Comparison of Machine Learning Models for Short Term Solar Power Output Forecasting at Loomis Chaffee

Student: Derek Yuan
Advisor: Dr. Mark D. LeBlanc

January 2021

Abstract: *The Loomis Chaffee School's solar field is the largest solar power plant among any K-12 school in Connecticut. Only recently completed in late 2019, it has and will continue to provide a significant portion of the school's electricity, forming a core part of Loomis' sustainability initiatives. Like all solar fields, its power generation is highly variable and dependent on external factors such as weather. For such photovoltaic power sources, reliable and successful integration into the larger power grid system depends upon knowledge and prediction of future power output. The application of machine learning models for solar power output forecasting has thus become popular in research and literature, replacing past approaches based on statistical or physical models. This project aims to determine the feasibility and performance of applying three machine learning algorithms, support vector machines; random forests; and k-nearest neighbors, for short term solar power output forecasting, through leveraging the rich data generated by Loomis' solar array in the first comprehensive study of that data. Only in situ data is used, and findings indicate that the training of the three models on the limited set of data features has promising utility in power prediction, with a significant improvement over a baseline persistence method for hourly prediction and best performance by support vector machines. Ideas to further their performance may include augmentation of solar plant data with local weather or time series specific data preprocessing.*

1. Introduction

Climate change has prompted the world to undergo a shift in electricity production from fossil fuels to renewable energy [29]. In late 2019, the Loomis Chaffee school actively became a part of that transition with the inauguration and dedication of a 1 megawatt solar array on its campus, the largest of its kind for any K-12 school in Connecticut. After two years of construction, the solar field provides twenty-five percent of the school's electricity needs, offsetting over 800 tons of CO2 [1]. A power purchase agreement (PPA) defines the relationship between Loomis Chaffee and the solar plant construction company Encon [2], where Encon will own the system and pay for construction and maintenance and Loomis will purchase electricity from Encon for twenty years.

This important contribution to Loomis Chaffee's sustainability objectives was initiated four years ago by Jason Liu, Loomis class of 2017, through a year-long independent project that culminated in a feasibility study on solar energy for the school [1]. Efforts promote student involvement with the solar plant and its operations, such as through inclusion in content in environmental science classes and construction of a hiking trail surrounding the array. Presenting the solar field's data and making it more accessible and informative to the community is another objective, and this project is an early example of such an endeavor.

Knowledge and insight in the field of solar energy and power generation data was gathered through meetings with domain experts and industry professionals related to two companies, PowerFactors [3] and AlsoEnergy [4], both of which conduct data analysis and perform asset management of Loomis' solar array. A meeting with Steve Hanawalt, the founder of PowerFactors, focused on learning about how his company is applying machine learning to the data they gather from solar energy fields. They have recently deployed such algorithms in fault detection and diagnosis and hope to extend its applications to power generation forecasting and prediction. Another meeting was held with Katherine Crider, the Director of Asset Management at the company Onyx Renewable Partners. She provided an insightful overview of AlsoEnergy's dashboard, which allows for real time access to solar power operations, and how to use it to obtain current and past data regarding power generation. Ms. Crider provided an account to access and download the data from that dashboard, allowing for the gathering of the data for the project.

Solar power has seen an increasing rise in usage across the world in light of climate change and the search for alternative energy sources [30]. However, their implementation and wide scale adoption into large scale grid systems presents a novel challenge stemming from their uncertain and intermittent production nature, with local weather being a significant factor [5][6]. As such, the prediction of future solar power production, such as in the short term or intra-day, is necessary for improved power load management and maintenance [7]. Without accurate forecasting, day to day operations and the health of the system may suffer from negative consequences, impacting grid operators and electricity consumers [8].

Much like the upward trend of solar power in recent years, there has been increased focus on developing prediction models for solar power, particularly using machine learning algorithms [9]. A variety of models have been researched, such as support vector machines [10][11][12], random forests [13], and artificial neural networks (ANN) [14]. Despite drawbacks, such as long training times for ANNs [10], machine learning has proven to be more versatile and useful than traditional methods based on physical models or statistical time series analysis [9]. Metaheuristic or optimization algorithms have also been explored as ways to improve machine learning models and their predictions [5][9][15].

This project aims to extend the efforts of Jason Liu, further benefiting the Loomis community by conducting the first in depth exploration of the data produced by the solar plant, specifically how machine learning can be applied to that data for power prediction. Specific focus is directed on the building and comparison of different machine learning models for short term forecasting and prediction of the solar field's power output given past data from the solar plant. Depending on the dataset used, the forecast horizon is 15 minutes or an hour ahead.

The structure of the rest of the paper is as follows: Section 2, Methods, describes this project's process in acquiring knowledge, analyzing data, and elaborates on how and which machine learning models were applied. Section 3, Results, summarizes the performance of the models and compares their error metrics to each other and a baseline persistence model. Section 4, Discussion and Future Work, offers possible ideas for further research. The data and project code, in the form of Jupyter Notebooks, can be found at https://github.com/dyuangm/Solar-power-prediction-with-machine-learning.

2. Methods

2.1. Data Collection

Power generation and onsite weather data for Loomis Chaffee's solar plant were gathered from the solar plant's online dashboard, AlsoEnergy [16]. A screenshot of the page on the dashboard to download data is shown in Figure 1. Data from a range of 12/23/2019 to 12/26/2020, obtained 12/27/2020, were downloaded in monthly batches with increments of 15 minutes from three main categories: power, temperature, and irradiance. The downloaded data, in comma-separated values (.csv) files, were labeled with the time range they contained and organized into folders labeled with their categories. The total size of the downloaded data was approximately 4 megabytes. In total, six raw data features were obtained from the solar plant's dashboard, shown in Table 1.

