Emily Ford, Senthilkumar Murali, and Vivek Verma

Dr. Zhuang

MAE/MSE 598: Machine Learning for Engineers

Machine Learning for Predicting and Interpreting Solar Panel Power Output

1.0 Introduction

Renewable energy is reliable, plentiful, greatly reduces carbon emission levels, and combats climate change caused by fossil fuel use. The leading source of renewable energy is solar power, and its generation is through photovoltaic cells. Though photovoltaic cells are considered a significant source for future energy generation, their return on investment and upfront costs hinder their deployments **[1]**. One of the challenges related to solar power is the lack of predictable supply because of continually changing weather conditions. Since photovoltaic cells generate electricity by converting solar energy to electric current, the amount of solar power produced in a day is significant in determining the output of the photovoltaic system. Therefore, the amount of electricity produced depends upon solar irradiance on a particular day, depending on various parameters such as location, time, and weather patterns.

Unfavorable weather reduces the output of the solar plant to a large extent. Therefore, to fulfill energy requirements, a solar power supply company must supplement the missing power by purchasing it from other companies running on fossil fuels. Currently, existing systems do not have enough capacity to entirely replace fossil fuels. These companies' power rates depend upon the amount of power required and the order's timeliness. Timely order placement with these companies helps meet the promised power supply goals and reduces the cost. Here prior knowledge of the power produced plays a crucial role in maintaining service quality and reducing cost. Thus, an accurate forecast of power output is becoming an important issue **[1]**.

Similar to the demand for power forecasting solutions, forecasting with machine learning techniques has gained popularity in recent years. The improved computational capacity of computers and the higher availability of quality data have made machine learning techniques highly useful for forecasting and clustering. To study this, we have used the "Solar Power Generation data" dataset from Kaggle **[2]**. The dataset has been gathered at two solar power plants in India over 34 days. Techniques from supervised and unsupervised machine learning have been used to study the solar power output, and are presented in this study.

2.0 Methods

This section of the paper shall review the different machine learning algorithms used in the project, including preprocessing techniques and the output data to be studied.

2.1 Preprocessing and Inputs

The data for this study is available on Kaggle and is entitled "Solar Power Generation Data" **[2]**. The data contains a total of 6,440 data points over the course of 34 days of solar power generation at two different power plants. Measurements are recorded at 15-minute intervals, typically just before sunrise and after sunset. All data results for this study are made by combining and mixing both power plant datasets and performing a 5-fold cross-validation. The average of the 5 fold results is presented as the final metric. Prior to any machine learning, the input data was pulled from the given datasets and converted to a usable format for the algorithms. The five inputs (the weather sensor data) were the date, time, ambient temperature, module temperature, and irritation amount. In order to be used as inputs into the regression and clustering analysis, the day and time data had to be converted into continuous numbers. The time was converted to minutes in a day using Python's DateTime library **[3]** and calling the hours and minutes from the time column. This meant that the minimum value for time was 0 and the maximum value was 1439 minutes in a day. For the year, month, and day information, the .toordinal() function from Python's DateTime library **[3]** was utilized to convert the date to the proleptic Gregorian ordinal. The proleptic Gegorian ordinal applies Gregorian calendar days to dates prior to its introduction in the 1500s, therefore January 1 of year 1 is day 1 **[4]**. All five inputs, including the remaining ambient temperature, module temperature, and irradiation values were normalized by subtracting the mean and dividing by the standard deviation via scikit learn's StandardScaler function **[5]**.

2.2 Regression via Supervised Learning

2.2.1 Brief Review of Neural Networks and Forest Ensembles

The supervised machine learning techniques applied to this non-linear, cyclical, solar power problem were artificial neural networks (ANNs), random forest (RF), extra trees forests (ET), and gradient boosted trees (GBT). The ANN is composed of layers of neurons with weighted connections and activation functions with learning rates dictating the connection between the input layer and the output layer. On the other hand, the forest models (random forest, extra trees, and gradient boosted trees) are ensemble models composed of a large number of weak individual machine learners. In RF, the best split of the data is determined by considering all of the input features and checking a criterion to select the most discriminative threshold **[6][7]**. Each individual decision tree in the RF ensemble does not use the entire set of training data, but a bootstrap sample made from subsets of the training data with replacement **[6]**

[8]. The Extra Trees (ET) regressor randomly draws splits for each feature and the best split, as measured by the chosen criteria, is selected as the splitting rule **[6] [7]** . In the ET regression model, the entire dataset is incorporated into each individual tree **[7]**. The prediction results of the individual trees are averaged to produce the output prediction in the RF and ET regressions. In GBT, an initial tree is trained with the entire dataset. All subsequent trees in the forest are trained to minimize the residual between the predicted and actual values of the previous tree **[7]**. The final prediction is calculated as the weighted sum of the predictions of each tree. To illustrate the need for machine learning to solve such complex problems, a simple linear regression model was additionally trained to compare the prediction quality to that of the machine learning methods. In terms of coding, ANN models were developed via Keras with Tensorflow backend **[9]**, while the forest ensembles come directly from the scikit-learn library **[10][11][12]**.

