JOURNAL OF
EMERGING INVESTIGATORS

Article

A novel encoding technique to improve non-weather-based models for solar photovoltaic forecasting

Khondoker Farihah Ahmed[1], Mauricio Hernandez[2], Aven Satre-Meloy[3]
[1]American International School of Dhaka, Dhaka, Bangladesh
[2]Duke University, Durham, North Carolina
[3]Oxford University, Oxford, England, United Kingdom

SUMMARY

Several studies have applied different machine learning (ML) techniques to the area of forecasting solar photovoltaic power production. Most of these studies use weather data as inputs to predict power production; however, there are numerous practical issues with the procurement of this data. This includes the high costs of procurement and lack of backup techniques if communication with weather data services fail. These practical issues are not widely considered yet in the current literature. This study proposes models that do not use weather data as inputs, but rather use past power production data as a more practical substitute to weather-based models. Similar studies have shown satisfactory accuracies, but this study proposes a novel data preprocessing technique—cyclical features encoding—that we hypothesized would boost model accuracy significantly. We used ML techniques to predict power production in a 24-hour time horizon, using input data of the past 48 hours of power production. The Random Forest model offered the best results, with a Pearson Correlation Coefficient of 0.97 (11% higher than previous studies), Mean Absolute Error of 0.0266 (60% better than previous studies), and Root Mean Squared Error of 0.0773 (38% better than previous studies). These results are comparable to state-of-the-art weather models in the field. Our proposed models demonstrate a better, cheaper, and more reliable alternatives to current weather models.

INTRODUCTION

In the last few years, innovations and research in the field of photovoltaic (PV) solar cells have led to a large increase in their efficiency (1). With the recent onset of climate change, PV cells are of special interest globally, as they are able to produce high amounts of energy with low installation costs and without emitting greenhouse gases (2). However, as the widespread integration of PV technology into the electrical grid is being considered, a key challenge to solve is solar power's intermittency (3). Like most renewable energy, solar PV technology is dependent on weather conditions, making its power production highly volatile and unreliable compared to other traditional energy sources such as fossil fuels, where humans control the amount of energy production by burning fossil fuels to match power consumption needs (3). This intermittency of solar energy poses large barriers to controlling and planning in energy systems. With the expansion of the electrical grid, especially as a result of the increasing number of decentralized renewable energy suppliers, it is highly important to forecast power production (the supply side) as well as forecast power consumption (the demand side) in order to ensure smooth grid scheduling (4). Therefore, forecasting models that can predict future solar power production and consumption are important in solving the intermittency problem; in fact, the progress of all other forms of intermittent renewable energy are also dependent on such models (4, 5).

In the past two decades, there has been increasing interest in the intersecting area of artificial intelligence (AI) and energy systems research. AI can deliver data-driven forecasts and perform predictive modelling for energy production, consumption, and system maintenance, as well as other key aspects of grid scheduling, all with a high accuracy and speed (4, 6).

For AI models that forecast solar PV power production, the most recently proposed state-of-the-art models in the literature primarily use weather data as inputs (3). These model inputs can include data such as solar radiation, temperature, pressure, different types of precipitation, humidity, cloud coverage, wind speed and direction, and other types of relevant meteorological data (7). However, while most PV power production forecasting models in the literature are primarily reliant on weather data, there are essential issues with obtaining weather data on a regular basis, which renders these models impractical. These types of weather data must be purchased regularly from weather service companies by energy providers, generating high costs for the use of such a model (4, 8). More 'simple' weather data, such as forecasts of the future temperature of an entire region or forecasted probabilities of precipitation in the region are sometimes offered free of charge. However, more 'complex' data that is highly specific or requires high accuracy, such as forecasted solar irradiation of a PV system at a particular location at a highly specific time, has to be purchased at a high cost. Obtaining this data on a daily, hourly, or sub-hourly basis increases these costs even more. Furthermore, constant communication with weather services is needed to obtain this data, especially for forecasts with smaller time increments. If communication failures between weather services and energy providers do occur, such as internet outages, weather-based models can then no longer be used to predict power generation. This poses immensely large risks to grid operations and scheduling (3).