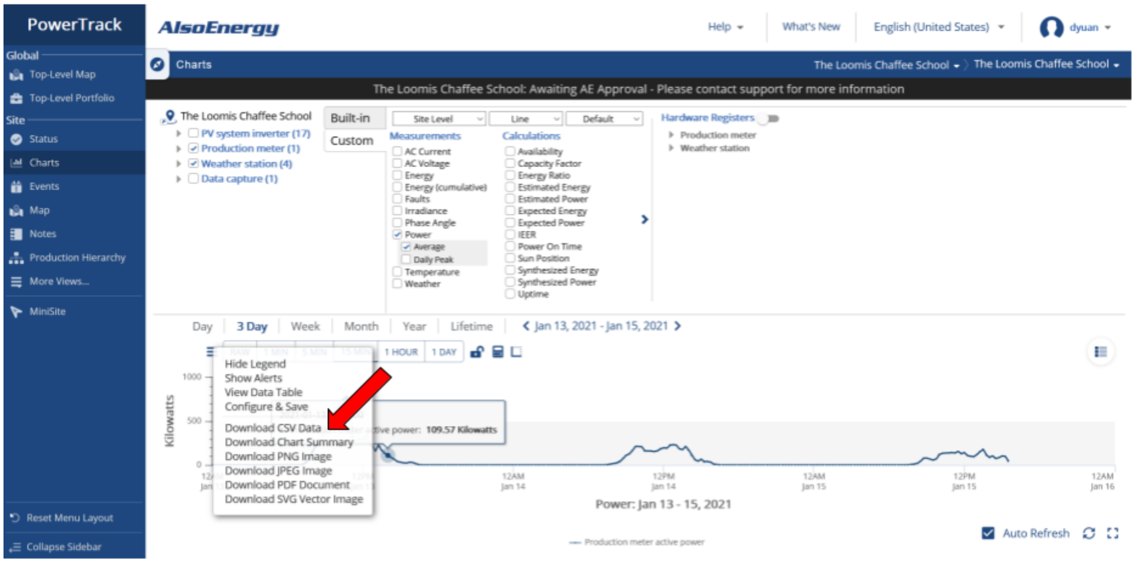


Figure 1: A screenshot of the dashboard used to obtain data with the button for downloading pointed out.

Category (units)	Data feature
Power output (kilowatts)	Average power
Temperature (Fahrenheit)	Module temperature
	Device temperature
	Ambient temperature
Irradiance (watts per-square meter)	Global horizontal irradiance (GHI)
	Plane of array irradiance (POA)

Table 1: The raw data features and their units collected from Loomis' solar plant's dashboard.

2.2. Data Exploration

Prior to conducting any machine learning, the data was first analyzed and inspected for familiarity and error. An example slice of solar power output for one week 12/24/2019 to 12/30/2019 is shown in Figure 2, making clear the wide variability of daily power output, which is due to varying weather conditions.

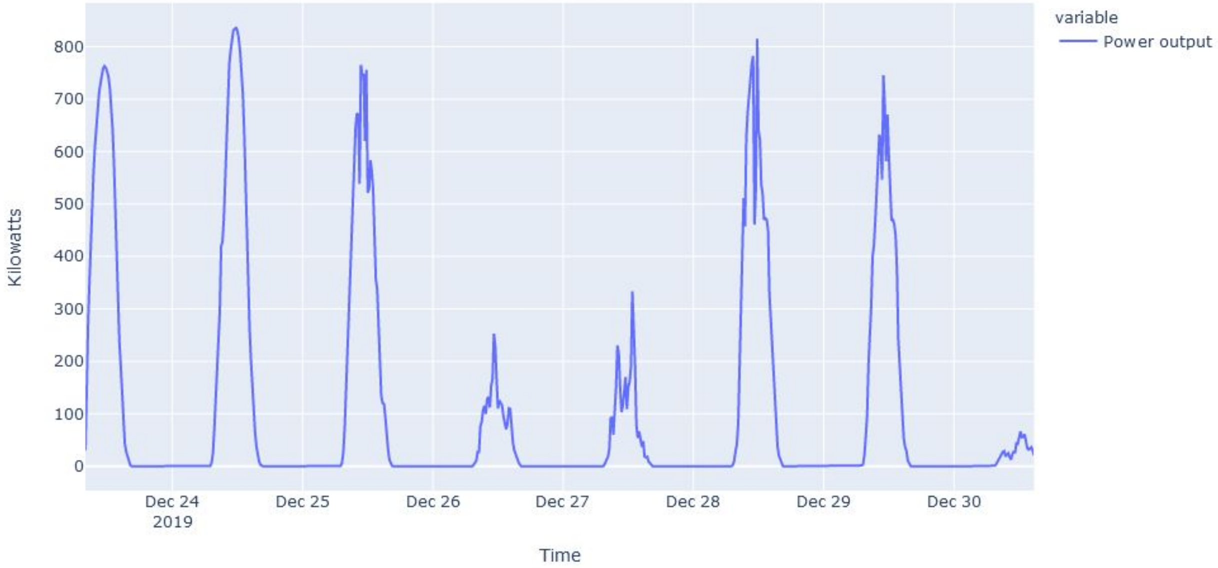


Figure 2: The power output of Loomis' solar field for one week 12/24/2019 to 12/30/2019.

A "missing values" count revealed that each data feature had over 2700 timestamps with missing values, mostly originating from when the solar array was inoperative for nearly two months over the summer. The average hourly output power for each month was calculated, indicating the most and least productive months, May and December respectively, as shown in Figure 3. This is to be expected, as those months are when the sun is highest and lowest in the sky in Windsor, CT. Figure 3 also shows that nearly all power production occurs between the hours of 6 AM and 7 PM. Data values occurring at night and outside that time frame are discarded from subsequent analyses and charts, due to their irrelevance in regards to solar power.