2.2.2 Supervised Learning Output and Metrics

For the regression-based analysis, the most important value to predict for the solar power plant output was determined to be the daily yield of all solar panels for each subset of time **[1]**. This sum is cumulative, meaning the value increases as the day progresses, as illustrated in Figure 1:

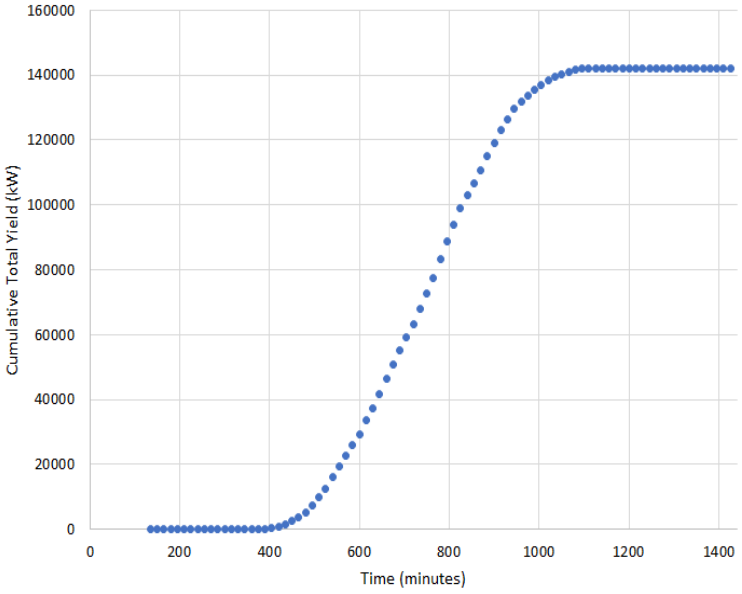

Figure 1: Plot of cumulative total yield (kW) throughout one day.

The sum of the daily yield (kW) of all solar panels is to be predicted for both solar power plants given the time, day, ambient temperature, module temperature, and irradiation information. Like the input values, the output values were normalized to ensure that the change in scale between inputs and outputs did not negatively impact the training and prediction accuracy. The outputs were returned to their original scale to

report final errors and for plotting of the results. The metric utilized to train the machine learners was the mean-squared error (MSE):

$$MSE = \frac{1}{n}\left(\sum_{i=1}^{n} \left|\frac{A_i - P_i}{A_i}\right|\right) \quad (1)$$

where n is the total number of data points, A_i is the actual value, and P_i is the predicted value. Other metrics tracked, but not used to train the models, were the mean absolute error (MAE) and the coefficient of determination (COD or R^2), given as:

$$MAE = \frac{1}{n}\left(\sum_{i=1}^{n} \left|\frac{A_i - P_i}{A_i}\right|\right) \quad (2)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(P_i - A_i)^2}{\sum_{i=1}^{n}(A_i - \bar{A}_i)^2} \quad (3)$$

Where $\bar{A}_i$ is the mean of the actual values.

2.3 Unsupervised Learning via K-Means Clustering

2.3.1 Brief Review K-Means Clustering

K-means is a method by which vectors are quantized. The aim is to partition n number of observations into K clusters in which every observation belongs to the cluster with the nearest mean (cluster centroid) acting as a prototype of the cluster **[13]**. The result is a partition of the data space into Voronoi cells with every class labeled **[14]**. The K-means clustering algorithm minimizes the variance within the clusters. The problem is computationally difficult, but with the help of efficient heuristics **[15],** the convergence is quick and arrives at a local optimum. These types of approaches are similar to the expectation-maximization **[16]** algorithm for mixtures of Gaussian distributions **[17].** Through an interactive approach, this is employed by K-means as well as Gaussian mixture modeling. Both use cluster centers to model a dataset, but K-means is able to find clusters of comparable spatial extent. On the other hand, the expectation-maximization algorithm allows the clusters to have different types of shapes **[18]**. The algorithm is also loosely related to the *K-nearest neighbor classifier* which is used for classification. Applying the 1-nearest neighbor classifier to the cluster centers obtained by a K-means classifies a new dataset into the already existing cluster, which is also known as the Rocchio algorithm **[19]**. Below is an example of the mathematical representation of the algorithm based on different distances:

Euclidean Distance: $d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$ (4)

Manhattan Distance: $d_1(p,q) = \sum_{i=1}^{n} |p_i - q_i|$ (5)

Minkowski Distance: $D(X,Y) = (\sum_{i=1}^{n} |x_i - y_i|^P)^{\frac{1}{P}}$ (6)

