SolarPACES 2022, 28th International Conference on Concentrating Solar Power and Chemical Energy Systems

Analysis and Simulation of CSP and Hybridized Systems

https://doi.org/10.52825/solarpaces.v1i.651

Published: 11 Dec. 2023

Informed Feature Selection for Data Clustering of CSP Plant Production

Matthew J. Tuman[1[https://orcid.org/0009-0003-8772-051X]],
and Michael J. Wagner[2[https://orcid.org/0000-0003-2128-4658]]

[1] Graduate Student, Department of Mechanical Engineering, University of Wisconsin-Madison

[2] Ph.D., Assistant Professor, Department of Mechanical Engineering, University of Wisconsin-Madison. 1500 Engineering Drive, Madison, WI 53706

Abstract. To make concentrating solar power (CSP) more cost competitive, rigourous optimizations must be run to improve plant design and operations. However, these optimizaitons rely on time consuming annual simulations that solve an electricity dispatch scheduling problem to maximize plant revenue. To reduce the runtime of annual dispatch simulations of CSP plants, a data clustering approach is utilized. This approach assumes that like days of revenue and electricity generation can be identified using weather and price data. Although weather and price are important factors for electricity production, this work investigates how thermal energy storage (TES) inventory at the beginning of a day, denoted as S_i, can be used as a supplemental feature to group like days. A framework for creating and training a deep neural network to predict S_i is proposed. This model is validated and assessed using eleven sets of testing data that were not used during training. Then, the data clustering approach is performed three seperate times with features of weather and price along with either S_i from the neural network, S_i from the full annual simulation, or no S_i. Ultimately, the results suggest that using S_i as an additional clustering feature improves the data clustering simulation accuracy by 1.4%.

Keywords: Data Clustering, Machine Learning, System Modeling, Performance Simulation

1. Introduction

The main benefit of concentrating solar power (CSP) technology is its ability to store collected thermal energy for use later in the power cycle. This allows the plant to dispatch electricity when the demand or price of electricity is high as opposed to relying solely on solar availability. However, on a levelized cost of electricity basis, CSP is currently more expensive than other renewables such as photovoltaic and wind turbine technology. To make CSP more cost competitive, optimizations of plant design and operations must be performed. These design optimization studies rely on hundreds or thousands of annual simulations of plant production that include electricity dispatch scheduling optimization, and therefore, they often take a considerable amount of time to run. To reduce runtime, a data clustering approach can be utilized [1]. This method assumes that there are collections of days that share similar profiles of electricity generation which can be identified using weather and price data. By simulating one representative for each group and stitching the representative days together, the total runtime can dramatically decrease. In addition to limiting the number of days simulated, each representative group is independent of the other and can thus be simulated in parallel. The method relies on the notion that there exist groups of days that share common plant generation characteristics, but the ability to identify these collections of like days using weather and price data is not always straightforward.

Previous work by Martinek and Wagner [1] has shown that this clustering approach is feasible and accurate using price and weather data to group like days of generation together. While weather and price heavily influence a plant's generation behavior, an additional factor of importance is the amount of thermal energy storage (TES) inventory that is available at the beginning of the day which will be referred to as S_i. For example, consider two days that have identical solar availability and market demand but only one of the days has nonzero TES inventory carried over from the prior day. This initial TES inventory will allow the plant to ramp up its power cycle and reach its rated power load before the first hours of sunlight. Clearly, these two days of production differ in that the day with initial TES inventory can produce more electricity. This scenario presents a problem for the data clustering approach as these two days appear to be identical through the lens of weather and price even though they have different amounts of revenue generated. Thus, this paper proposes that S_i, in addition to weather and price, should be used as features. In this work, features refer to the quantifiable properties of a day that are used as inputs for a clustering algorithm. Unfortunately, obtaining S_i is not as simple as forecasting weather or price data. Like plant generation, S_i is a function of electricity price, weather, design specifications of the plant, and prior day operations. Because of the numerous factors impacting S_i, it is a challenge to accurately predict it throughout the year. To solve this problem, a deep neural network is utilized which predicts S_i using available weather and price data.