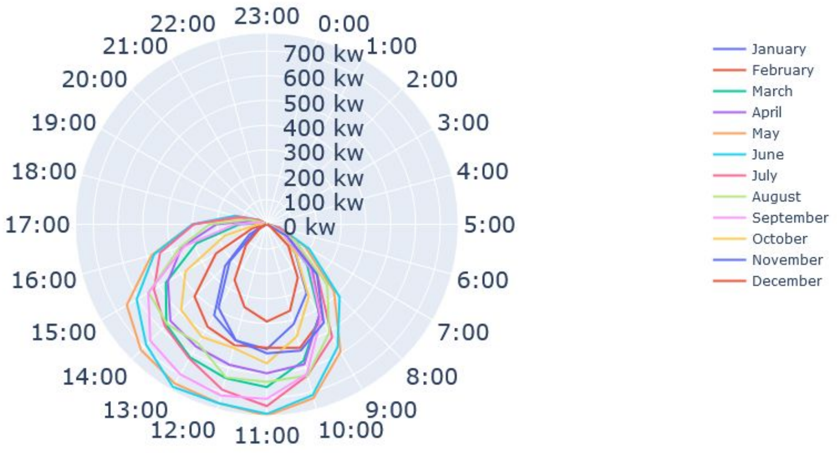
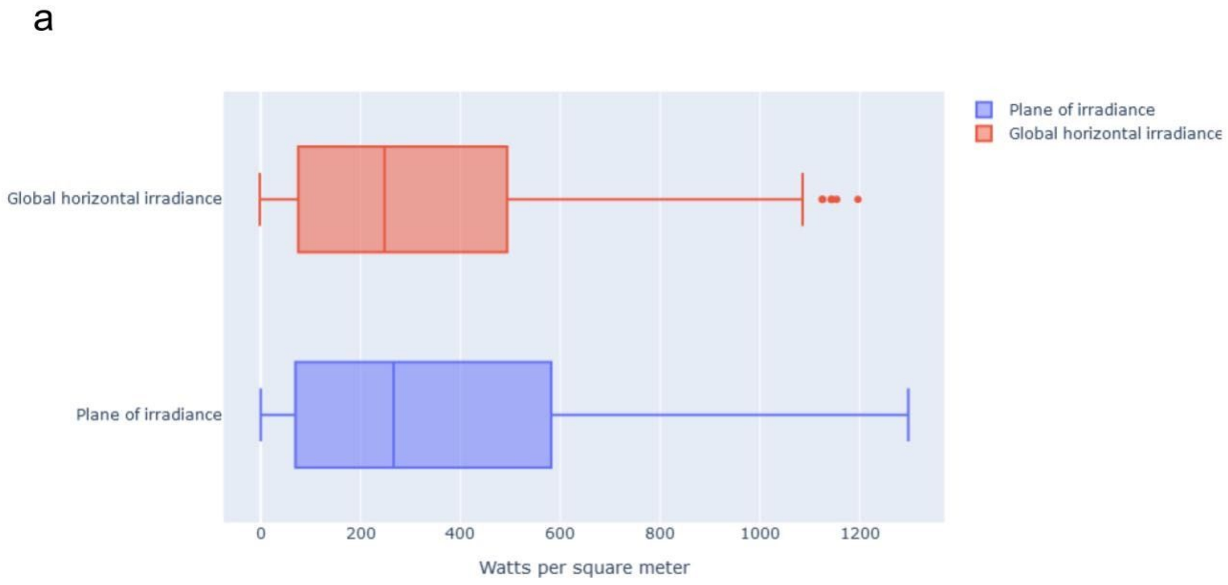


Figure 3: The average hourly power output in kilowatts by month. The least productive month, December, is represented by the smallest circle in red, and the most productive, May, by the largest circle in orange.

Box and whisker plots, Figure 4, and histograms, Figure 5, were drawn for each of the six categories, in order to observe distribution and detect anomalous values. Out of the six, only the device temperature had erroneous values, where the recorded temperature was far below reasonability, as shown in Figure 4b. The histograms of the power and two irradiance data, in Figure 5, the three of which are highly correlated, are skewed left, explained by how maximum solar intensity only occurs a small portion of the day around noon. Irradiance means the amount of power an area receives through sunlight, with global horizontal irradiance referring to a horizontal area having zero degrees tilt and plane of array irradiance measuring the irradiance for an area with the tilt of the solar panels.