2.3.2 Unsupervised Learning Output and Metrics

The solar power plant dataset contained many parameters that could be used to understand the relationship between time and the energy generated. The *AC power* generation output was chosen to study the clustering portion of the project. The power generated in power plants over time was plotted. Below is the plot:

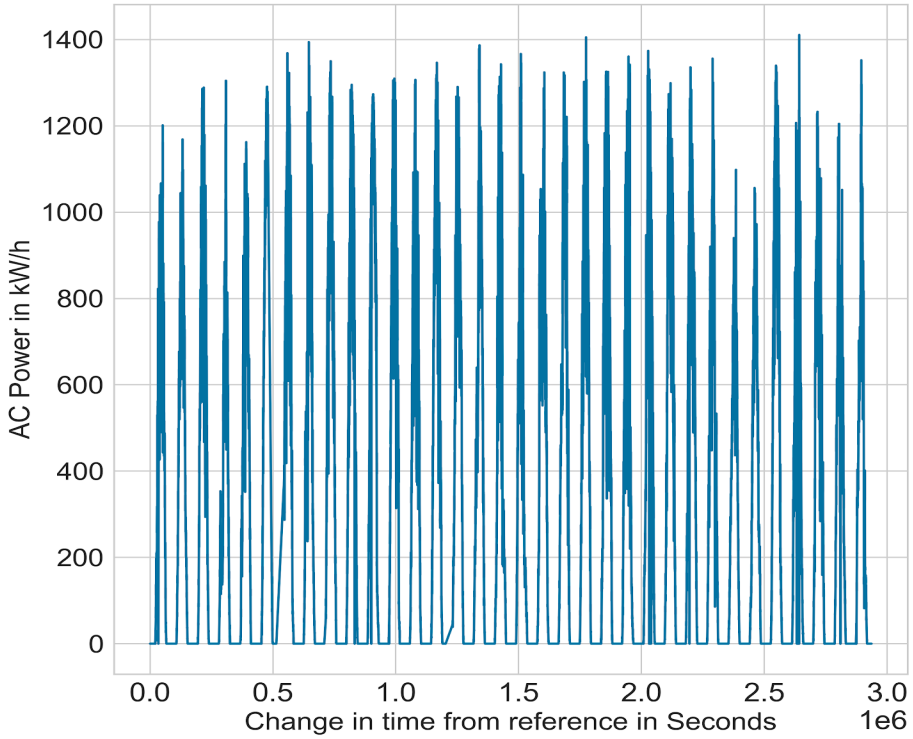

Figure 2: AC Power vs. time

AC power generated is in kW/hour. The start date of the dataset was taken as the reference date. The reference time was the time at which the data collection started. ['15-05-2020 00:00']. The time elapsed was taken as the difference between any future date and the reference date. This difference in time was

then converted into seconds and plotted on the x-axis. A cyclic portion of data was then chosen from the graph above to acquire a clear representation of the characteristics of the data. Below is one day's worth of data that was chosen for representation and performing K-means.