In this work, annual CSP plant simulations are performed using the System Advisor Model (SAM) developed by NREL [2]. Specifically, the Molten Salt Power Tower module [3] is used along with a thermal energy dispatch optimizer that solves the hourly production schedule for a day based on a 48-hour time horizon of price and weather data [4]. Multiple full dispatch simulations are utilized to compare the accuracy of the clustering approach. This work introduces the framework of the data clustering approach, provides a description of the deep neural network used to predict S_i, and presents a case study that suggests using S_i as an additional feature for clustering improves the accuracy of the simulation.

2. Methodology

2.1. Data Clustering Approach

At a high level, the data clustering approach follows the steps outlined below. Each step is described in more detail in the following section.

1) Partition the year into discrete time blocks
2) Characterize each time block according to features of interest: e.g. weather, price, and S_i
3) Apply weighting factors to the features to adjust relative importance
4) Use a data clustering algorithm to identify a user-specified number of clusters
5) Use SAM and the Molten Salt Power Tower module to simulate a single exemplar from each cluster
6) Reassemble the exemplar simulations into a full year according to cluster size

The first step is to divide the year into partitions consisting of one previous day, two operating days, and one next day. This process is illustrated in Figure 1. Although only the first two partitions are visualized in Figure 1, the entire year is broken up into these partitions of four days. The decision to partition the year into these four-day groups is based upon [1].

Next, to obtain data features for the clustering algorithm, weather, price, and S_i features are assigned for each partition. Each day type (Previous, Operating, and Next) is divided into N_{div} equal time divisions. This is illustrated in Figure 2, where $N_{div} = 3$.

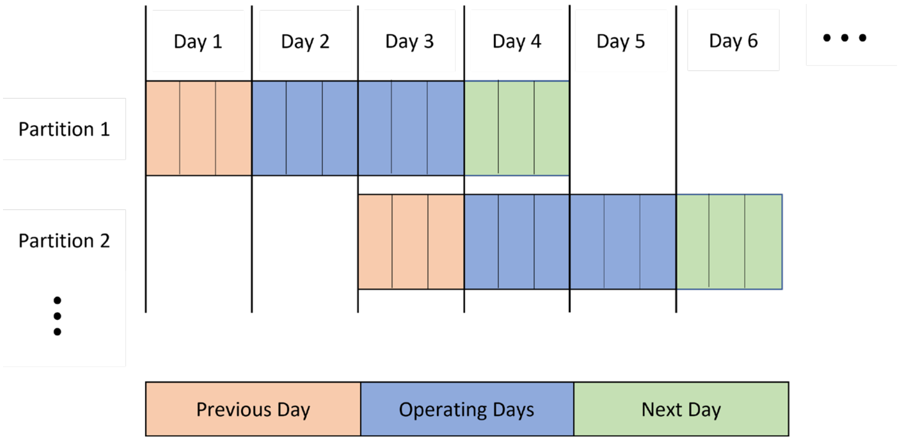

Figure 1. Illustration of the year partitioned into groups consisting of one previous day, two operating days, and one next day.

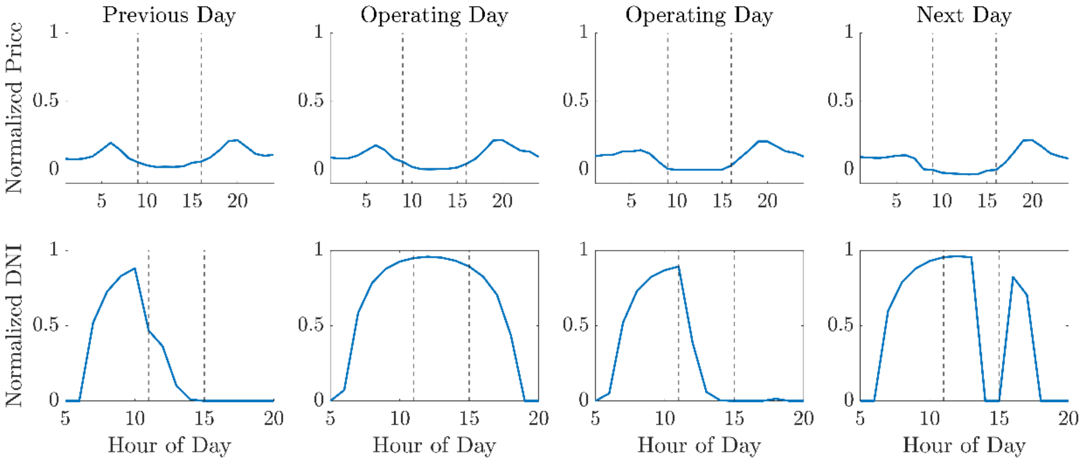

Figure 2. Normalized weather and price data for an individual partition.