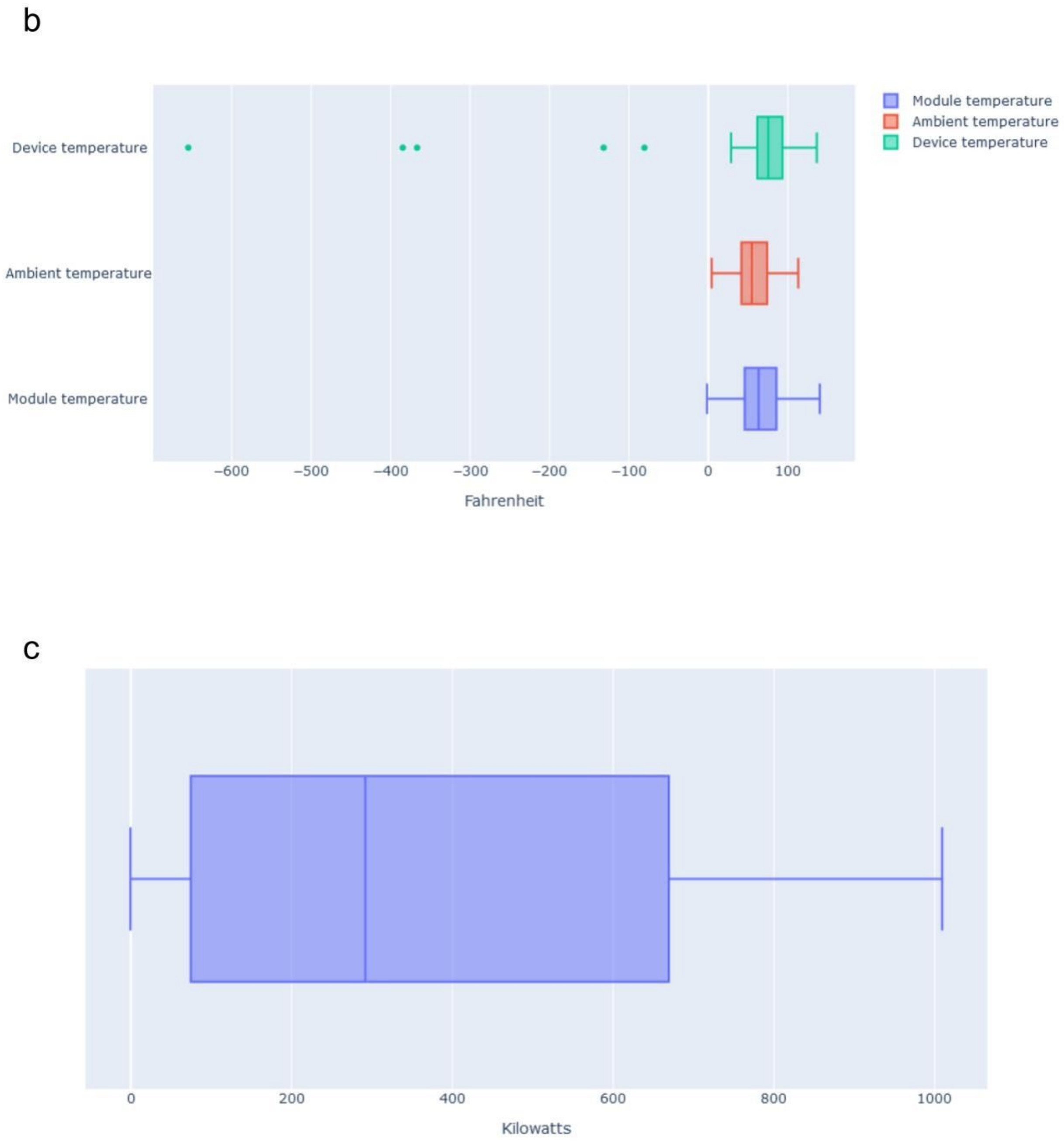


Figure 4: Box plots of the downloaded data by category excluding nighttime values.

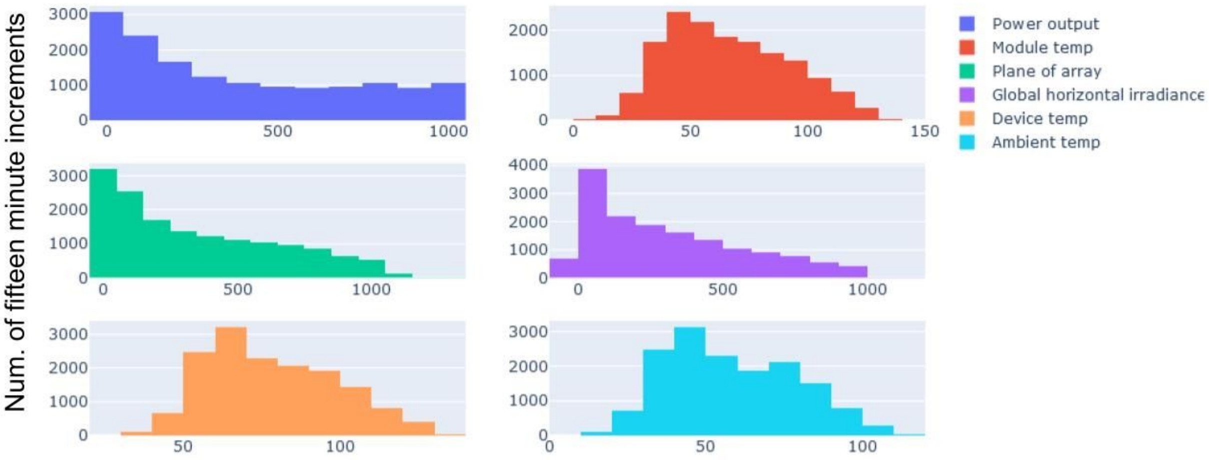


Figure 5: Histograms for each data feature excluding nighttime and anomalous temperature values.

Finally, Figure 6 shows a correlation heatmap of the six types of data. Unsurprisingly, both types of irradiance (POA, GHI) are strongly correlated with power, but the module temperature also shows a notable correlation (0.67). All the data was manipulated and managed in Python v3.8.3 scripts using the Pandas [17] and Numpy [18] libraries, and all plots generated using either the Seaborn [19] or Plotly [20] graphing libraries.

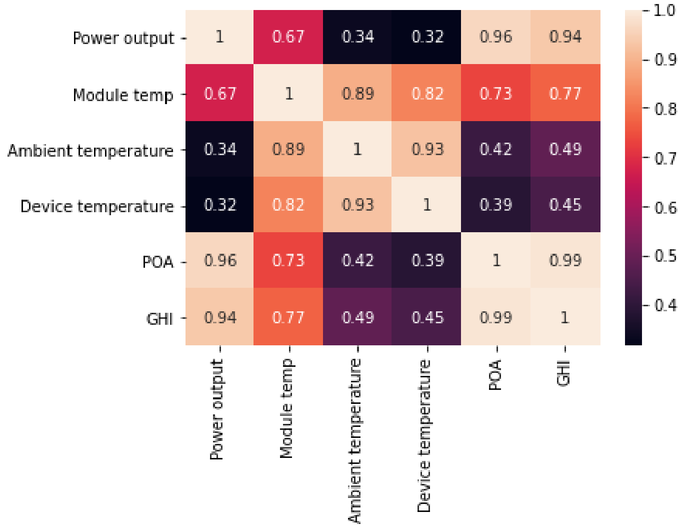


Figure 6: Heatmap showing correlations of the six types of data with each other.

2.3. Data Preprocessing

Before the data can be used to train machine learning models, it must be preprocessed, using knowledge from the previous data exploration. First, timestamps with missing values were identified and corresponding rows removed. The aforementioned anomalous device temperature values were replaced with the median device temperature value. Irradiance and power values less than zero were set to zero. An additional feature was extracted from each timestamp in order to include the effect that the time of day has on solar power, indicating the percentage of time passed since sunrise until noon for morning values and a symmetric value for afternoon values. The Astral Python module [21] was used to obtain sunrise, sunset, and noon times. Table 2 contains example values for a given day's sunrise, noon, and sunset times. This feature cycles from zero at night to one at noon and back each day and is intended to represent the cyclical change of solar intensity connected to time throughout the day.

