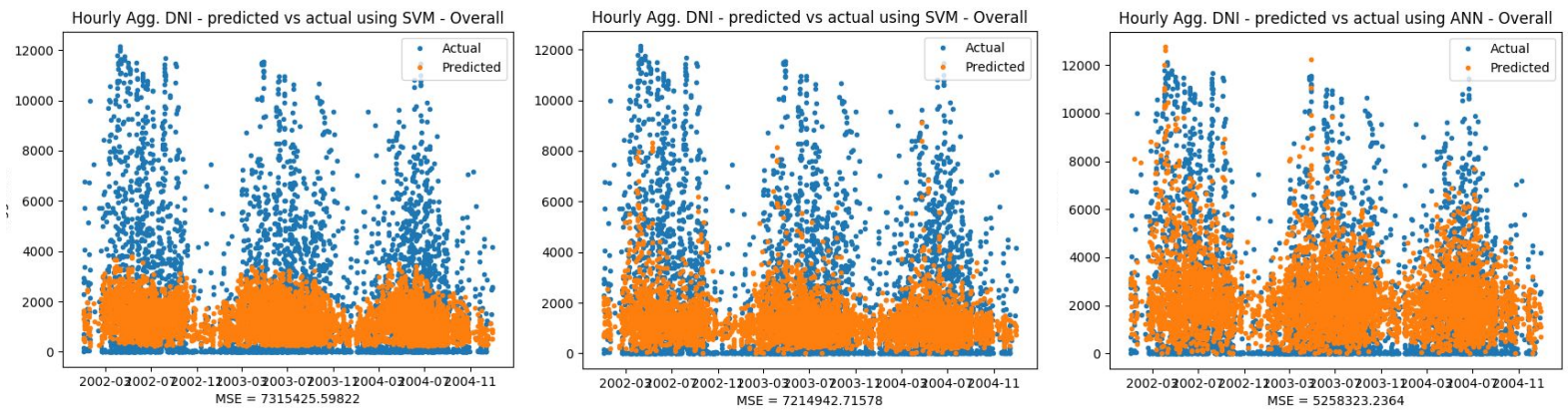


### Ensembles

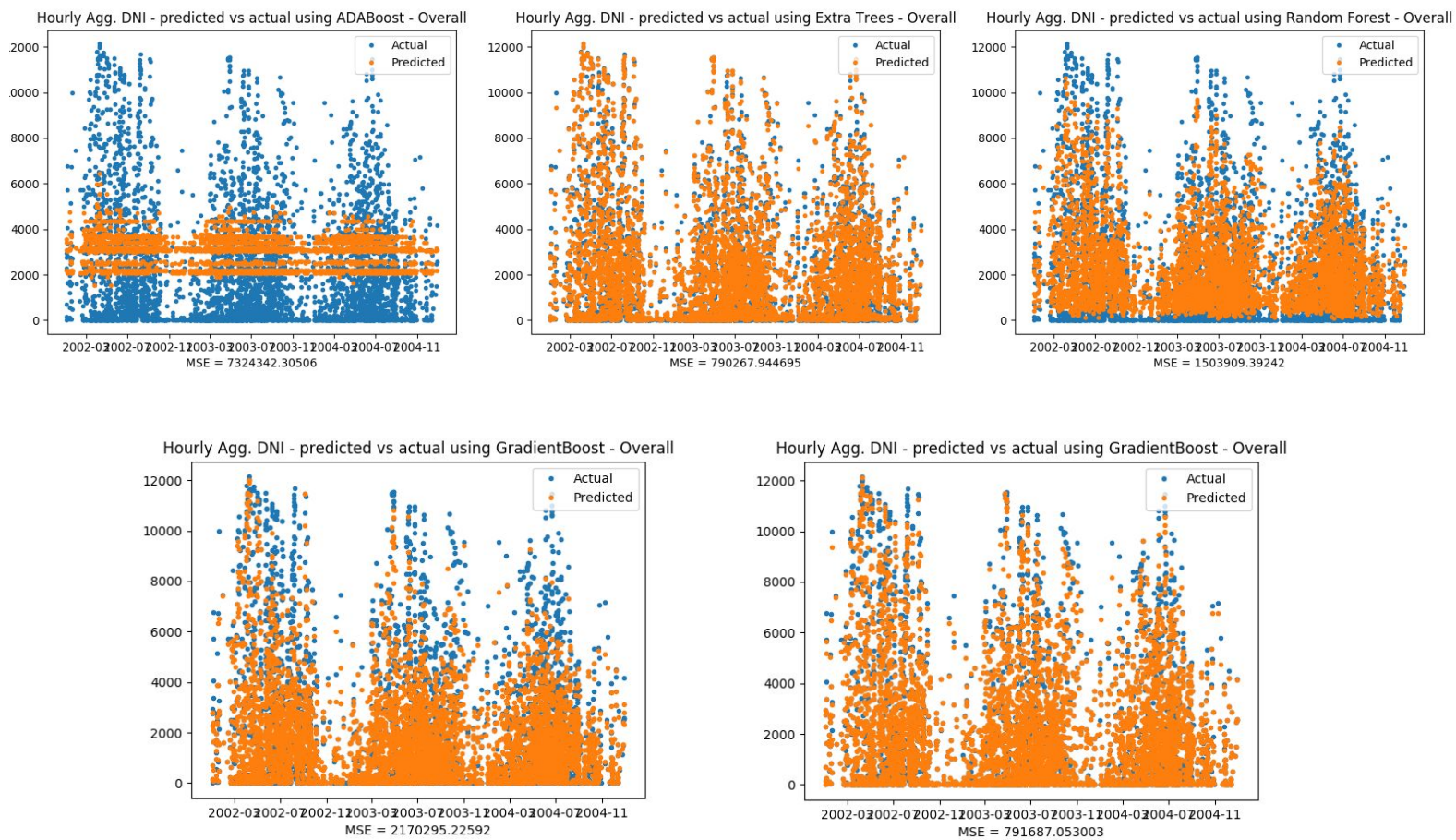


## Forecasting Hourly DNI, Input = Past 5 Years DNI

### SVMs/ANNs