Note that the equal divisions for DNI are formed using summer daylight hours, and the divisions for price are created using all twenty-four hours of the day. The average weather and price value of each division is then computed using data that is normalized by the maximum value of each respective feature. Lastly, S_i for the first operating day, normalized by the annual maximum value, is computed using the neural network described in the next subsection.

With the features defined for each partition, the next step is to apply a weighting factor. Because the data used to create the features is normalized to a maximum of one and the clustering algorithm is based on distance, multiplying certain features by a weighting factor can increase their relative importance when forming clusters.

$$x_{p,f}^{w} = w_f * x_{p,f} \tag{1}$$

Here, $x_{p,f}^{w}$ is the weighted data feature corresponding to the f^{th} feature and the p^{th} partition. This value is computed using the unweighted data feature $x_{p,f}$ and the weight for the f^{th} feature, w_f.

The fourth step uses the weighted data as input for a data clustering algorithm. This work utilizes affinity propagation [5] which forms clusters based upon Euclidean distance between datapoints. This iterative algorithm is particularly useful because it also computes exemplars which are the most representative individual of a cluster. Ultimately, this algorithm will form clusters that contain four-day partitions (from Step 1) that have similar values of weather, price, and S_i. The algorithm additionally determines which partition within each cluster is the most representative.

After using the affinity propagation algorithm, each four-day exemplar is then simulated using SAM and the Molten Power Tower module. The previous day is included in the simulation to regulate the TES inventory such that the stored thermal energy at the beginning of the first operating day will mimic what is computed in the full dispatch simulation. Furthermore, the next day is included to allow the dispatch optimization to solve its scheduling problem based on the 48-hour time horizon. Because each exemplar simulation is independent of the other, these run in parallel which also reduces runtime.

Lastly, the exemplar simulation results are reassembled to form an annual simulation. For each partition belonging to a specific cluster, the electricity and revenue generation time series for the two operating days is filled in with the output from the respective exemplar simulation. This accounts for almost the entire annual simulation. However, as illustrated in Figure 1, the first day is not accounted for by an operating day, and the last two days of the year will also not be accounted for. Thus, simulations of these unrepresented days are performed, and the output is used to complete the annual simulation.

2.2. Neural Network for Initial TES Inventory

A deep neural network is utilized to solve the regression problem of predicting S_i throughout the year. Because the dispatch optimization solves over a 48-hour time horizon, two days of weather and price data are used to estimate the TES inventory at the 24-hour mark. This is illustrated in Figure 3.

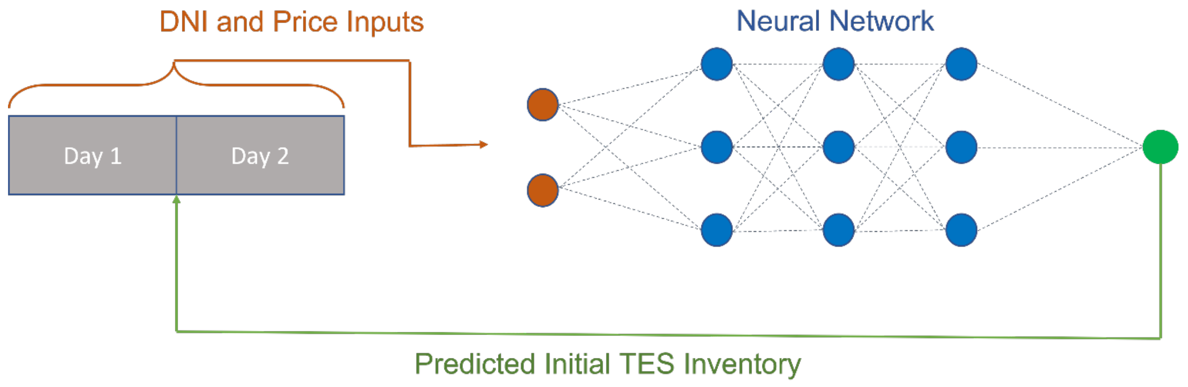

Figure 3. Visualization of neural network workflow to predict S_i at the start of Day 2 (Note: the neural network does not represent the architecture used).