12/23/2019 Sunrise: 07:15 Noon: 11:49 Sunset: 16:23	
Time	Value
09:30	0.49
11:45	0.98
15:30	0.19

Table 2: Example new values after timestamp conversion for 12/23/2019.

For the prediction of one step ahead power, additional features were added: the current power, the power one step behind, and the power two steps behind. Table 3 shows all the features that make up the final dataset, split into the inputs to the machine learning models and the target data. The dataset was then split into a training set consisting of the first 80% of the data, for building each machine learning model, and a testing set consisting of the remaining 20%, for evaluating each model's performance. Both sets were then normalized to a range of -1 to 1 by individual feature. Normalization is common and often necessary for machine learning, especially for support vector machines [22]. In all, four datasets were considered for machine learning, combining including/excluding nighttime values and fifteen-minute vs hourly increments of data. The size of the datasets ranged from 32914 rows for 15 minute increments including nighttime values to 4061 rows for hourly increments excluding nighttime values.

Input data (9 columns)			**Target data (1 column)**
Current power	Power one increment behind	Power two increments behind	One increment ahead power
Module temperature	Ambient temperature	Device temperature	
Plane of array irradiance	Global horizontal irradiance	Cyclical value from timestamps	

Table 3: Final dataset and features

2.4. Applying Machine Learning Models

Three popular machine learning models for regression tasks were selected: support vector machines, random forests, and k-nearest neighbors. All code was written in Python, and the implementations used were from the versatile Scikit-Learn library [23]. This goal of power prediction is a supervised regression problem, as the output of the algorithm is a continuous value and the future target power is known for training. During the fitting of each model to the data, hyperparameter tuning was conducted to optimize each model. The hyperparameters must be chosen prior to training, and thus have a significant impact on the quality of results.

Hyperparameters were optimized through grid search [24] and cross validation, where a parameter space is provided to exhaustively search through. In cross validation, each combination of parameters is validated through training and scoring on smaller subsets of the training set. A course parameter grid is provided first, then subsequent grid searches focus on the best section of that grid for refinement. All grid searches were run with four fold cross validation, using four smaller subsets of the training data for validating each hyperparameter combination.

Support vector machines (SVM) can work with linear and non linear datasets, a powerful and often used model. For regression, they attempt to find sets of support vectors in the data that determine the hyperplane that contains the most data points with the fewest outside. The data given to a support vector machine is transformed to higher dimensions through kernels. The adjustment of hyperparameters in each kernel is particularly important for this type of model, and three parameters were chosen for tuning: C, which determines the penalty for points not on the plane; gamma, which controls the amount of influence of a single data point; and the type of kernel [25]. The initial ranges for C were from 10^0 to 10^4, gamma 10^{-5} to 10^2, and to use either the linear or radial basis function for the kernel.

Random forests (RF) work by combining the results of several decision trees. An individual decision tree makes a prediction independently based on a random subset of features and data, and the random forest averages the predictions of all the trees, reducing the risk of overfitting. A number of random forest hyperparameters were selected for adjustment: the number of estimators, maximum number of features to consider for each split, maximum depth of a tree, minimum samples per split, minimum samples per leaf node, and whether to bootstrap, that is randomly select a data subset, for each tree. Due to the number of hyperparameters, a randomized grid search instead of a full exhaustive one was first conducted, and normal grid search performed on its best performing values. Figure 7 shows a portion of one decision tree used for prediction.

K-nearest neighbors (KNN) is one of the more simple and intuitive machine learning models, yet still highly useful. To make predictions given data inputs, it considers the k nearest neighboring data points to that input for the output, such as by averaging them. The hyperparameters optimized for this model were the number of neighbors, the type of weight to give each neighboring point, and the distance metric for determining the similarity between data points. The initial parameter space for the number of neighbors was 0 to 19, weights either uniform or distance, and the distance metric either Manhattan, Euclidean, or Minkowski.

Finally, a persistence model (PM) was created as a simple baseline for comparison purposes [7]. It defines the prediction for the output of future power to simply be the current power, as if the current production persisted.

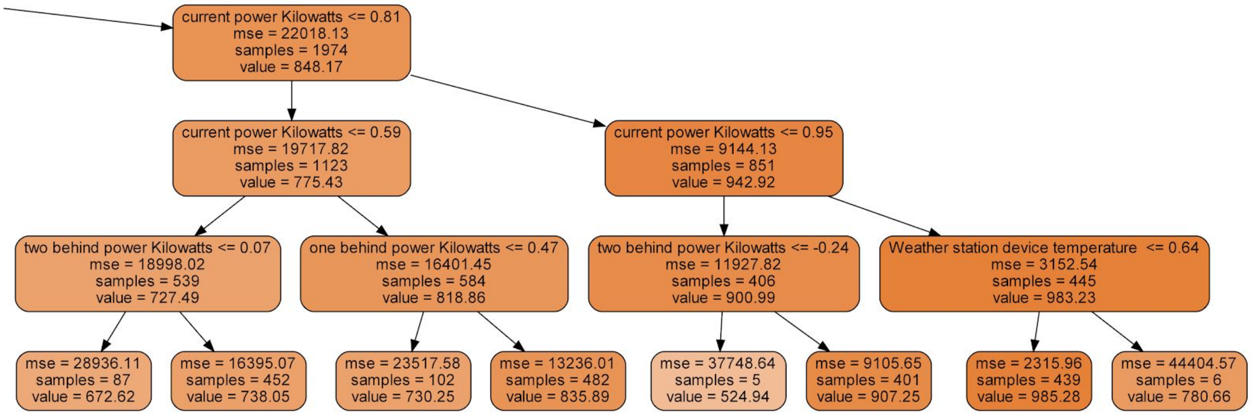


Figure 7: A portion of a random forest decision tree trained on data from the power plant. Predictions are made by following successive nodes and splits according to input data until reaching a leaf node. For example at the topmost node, if the current power Kilowatts is greater than 0.81, then the right node is followed.