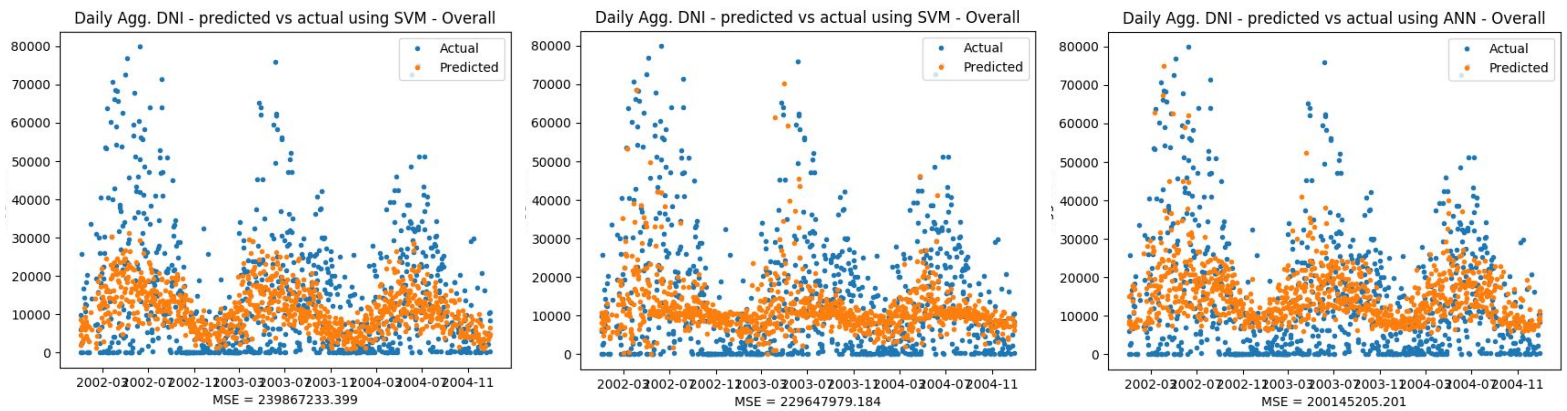
### Ensembles



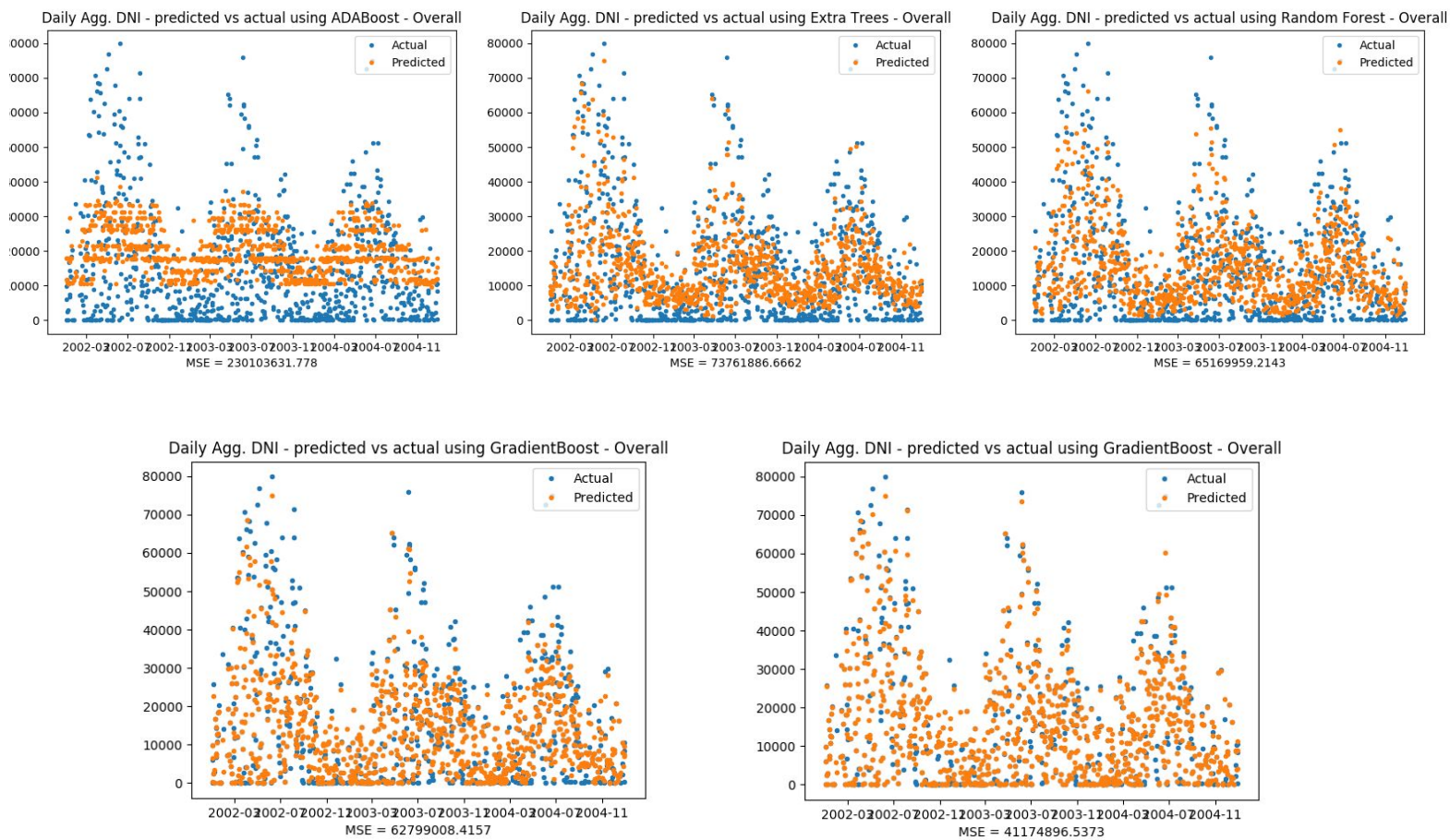


## Forecasting Daily DNI, Input = Past 5 Years DNI + GHI on that Date

### SVMs/ANNs