Similar to Step 2 in the data clustering approach (Figure 2), each day of input is broken up into N_{div}^{NN} divisions based upon summer daylight for DNI or the full 24-hour day for price. However, instead of computing the average of the normalized data, the integral of the unnormalized DNI and price is used. The decision to use unnormalized data was made such that a trained neural network could be used for various weather or price files. Nonetheless, because the integrated DNI magnitude is considerably larger than that of price, the integrated DNI for each division is divided by 100. Scaling the inputs to a similar magnitude is important because neural networks adjust their weights via gradient descent. The step-size at which the weights are changed during gradient descent is proportional to the input value. Thus, inputs with larger magnitudes will have weights that change more drastically between iterations. With DNI and price inputs having similar magnitudes, the neural network can “learn” at the same rate which often decreases the number of iterations required to obtain a near optimal solution.

The neural network architecture was constructed using the Keras module from the python package TensorFlow. The network was constructed with 7 hidden layers consisting of 50, 25, 10, 25, 50, 25, and 10 nodes respectively along with one output layer that contains a single node. For all the hidden layers, the ReLu activation function is utilized, while the output layer uses a linear activation function. The architecture of the network was chosen following the conventional wisdom that many hidden layers with few nodes typically outperforms a network that has few hidden layers with many nodes. Additionally, the authors found that this architecture could be reliably trained for numerous plant designs and weather files.

3. Case Study and Results

The following case study was conducted to determine (1) whether a trained neural network can accurately predict S_i for sets of data that were not used to train the neural network (2) whether using the predicted S_i from a neural network as a feature improves simulation accuracy, and (3) the extent to which the (imperfect) neural network approach can be improved by comparing it to perfect prediction.

Weather data for Daggett, CA was acquired using NREL's National Solar Radiation Database (NSRDB). For this location, a typical metereological year (TMY) file was downloaded along with the measured data from odd years ranging from 1999 to 2019. The electricity price data is a locational marginal price (LMP) time series taken from the Califorinia Independent Operator (CAISO) database [6] for the "IRONMTN_2_N001" node from 2019-2020. A price multiplier profile is computed by dividing the hourly LMP values by the average LMP over the full year. Then, using the weather and price data as inputs to the Molten Tower Module in SAM, twelve annual dispatch simulations were run with the intent to serve as a comparison for the data clustering approach simulations. The plant design used in all simulations has 10 hours of TES capacity and a rated power of 65 MWe (650 MWe-hr TES capacity) .

For this case study, the results from the TMY weather file simulation are used to train the neural network, and N_{div}^{NN} is set to four which results in sixteen input values. Then, the remaining eleven simulations are used to assess the accuracy of the neural network's prediction and the accuracy of the data clustering approach using features of weather and price along with either (i) no S_i, (ii) using neural-network-predicted S_i, or (iii) using the simulated S_i taken from the full dispatch simulatio (what the neural network is trying to match). The weighting factors and divisions for each clustering approach are provided in Table 1. The weights for the S_i case studies are chosen based on the assumption that similar days of revenue generation can be identified using S_i along with the price and weather for the two operating days. However, for the no-S_i case study, there is an added emphasis on the features of the previous day. Specifically, the weights for previous day price and weather are increased from 0.1 to 0.5. This change in weights attempts to increase the relative importance of prior day operations with the hope that similar operations will yield similar S_i. For the three case studies investigated, 30 sets of clusters are used for the simulations.

Table 1. The weights and divisions used for each feature used in the data clustering approaches in the format (Weight (w_f) / N_{div}).