3. Results

The three models and the persistence method were fit on four datasets, which differed in their inclusion/exclusion of nighttime values and a forecast horizon of either 15 minutes or an hour. To compare all the combinations of models and datasets, two common error metrics were used, root mean squared error (RMSE) and the coefficient of determination or r-squared score (r^2). For RMSE, the error between each prediction and corresponding target value is squared, then the square root is taken of the average squared error, with kilowatts as the units. Because the error is first squared, this metric gives a higher penalty to errors of larger magnitude, and it generally represents the standard deviation of error for a single observation or future prediction [26]. The r-squared score reflects how similar a regression model's predictions are to the data it has been given. Varying from 0 to 1, the higher the r-squared value is higher the correlation [27].

Table 4 shows the RMSE and r-squared calculated on daytime values of the test set for each model fitted to each dataset, along with the percentage error of the RMSE relative to the 1 megawatt capacity of Loomis' solar plant. Comparing the results shown in Table 4 reveals how the inclusion nighttime values has a marginal impact on the performance, if not slightly improving it. The lowest RMSE is 71.99, indicating the best forecasting capabilities, representing a support vector machine trained on the 15 minute increments and including nighttime values dataset. Support vector machines also performed the best for all datasets and the worst model was k-nearest neighbors, though all models were more accurate than the persistence model. Overall, all three models show similar error metrics and thus performance by dataset, outperforming the persistence model for all datasets. The average RMSE for the models trained on the two datasets with 15 minute increments, 73.85, is substantially lower than the average RMSE for the two datasets with hourly increments, 134.61. For hourly forecasting, a major decrease in RMSE, at least 20%, is observed for all models compared to the persistence model, as opposed to a relatively negligible accuracy boost, only at least 3%, for the same case for 15 minute predictions. Figure 8 shows a selected portion of support vector machine predicted outputs plotted against the ground truth across all four datasets, showing how accuracy and performance varies depending on the data. It may appear that the machine learning predictions seem to curiously resemble a delayed version of the target values, somewhat akin in behavior to the persistence model.

	Including nighttime values				**Excluding nighttime values**			
15 minutes ahead	Model	RMSE	% error	R^2	Model	RMSE	% error	R^2
	SVM†	71.99	7.20	0.93	SVM*	73.99	7.40	0.94
	RF	72.46	7.25	0.93	RF	74.51	7.45	0.93
	KNN	73.53	7.35	0.92	KNN	76.60	7.66	0.93
	PM	75.85	7.59	0.93	PM	79.16	7.92	0.93
1 hour ahead	Model	RMSE	% error	R^2	Model	RMSE	% error	R^2
	SVM*	122.93	12.29	0.79	SVM*	123.92	12.39	0.81
	RF	126.91	12.69	0.75	RF	130.16	13.01	0.77
	KNN	137.26	13.73	0.71	KNN	136.41	13.64	0.74
	PM	174.38	17.44	0.62	PM	178.07	17.81	0.64

Table 4: The best RMSE for each model for each dataset, alongside the corresponding r-squared score and percent error. A * indicates the model with the lowest RMSE for each dataset, and a † indicates the least overall.

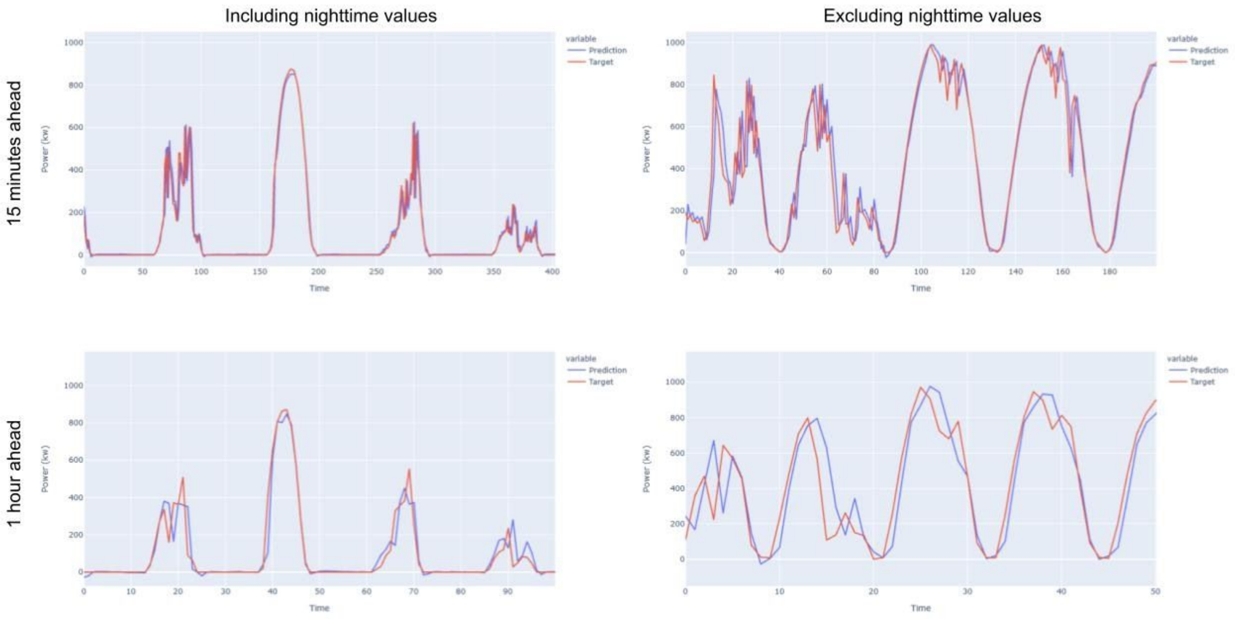


Figure 8: Selected portions of the graphs of support vector machine predictions against the actual power output graphs across all four datasets. Blue represents the prediction and red the target.