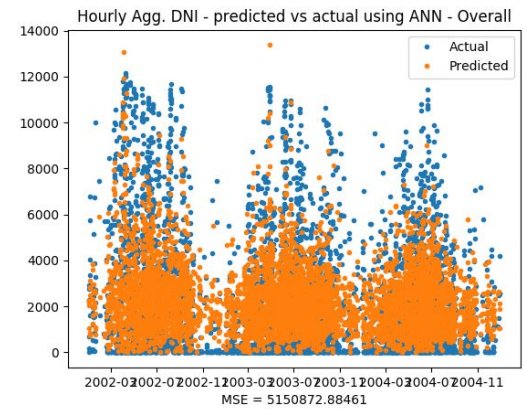
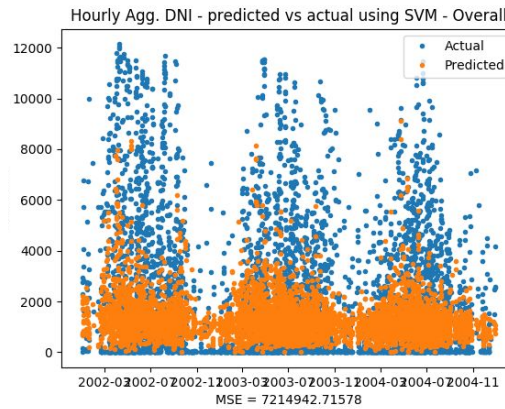
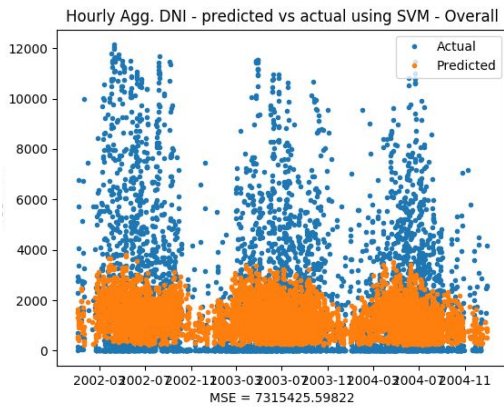


### Ensembles

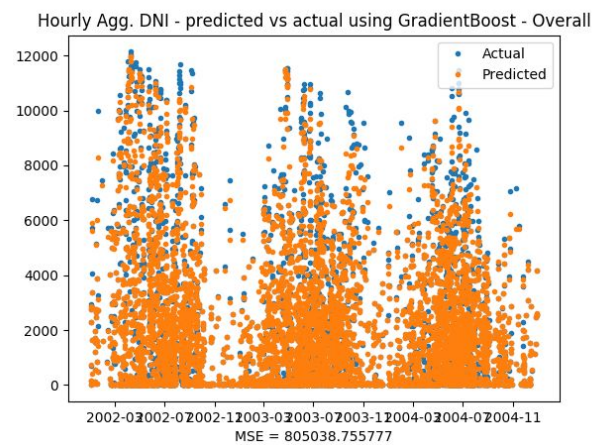