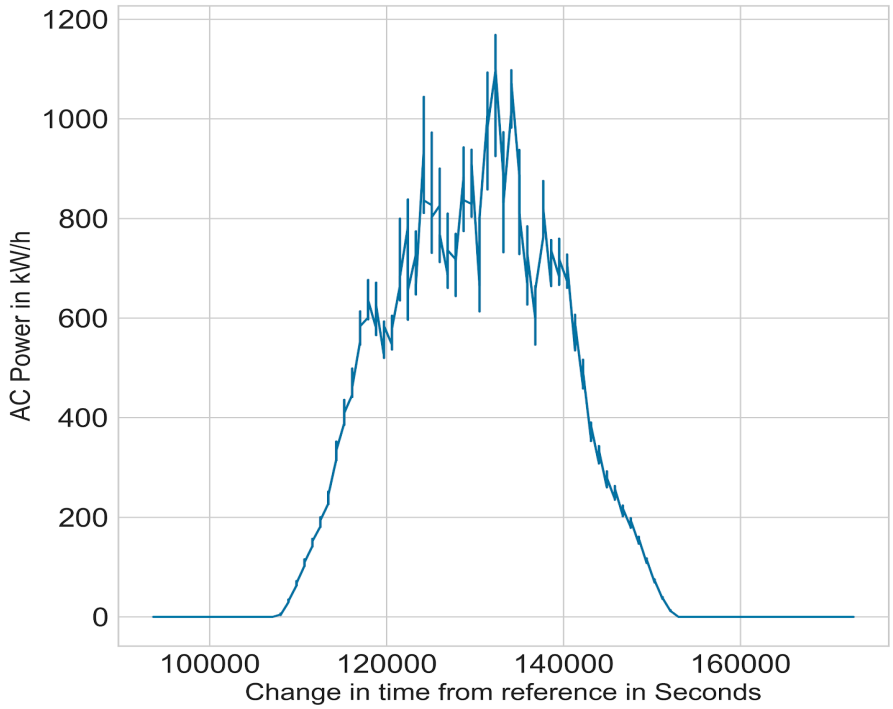

Figure 3: One day's worth of AC power generation vs time data

K-means clustering was performed on the above data to establish a relationship between time vs the power generated during that time. An elbow plot was used to estimate the best value of K for clustering of the data. An elbow plot was generated as the within-cluster-sum-of-squares (WCSS) vs K value. The number of clusters chosen for any given dataset cannot be random since each cluster is formed by comparing and calculating the distance of data points within a cluster to its centroid. The WCSS was used to make an objective estimate of the best number of clusters, K, to use. WCSS is the sum of the squares of the distances of every data point in all clusters to their respective centroids.

$$WCSS = \sum_{C_k}^{C_n} \left(\sum_{d_i \, in \, C_i}^{d_m} distance(d_i, C_k)^2 \right) \tag{7}$$

Here, C is the cluster centroids and d is the data points in each cluster. The main objective is to minimize the sum. The threshold value of K can be found using the elbow plot. Firstly, K is initialized randomly for a range of values and is plotted against the WCSS for each K. The K value was randomly selected until the WCSS decreases rapidly **[20]**.

2.4 Classification

The objective of the classification using this data set was to identify faulty solar panels. But, the data did not have labels that identify the panels as faulty. The idea was to create labels using K-means clustering and then use them for classification using the Random Forest classifier. But using the daily yield, DC power, and AC power, an inference could not be drawn to label the inverters working as faulty or normal. Upon deeper analysis, it was determined that continuous voltage and current data would be required at the inverter level to do the labeling **[21]**. In the below image, classification has been performed based on the power output and irradiation. If there is sunlight and no power was generated then the inverter was identified as faulty. Note how very few of the points were identified as faulty.

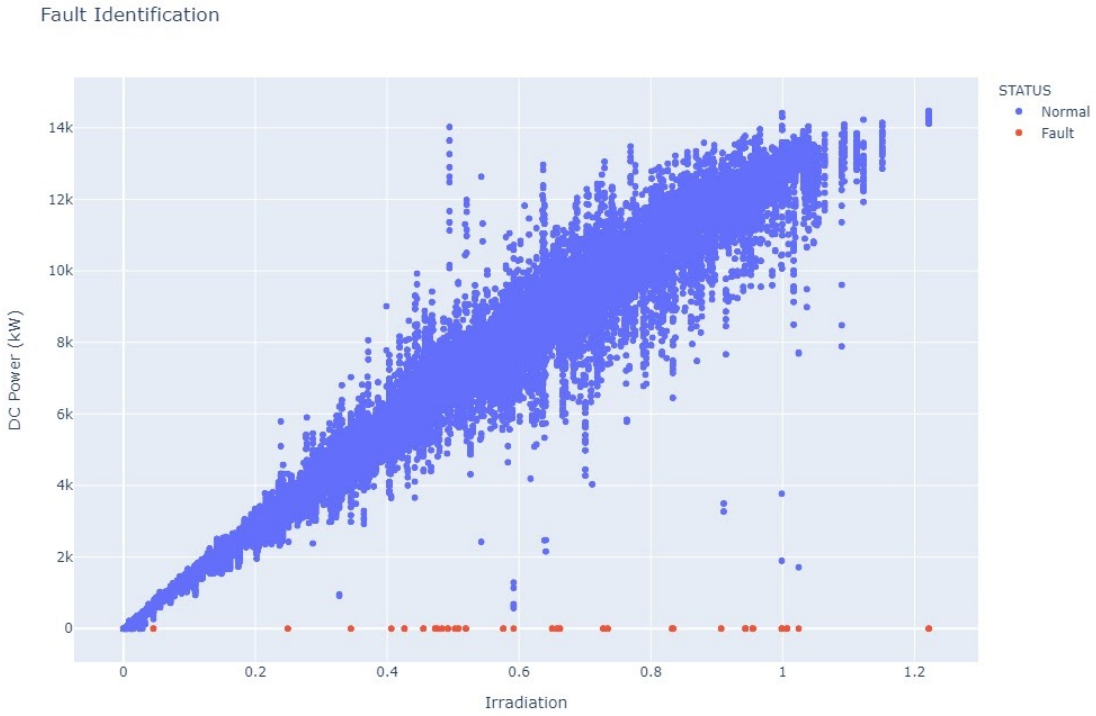

Figure 4: Classification of inverters based on their power output

3.0 Results and Discussion

3.1 Regression Results

The hyperparameters to optimize in the ANN models were the number of hidden layers, the number of neurons in each hidden layer, and the dropout rate. ReLu activation function with a learning rate of 0.001 and an RMSprop optimization scheme was used. For the RF, ET, and GBT models, the number of trees in the forest, the maximum depth of the trees, the minimum number of samples before splitting, and the minimum number of samples per leaf were tuned. Coarse optimization of the hyperparameters followed a

random search pattern, found to be the most efficient method to optimize parameters **[22]**, by generating 30 different random combinations of hyperparameters. For all the models, the parameters which minimized the 5-fold cross-validation MSE were used as the basis for the final models, with some additional grid-based fine-tuning. Table 1 showcases the final hyperparameter settings after tuning.