Hyperparameter tuning led to increases in accuracy for all three models, as shown in Table 5. Grid search was run twice on the models for refinement, after which there was no significant increase in performance. Figure 9 shows a contour plot of the two hyperparameters, C and gamma, for a support vector machine trained on the largest dataset after the first grid search, with the color indicating the respective RMSE and quality of prediction. The lighter regions indicate the combinations of hyperparameters that led to better accuracy during training, which improved the models.

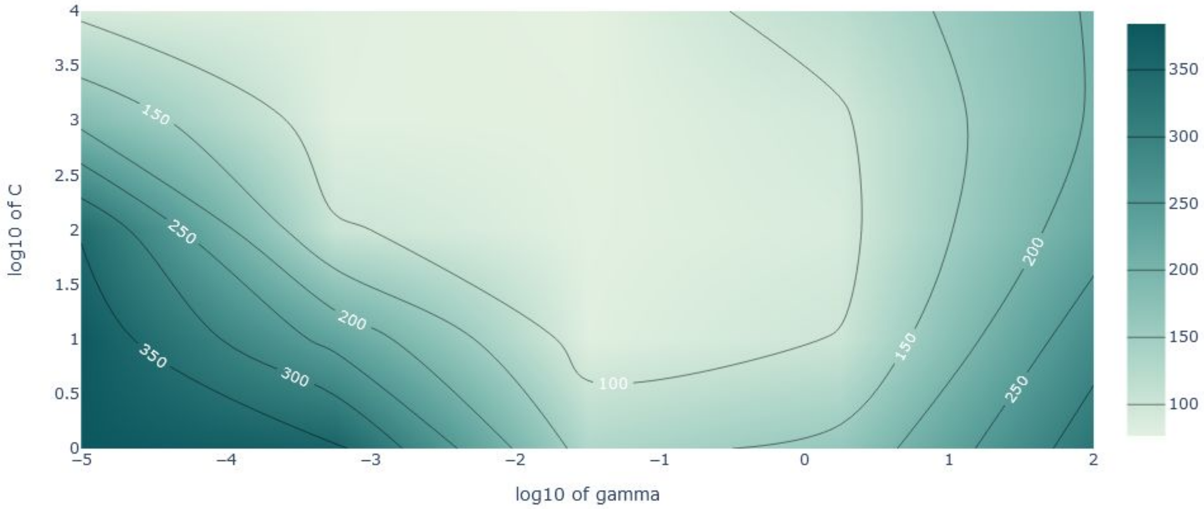


Figure 9: Contour plot of C and gamma, two hyperparameters for SVMs, and their corresponding RMSE. The lighter the color, the better the performance of the model.

	Including nighttime values			**Excluding nighttime values**		
	Model	1st run	2nd run	Model	1st run	2nd run
15 minutes ahead	SVM	72.16	71.99	SVM	74.15	73.99
	RF	72.46	72.49	RF	74.51	74.63
	KNN	73.67	73.53	KNN	76.69	76.60
	Model	1st run	2nd run	Model	1st run	2nd run
1 hour ahead	SVM	139.82	122.93	SVM	135.82	123.92
	RF	127.19	126.91	RF	131.13	130.16
	KNN	137.26	137.26	KNN	136.41	136.41

Table 5: The RMSE for each model after each run of grid search by dataset. In most cases the results show better performance for all models.

Each feature's relative importance in this prediction task can be determined by a random forest [28]. Table 6 shows the feature importances calculated by a random forest on the dataset which is in hourly increments and excludes nighttime values. By far the current power is the most influential factor, and the cyclical time value also appears to have significant impact. Training on different datasets however results in different feature importances.

Variable	**Importance percentage**	**Variable**	**Importance percentage**	**Variable**	**Importance percentage**
Current power	64.17	Two steps behind power	3.60	GHI	1.39
Cyclical time value	20.06	One step behind power	1.96	Ambient temperature	1.03
POA	5.58	Module temperature	1.47	Device temperature	0.75

Table 6: The relative importances of each data feature in prediction regression calculated by a random forest trained on the data.

4. Discussion and Future Work

Support vector machines, random forests, and k-nearest neighbors models were trained across four different datasets from Loomis' solar field to perform the regression task of short term solar power output prediction. The raw data included power, temperature, and irradiance information, which was preprocessed and augmented prior to model training. The models' varying errors and performance after hyperparameter tuning, shown in Table 4, were compared to a baseline persistence model. Including nighttime values led to marginally better performance, and the longer hourly forecasting horizon led to a larger root mean squared error and lower accuracy compared to the shorter 15 minute horizon. The best model was a support vector machine with a RMSE of 71.99 and r-squared of 0.93 on the dataset with 15 minute increments and including nighttime values. Though all models outperformed the persistence method, the persistence model had only slightly higher and still similar error metrics to the machine learning models for 15 minute ahead prediction, indicating that the performance of machine learning models for this time horizon were somewhat negligible compared to a naive guess. However, the models significantly improved in error and accuracy for hour ahead predictions compared to the persistence model, showing how they have learned from the data and have some utility in forecasting. The varying performances of the persistence model also confirms that power output undergoes a wider range of variability in hourly timeframes and a smaller range for 15 minute increments. Feature importance analysis through random forests, shown in Table 6, reveals that among the most decisive types of data for prediction are the current power and a cyclical representation of time calculated from timestamps, indicating that the second, artificially extracted feature had a noticeable impact on prediction. The tendency for the models to output what appears to resemble a delayed version of the target values, as seen in Figure 8, likely stems from how the inputs were mostly data from one time step behind. This phenomenon may explain why the average RMSE for models trained on 15 minute increment data is considerably lower than the average RMSE on hourly increment data; the decrease in variability from one fifteen minute timestamp to the next versus for hourly timestamps would mean that a prediction of a delayed value would not differ as much from the target. Though the time series aspect of the data was somewhat leveraged with the inclusion of past power production values as input, future work could focus on applying more data preprocessing and feature engineering strategies characteristic of time series, such as decomposition and differencing, for hourly prediction given its relative success compared to the shorter horizon. Though the models had promising prediction capabilities based only on data gathered from the power plant itself, especially for hour ahead forecasting, further research could expand the data to also include local weather information obtained from weather agencies, such as cloudiness and humidity, alongside simply training on larger datasets that become available as time goes on.