Case Study	Initial TES	Ave. DNI Operating Days	Ave. Price Operating Days	Ave. Clearsky DNI Operating Days	Ave. DNI Previous Day	Ave. Price Previous Day	Ave. DNI Next Day	Ave. Price Next Day
No TES	NA / NA	1 / 4	1 / 4	1 / 4	0.5 / 2	0.5 / 2	0.5 / 2	0.5 / 2
Predicted TES	0.75 / NA	0.95 / 4	0.75 / 4	0.6 / 4	0.1 / 2	0.1 / 2	0.1 / 2	0.1 / 2
Perfect TES	0.75 / NA	0.95 / 4	0.75 / 4	0.6 / 4	0.1 / 2	0.1 / 2	0.1 / 2	0.1 / 2

3.1. Neural Network Validation

As mentioned previously, the neural network is trained using the data from the TMY simulation and then used to predict S_i for the remaining weather files. Here, testing data refers to data that was not used to train the neural network. A comparison of the neural network performance for trained data and for testing data is presented in Figure 4.

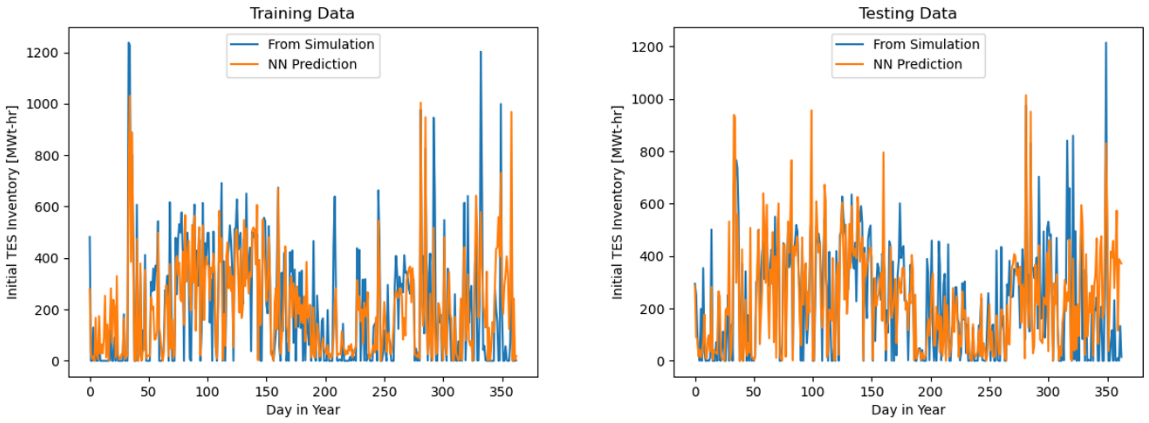

Figure 4. Neural network predictions of S_i throughout the year for the weather file it was trained with (left) and a weather file it was not trained with (right).

If the neural network performed perfectly, it would match the blue line. Thus, as shown in Figure 4, the imperfect neural network predicts S_i relatively well for the training and testing dataset. To quantify the performance of the neural network, Equation 2 is used.

$$error_{TES} = \frac{1}{n}\sum_{i=1}^{n}\frac{|S_{i,sim}-S_{i,NN}|}{TES_{max.\ capacity}} * 100 \quad (2)$$

This metric computes the average, absolute error between the simulated and predicted S_i as a percentage of the maximum TES capacity. For the training data, the error is 7.4%, and the average error for the eleven sets of testing data is 8.3%. The slight increase in error from the training to testing sets suggests that the trained neural network translates well to untrained data. Thus, although the predicted S_i is not perfect, it should still be able to serve as a reliable estimate when used as a feature for clustering. The inability to predict high and zero S_i days will be discussed in the future work section.

3.2. Case Study Findings

With the neural network validated, simulations of the eleven testing weather files are performed using the data clustering approach. Three data clustering simulations are run for each weather file corresponding to the three sets of features outlined in Table 1. The total, absolute percent error between revenue units of the full dispatch simulation and the data clustering approach is computed to compare the accuracy between the clustering approaches.

$$\%\ error = \frac{|revenue_{full} - revenue_{cluster}|}{revenue_{full}} \cdot 100 \tag{3}$$

In addition to the total percent error of revenue, the mean squared error of daily revenue is computed. The average error metrics for the eleven simulations are provided in Table 2.

Table 2. Average error metrics for the three data clustering approaches.