Table 1: Final hyperparameters used for each ML model and dataset.

Model	Hyperparameter	Final Model Hyperparameter Value
ANN	# hidden layers	4
	# starting neurons	58
	Drop rate	0.0
Random Forest (RF)	# of trees	99
	Maximum depth	14
	Min # samples before split	4
	Minimum # of samples on leaf	2
Extra Trees (ET) Forest	# of trees	186
	Maximum depth	20
	Min # samples before split	3
	Minimum # of samples on leaf	3
Gradient Boosted Trees (GBT)	# of trees	190
	Maximum depth	11
	Min # samples before split	6
	Minimum # of samples on leaf	8

The average and standard deviation of the five-fold cross-validation for each model to predict the cumulative total daily yield of each solar power plant is shown in Table 2 below.

Table 2: ML results of the regression algorithms for predicting the sum of the daily yield of all solar panels at the generating plants. Average and standard deviation from 5-fold cross-validation is reported. The ML model with the lowest RMSE is shown in **bold**.

Model Type	RMSE (kW)	MAE (kW)	R^2
Linear Regression	$3.15 * 10^4 \pm 5.98 * 10^3$	$2.40 * 10^4 \pm 433.52$	0.702 ± 0.013
ANN	$1.32 * 10^4 \pm 5.19 * 10^3$	$8.01 * 10^3 \pm 288.42$	0.948 ± 008
Random Forest	$1.23 * 10^4 \pm 4.10 * 10^3$	$6.71 * 10^3 \pm 282.06$	0.955 ± 0.004
Extra Trees Forest	$1.41 * 10^4 \pm 3.75 * 10^3$	$8.25 * 10^3 \pm 153.55$	0.941 ± 0.004
Gradient Boosted Forest	$\mathbf{1.05} * 10^4 \mathbf{\pm 3.91} * 10^3$	$\mathbf{5.68} * 10^3 \mathbf{\pm 163.87}$	$\mathbf{0.967 \pm 0.004}$

To visually illustrate the differences between the machine learning predicted and true cumulative total daily yield values, one fold of cross-validation results of predicted total daily yield (kW) vs. true data values are shown in Figure 4. Note that a completely accurate prediction would form a line of identity, where the predicted and actual values align.