References

[1] Liu, J., Dyreson, J., et al. "LC SOLAR." 2018 https://docs.google.com/presentation/d/1SFo0CpPELpUANTV5xDDGSKEO_t4glNvttzsFWGhB6nA/edit#slide=id.p1

[2] "ENCON Heating and Air Conditioning" https://www.goencon.com/

[3] "Power Factors, LLC" https://pfdrive.com/

[4] "AlsoEnergy, Inc." https://home.alsoenergy.com/

[5] Lai, J.-P.; Chang, Y.-M.; Chen, C.-H.; Pai, P.-F. (2020). A Survey of Machine Learning Models in Renewable Energy Predictions. *Appl. Sci.* 2020, 10, 5975. https://doi.org/10.3390/app10175975.

[6] Sobrina Sobri, Sam Koohi-Kamali, Nasrudin Abd. Rahim, Solar photovoltaic generation forecasting methods: A review, Energy Conversion and Management, Volume 156, 2018, Pages 459-497, ISSN 0196-8904, https://doi.org/10.1016/j.enconman.2017.11.019.

[7] Majidpour, M.; Nazaripouya, H.; Chu, P.; Pota, H.R.; Gadh, R. Fast Univariate Time Series Prediction of Solar Power for Real-Time Control of Energy Storage System. *Forecasting*, 2019, 1, 107-120. https://doi.org/10.3390/forecast1010008

[8] Cyril Voyant, Gilles Notton, Soteris Kalogirou, Marie-Laure Nivet, Christophe Paoli, Fabrice Motte, Alexis Fouilloy, Machine learning methods for solar radiation forecasting: A review, Renewable Energy, Volume 105, 2017, Pages 569-582, ISSN 0960-1481, https://doi.org/10.1016/j.renene.2016.12.095.

[9] Akhter, Muhammad Naveed; Mekhilef, Saad; Mokhlis, Hazlie; Mohamed Shah, Noraisyah: 'Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques', IET Renewable Power Generation, 2019, 13, (7), p. 1009-1023, DOI: 10.1049/iet-rpg.2018.5649
IET Digital Library, https://digital-library.theiet.org/content/journals/10.1049/iet-rpg.2018.5649

[10] Alireza Zendehboudi, M.A. Baseer, R. Saidur,
Application of support vector machine models for forecasting solar and wind energy resources: A review, Journal of Cleaner Production, Volume 199, 2018, Pages 272-285, ISSN 0959-6526, https://doi.org/10.1016/j.jclepro.2018.07.164.

[11] Belaid, Sabrina & Mellit, Adel & Boualit, Hamid & Mohamed, Zaiani (2020). Hourly global solar forecasting models based on a supervised machine learning algorithm and time series principle. *International Journal of Ambient Energy*. 1-25. https://doi.org/10.1080/01430750.2020.1718754.

[12] Alfadda, Abdullah & Adhikari, Rajendra & Kuzlu, Murat & Rahman, Saifur (2017). Hour-ahead solar PV power forecasting using SVR based approach. 1-5. https://doi.org/10.1109/ISGT.2017.8086020.

[13] Da Liu, Kun Sun (2019). Random forest solar power forecast based on classification optimization, *Energy*, Volume 187, 2019, 115940, ISSN 0360-5442, https://doi.org/10.1016/j.energy.2019.115940.

[14] Al-Dahidi Sameer, Ayadi Osama, Adeeb Jehad, and Louzazni Mohamed (2019). Assessment of Artificial Neural Networks Learning Algorithms and Training Datasets for Solar Photovoltaic Power Production Prediction, *Frontiers in Energy Research*, Volume 7, 2019, Page 130, ISSN 2296-598X, https://doi.org/10.3389/fenrg.2019.00130.

[15] Guo-Qian Lin, Ling-Ling Li, Ming-Lang Tseng, Han-Min Liu, Dong-Dong Yuan, Raymond R. Tan, An improved moth-flame optimization algorithm for support vector machine prediction of photovoltaic power generation, Journal of Cleaner Production, Volume 253, 2020, 119966, ISSN 0959-6526, https://doi.org/10.1016/j.jclepro.2020.119966.

[16] “PowerTrack” https://apps.alsoenergy.com/Account/Login?returnUrl=%2fpowertrack

[17] “pandas documentation” https://pandas.pydata.org/docs/

[18] “NumPy v1.19 Manual” https://numpy.org/doc/stable/

[19] “seaborn: statistical data visualization” https://seaborn.pydata.org/

[20] “Plotly Python Open Source Graphing Library” https://plotly.com/python/

[21] “Astral v2.2” https://astral.readthedocs.io/en/latest/index.html

[22] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, “A Practical Guide to Support Vector Classification.” 2016, https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[23] “scikit-learn” https://scikit-learn.org/stable/

[24] “3.2. Tuning the hyper-parameters of an estimator”
https://scikit-learn.org/stable/modules/grid_search.html
[25] “Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters”
https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167
[26] “What does RMSE really mean?”
https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e
[27] “Data Science: Explaining R² in Statistics”
https://towardsdatascience.com/data-science-explaining-r%C2%B2-in-statistics-6f34e7f0a9bb
[28] Brownlee, J. “How to Calculate Feature Importance With Python.”
https://machinelearningmastery.com/calculate-feature-importance-with-python/
[29] Kepa Solaun, Emilio Cerdá,
Climate change impacts on renewable energy generation. A review of quantitative projections, Renewable and Sustainable Energy Reviews, Volume 116, 2019, 109415, ISSN 1364-0321,
https://doi.org/10.1016/j.rser.2019.109415.
[30] “EIA expects U.S. electricity generation from renewables to soon surpass nuclear and coal”
https://www.eia.gov/todayinenergy/detail.php?id=42655
Code repository: https://github.com/dyuangm/Solar-power-prediction-with-machine-learning