Metric	Average Value
No S_i: MSE [Revenue Units]^2	30882
Predicted S_i: MSE [Revenue Units]^2	30538
Perfect S_i: MSE [Revenue Units]^2	26938
No S_i: Percent Error [%]	4.67
Predicted S_i: Percent Error [%]	3.86
Perfect S_i: Percent Error [%]	3.23

Table 2 reveals the highest error for the two metrics corresponds to the clustering approach when S_i was not used as a feature. Furthermore, the lowest error corresponds to the clustering approach when perfect S_i, taken from the full dispatch simulation, was used as a feature. Interestingly, the clustering simulations using the predicted S_i from the neural network has average errors that fall in between the two extremes. As seen in the previous subsection, the accuracy of S_i from the neural network is not perfect, and the results suggest this decreases the accuracy of the clustering approach. Nonetheless, including S_i as a feature still improves the accuracy of the simulation compared to the case where it is not used.

4. Conclusion

This work utilizes a data clustering approach that can decrease the runtime of an annual simulation by reducing the number of days simulated and by introducing the possibility for parallel computing. The contribution of this work is the investigation of using initial TES inventory as a feature for clustering, in addition to weather and price, to improve model accuracy. A framework for estimating S_i using a deep neural network is presented. This framework uses weather and price over a 48 hour horizon as input and trains the network with simulation output for a location of interest. A case study is presented wherein the accuracy of the neural network is evaluated along with the effects of using S_i as a feature for the data clustering approach. After training the network with TMY data for Daggett, CA, the network performed remarkably well in that the accuracy of S_i only slightly degraded between the training data and the eleven sets of testing data. Next, the data clustering approach was used to perform simulations for eleven weather files for the three cases when S_i was not used as a feature, when the neural network predicted S_i was used as a feature, and when S_i taken from the full dispatch simulation was used as a feature. The simulation accuracy was assessed with two error metrics, and the results suggested that using S_i as a feature can improve the results of a clustered simulation. However, because the neural network was unable to perfectly predict the initial TES inventory throughout the year, these simulations were not as successful as the simulations that used perfect S_i. Ultimately, this work suggests that using S_i as a feature can improve the data clustering simulation accuracy, and that a neural network can be used to accomplish this task.

While this methodology allows for significant reduction in runtime of a batch of multiple years of weather data, the necessity of running one full annual simulation to train the neural network remains. Future work should determine the minimum number of simulation days that are necessary to adequately train the network. Furthermore, the robustness of a trained neural network needs to be investigated. For example, it is unknown how well the neural network in this work would predict S_i for an annual simulation that uses a different price profile. Lastly, the neural network struggles to predict high and zero S_i days. Different inputs to the neural network along with various architectures should be considered to alleviate this problem.

Data availability statement

Data and models are available upon request made to the corresponding author.

Author contributions

Tuman: Conceptualization, Methodology, Investigation, Visualization, Writing – original draft. Wagner: Funding acquisition, Methodology, Investigation, Software, Supervision, Project administration, Writing – review & editing.

Competing interests

The authors declare no competing interests.

Funding

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number DE-EE0009803.

References

1. J. Martinek, and Michael J. Wagner. “Efficient Prediction of Concentrating Solar Power Plant Productivity Using Data Clustering.” Solar Energy 224. June (2021): pp. 730–41. https://doi.org/10.1016/j.solener.2021.06.002.
2. Blair, N., Dobos, A.P., Freeman, J., Neises, T., Wagner, M., Ferguson, T., Gilman, P., Janzou, S., 2014. System Advisor Model, SAM 2014.1.14: General Description. Report. National Renewable Energy Laboratory, NREL/TP-6A20-61019.
3. Wagner, M.J., 2008. Simulation and predictive performance modeling of utility-scale central receiver system power plants. Thesis. University of Wisconsin-Madison.
4. Wagner, M.J., Newman, A.M., Hamilton, W.T., Braun, R.J., 2017. Optimized dispatch in a first-principles concentrating solar power production model. Applied Energy 203, 959-971. https://doi.org/10.1016/j.apenergy.2017.06.072.
5. Frey, B.J., Dueck, D., 2007. Clustering by passing messages between data points. Science 315, 972-976. https://doi.org/10.1126/science.1136800.
6. CAISO, 2018. California ISO Open Access Same-time Information System (OASIS). URL: http://oasis.caiso.com/mrioasis/logon.do