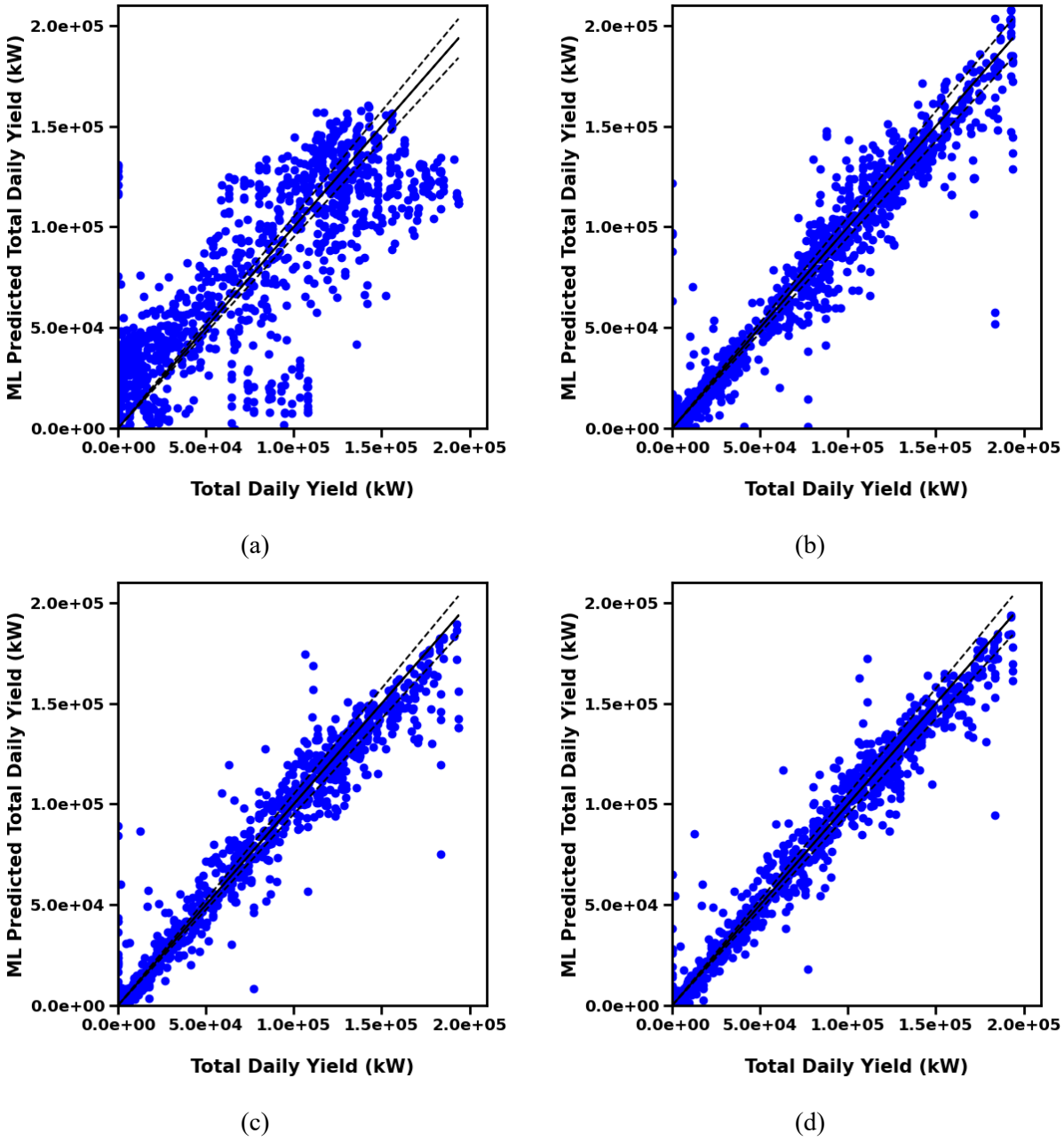

Figure 5: ML regression of total daily yield of all solar panels (kW) from regressors: (a) Linear, (b) Artificial Neural Network (ANN), (c) Random Forest (RF), and (d) Gradient Boosted Forest (GBT). The solid line represents the line of ideality, and the dashed lines represent a ±5% bound.

Clearly, the linear regression model is unable to predict the high total daily yield that occurs at the end of the day just before sunset. On the other hand, the ML models predicted total daily yield values are much closer to the true data points, regardless of the true total daily yield (kW). This regression represents a successful first step in the future prediction of solar power plant output given known or forecasted weather sensor data. Using ML learners such as these, plant operators can better predict how much additional energy they will need to purchase each day and provide greater reliability to their customers.

3.2 Clustering Results

The hyper-parameter for clustering was the optimum K value, which was found using the elbow point plot given in Figure 5.

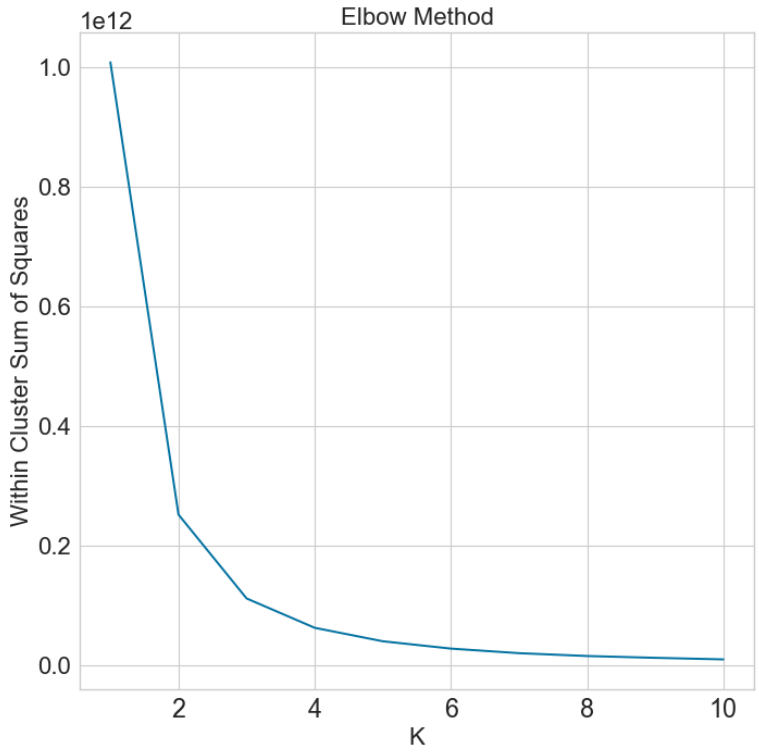

Figure 6: Elbow plot

The optimum value of K came out to be 3 from this plot. Using this value, K-means clustering was performed. The labeled data and cluster centroids per the K-Means model were plotted to visualize the relationship between the chosen parameters. The graph along with their centroids is plotted in Figure 6.