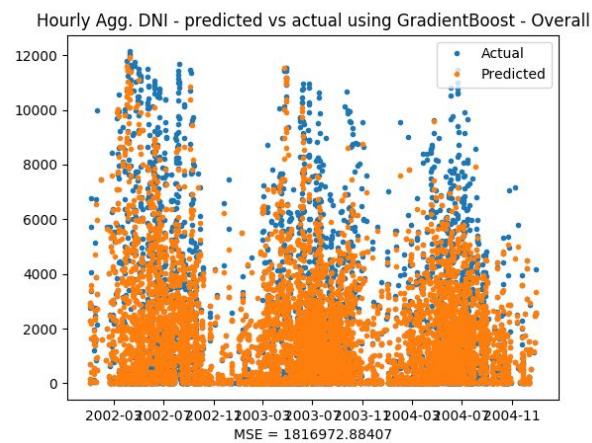
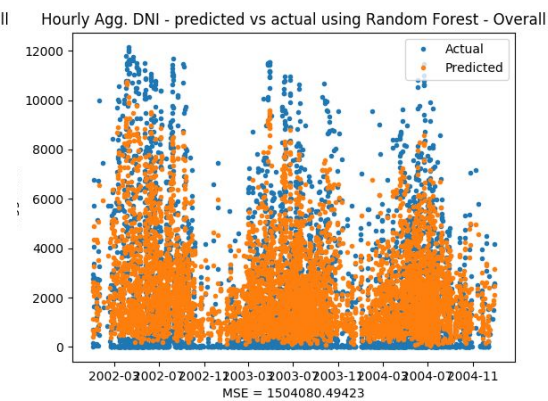
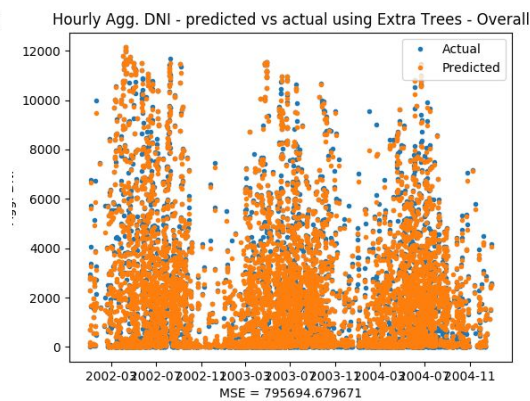
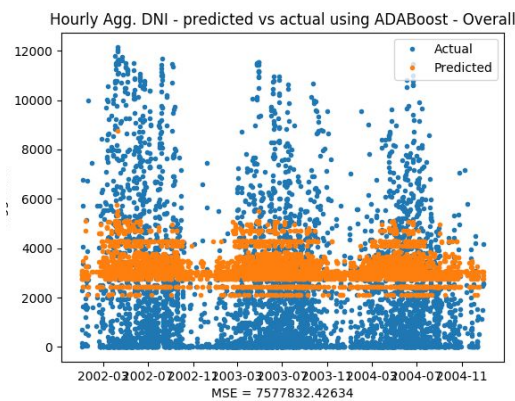