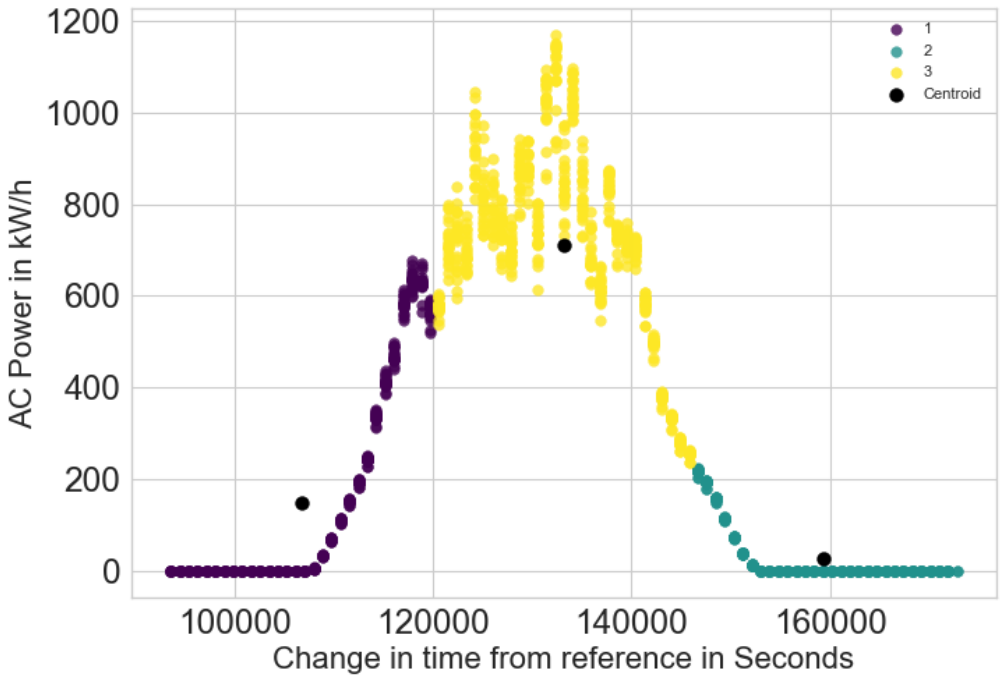

Figure 7: K-Means along with clusters

Time frames presented in Table 3 correspond to the respective clusters of power generation.

Table 3: Clustering data Inference. Relationship between time and power label

Time	Power Label	Centroid Location [x, y]
12 AM - 9 AM	Medium Yield	[1.06e+05 , 1.48e+02]
9 AM - 4 PM	High Yield	[1.33e+05 , 7.10e+02]
4 PM - 12 AM	Low Yield	[1.59e+05 , 2.76e+01]

So there are 3 different time zones clustered out by the K-Means that correspond to different power labels. It is notable that 9 AM to 4 PM was found to be the best time to operate the solar panels for highest power generation, and the second highest comes in the 12 AM to 9 AM window with a medium power level yield, and finally 4 PM to 12 AM had the lowest power yield. These findings suggest that all maintenance activities should be scheduled in the low yield window (the evening) in order to get the highest yield from the plant.

3.3 Classification Discussion

A solar panel may not produce energy due to the presence of shade, panel damage, inverter failures, or an unclean panel surface. If these defects can be detected, the downtime of solar panels can be reduced, and

hence maximize energy output. Such fault detection is a classification task that can be done using machine learning. Unfortunately, classification could not be performed from this dataset because the number of faulty points is too low to reach conclusive training for common machine learners. One solution would be to find enough points to train a model to classify panels as working or faulty. Another option for future studies would be to employ machine learners which specialize in imbalance data sets, where the majority of the data points are one class, but it is most important to identify the few outlier classes - such as machine learners that can predict fraud **[23]**. To identify inverter damage, continuous-time voltage and current data is needed. For the case where there is a presence of shade over a few panels, the system should not send out an alert informing a fault, but rather it should understand that it may be due to the clouds blocking direct sunlight. Finally, for the case where the panels need cleaning due to the collection of dust on the panels' surface, labeled maintenance data indicating unclean panels would help to train machine learners to detect the reduced power output over time.