## Forecasting Hourly DNI, Input = Past 5 Years DNI + GHI

### SVMs/ANNs



### Ensembles





## Conclusion

From the above results we can see that ensemble methods perform quite a bit better than time series or simple AI/ML methods like SVMs or ANNs. Really, the choice of model is dependent on the problem in question, one should choose the right tool for the right job.

Time Series methods are relatively easy to tune, and can work with a few data points to fit a model and make predictions. Further, since they are quick to train, training for rolling windows is pretty simple. AI/ML models on the other hand, are heavily data driven, and their making no assumptions about the model (unlike the linear models of time series methods), comes at a cost of needing more training data and time for fitting models.

Ensembles work the best for forecasting DNI, especially **GradientBoost** and ExtraTrees methods. However, they require the most amount of data to train, as well as the most amount of parameter tuning. They are best used when there is a large amount, of (relatively) good data available.

Perhaps the most important thing is to have really good data!

## Future Work

Despite getting some good looking fits graph-wise, the MSE is still very high. This is especially true for Daily Summed forecasts, which is quite possibly due to large input values. A reason for this can be the quality and the amount of data - lots of data can potentially mean lots of poor quality data (as evidenced by the large number of *near zero* values in the graphs in many cases). Ways to make the predictions more robust, including things like bias correction of the data must be explored.

Presently, for AI/ML methods, very few iterations (~1000) are being run, increasing these will increase the training time, but will likely improve performance.

In our current work, the error metric is simply the MSE. The next step should be characterising errors for each method - calculating the percentage of days when the error exceeds a threshold, and looking into rectifying that, by averaging/ensembles, etc.

A further avenue to explore is the spatial correlation of values. This might prove to be useful to forecast for a site whose prior data may not be available, but historically is shown to be highly correlated with the sites whose data is available. It can also be useful to make more robust predictions, in case the past data for the site for which we are predicting is of poor quality.



## Code

Python source code, data used and graphs available publicly on [GitHub](#)

## Sources

1. Cooperative Network for Renewable Resource Measurements (CONFRRM) for Data
2. “An Introductory Study on Time Series Modeling and Forecasting” - R. Adhikari, R. K. Agrawal