4.0 Conclusions

Several key concepts of machine learning were shown through this report. The importance of pre-processing the data and converting it into a format that would be understandable by the machine learning models was found when using date and time as inputs. There was a need to normalize the data due to a mismatch in scale between the inputs and outputs. Hyperparameters and their tuning clearly affected the accuracy of the regression and the groupings of the clustering results. It was found that random coarse tuning and grid-based fine tuning were time-efficient for the regression models, while the K-means clustering models benefited from a visual elbow plot to determine the best K value. Different machine learning algorithms like artificial neural networks, random forest, gradient boosted forest, and K-means clustering were successfully utilized to produce meaningful solar power output data. Artificial neural networks, random forests, and gradient boosted trees were able to accurately regress the cyclical, non-linear, cumulative total daily yield (kW) given weather data to forecast power output. Subject-matter research on the topic of solar power plants and their operation were applied to justify the three clusters found. From these clustering results, the middle part of the day was identified as when the power output reaches a peak. This result suggests that maintenance should be scheduled in the evenings when power output is at a minimum, and thus maximizing the output from a solar power plant. Finally, classification of the individual solar panels into faulty and normal categories were discussed in theory, but were not able to be performed due to a lack of necessary data to train the machine learning classifiers.

5.0 References

[1] F. Jawaid and K. Nazir-Junejo, "Predicting daily mean solar power using machine learning regression techniques," *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, Dublin, pp. 355-360, 2016. DOI: 10.1109/INTECH.2016.7845051.

[2] A. Kannal, "Solar Power Generation Data," *Kaggle*, 18-Aug-2020. [Online]. Available: https://www.kaggle.com/anikannal/solar-power-generation-data. [Accessed: 10-Dec-2020].

[3] "datetime - Basic date and time types," *datetime - Basic date and time types - Python 3.9.1 documentation*. [Online]. Available: https://docs.python.org/3/library/datetime.html. [Accessed: 10-Dec-2020].

[4] "Python Course," *Python Tutorial: A Tutorial*. [Online]. Available: https://www.python-course.eu/python3_time_and_date.php. [Accessed: 10-Dec-2020].

[5] "sklearn.preprocessing.StandardScaler," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html?highlight=standardscaler. [Accessed: 10-Dec-2020].

[6] T. Oey, S. Jones and G. Sant, "Machine learning can predict setting behavior and strength evolution of hydrating cement systems," *Journal of the American Ceramic Society,* vol. 103, no. 1, pp. 480-490, 2020.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[8] B. F. Huang and P. C. Boutros, "The parameter sensitivity of random forests," *BMC Bioinformatics,* vol. 17, pp. 331, 2016.

[9] Keras. Team, "Keras documentation: The Sequential class," *Keras*. [Online]. Available: https://keras.io/api/models/sequential/. [Accessed: 10-Dec-2020].

[10] "sklearn.ensemble.RandomForestRegressor," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html?highlight=random+forest. [Accessed: 10-Dec-2020].

[11] "sklearn.ensemble.GradientBoostingRegressor," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html. [Accessed: 10-Dec-2020].

[12] "sklearn.ensemble.ExtraTreesRegressor." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html?highlight=extra+trees. [Accessed: 10-Dec-2020].

[13] Y. Li and H. Wu, "A Clustering Method Based on K-Means Algorithm," *Physics Procedia*. Vol 25, pp. 1104-1109, 2012. DOI: 10.1016/j.phpro.2012.03.206.

[14] B. Kalantari,"The State of the Art of Voronoi Diagram Research," *Transactions on Computational Science XX. Lecture Notes in Computer Science*, vol 8110, 2013. Springer, Berlin, Heidelberg.

[15] M. Romanycia and F. Pelletier, "What is a heuristic?," *Computational Intelligence,* vol. 1, pp. 47-58, 1985. DOI: 10.1111/j.1467-8640.1985.tb00058.x.

[16] T.K. Moon, "The expectation-maximization algorithm," *Signal Processing Magazine*, IEEE. vol 13, pp. 47-60, 1996. DOI: 10.1109/79.543975.

[17] J. Krithika Datta, "Normal Distribution," *Journal of Conservative Dentistry*, vol. 17, pp. 96-97, 2014. DOI: 10.4103/0972-0707.12417

[18] J. Zambrano, "Gaussian Mixture Model - method and application". 2017. DOI: 10.13140/RG.2.2.32667.77602.

[19] A. Zeng and Y. Huang "A Text Classification Algorithm Based on Rocchio and Hierarchical Clustering" *ICIC 2011: Advanced Intelligent Computing,* vol. 6838, pp. 432-439, 2011. DOI: 10.1007/978-3-642-24728-6_59.

[20] M. Aamir and S. Zaidi, “Clustering-based Semi-Supervised Machine Learning for DDoS Attack Classification,” *Journal of King Saud University - Computer and Information Sciences*, 2019. DOI: 10.1016/j.jksuci.2019.02.003.

[21] R. Nijman, Roeland. “Automatically and real-time identifying malfunctioning pv systems using massive on-line PV yield data,” 2018.

[22] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research,* vol. 13, pp. 281-305, 2012.

[23] C. Su, S. Ju, Y. Liu, and Z. Yu, “Improving Random Forest and Rotation Forest for highly imbalanced datasets,” *Intelligent Data Analysis,* vol. 19, pp. 1409-1432, 2015.