However, there is another type of input data that can be used as a potential better alternative in power production forecasting models: past power production data. This is because past power production data is freely available

to energy providers, as they are the ones producing this data. Furthermore, it is more reliable as no communication is needed with external services. Also, the past power production data has a high, direct correlation with weather data, meaning that it can be used as a potentially effective substitute (8). Previous studies have shown that simple models that use past power production data only can indeed yield satisfactory results; however, they are not yet comparable in accuracy to the current state-of-the-art weather models in the field using weather data (3). Thus, it has been proposed by these studies that such non-weather-based models only be used as a backup or an emergency solution to traditional weather-based models (3, 8). One such study by Ordiano *et al.* used past power production data and time as inputs only, predicting the next 24 hours of power output using the past 48 hours of power output data (3). This study was highly consequential in the field because it was one of the first to point out that practicality should be a key factor considered in the development and evaluation of PV power forecasting machine learning (ML) models. The authors examined four simple polynomial techniques and two artificial neural network (ANN) models that yielded quite low errors: a mean absolute error (MAE) from 0.0664 to 0.0725, a root mean squared error (RMSE) from 0.1247 to 0.133, and Pearson Correlation Coefficients (PCC) from 0.8581 to 0.8765. However, current state-of-the-art weather-based models in the literature are able to perform better than these metrics and also account for extreme weather volatility (4). Hence, the authors proposed these non-weather-based models as only a backup offline solution in case of communication failures with weather services (3). This study was one of the first studies in this area of research to recognize practicality as a key factor to evaluating a model (thus its approach to using non-weather data) (3). Other works in the field usually consider complexity of model, user-friendliness, accuracy, and speed (4).

Our project builds on this previous work by using a novel data preprocessing technique to significantly increase the accuracies of non-weather-based models: cyclical features encoding of time features in the data. A non-weather-based model has two types of data in its inputs: past power production and time. In order to be processed by the ML model, the time input can be broken down into multiple relevant numerical inputs, such as day of the month, month number, etc. However, adding these time features to a ML model can increase the prediction error. This is because many of these time features are cyclic in nature, but the ML model interprets them as linear. For example, if the model is training on data from January 31st and February 1st, it is likely that their power production output are similar, as they are only 24 hours apart. Thus, a ML model should be able to detect a close relationship between the two dates' data. However, while considering the time features data for these two dates, the 'day of month' feature will be 31 and 1, respectively. This will cause the model to interpret the relationship between these two dates as highly dissimilar, which may decrease the accuracy of the model. To make the model understand cyclical time features, feature engineering methods such as cyclical feature encoding can be used. Mathematical functions like sine and cosine can transform linear time data into cyclical data. Thus, we predict that encoding cyclical features will allow for the model to better interpret the data, leading to a significantly higher accuracy in predictions.

The goal of this study was to investigate if the novel proposed technique of cyclical features encoding improve the accuracy of non-weather-based models, and whether they would be high enough to be a viable substitute to current state-of-the-art weather-based models. In order to directly build on previous results, we used the same methodology and the same dataset, and then tested our novel cyclical features encoding preprocessing technique to determine if the accuracy could be increased. We hypothesized that if cyclical features encoding is used for time features, then the model accuracy will increase significantly, because the model will be better able to interpret the cyclical nature of the time data. We used ML techniques to predict power production in a 24-hour time horizon using input data of the past 48 hours of power production. The Random Forest model offered the best results, with a PCC of 0.97 (11% higher than previous studies), MAE of 0.0266 (60% better than previous studies), and RMSE of 0.0773 (38% better than previous studies). These results are comparable to state-of-the-art weather models in the field, and thus our proposed models demonstrated a better, cheaper, and more reliable alternative to current weather models.

RESULTS

In this study, we evaluated the performance of seven different neural networks (NN), alongside various traditional models (**Table 1**). For the NNs, we changed two variables in the model architecture: the number of units in each of the dense layers (ranging from 8 to 1000 units), and the number of epochs (ranging from 10 to 30 epochs) (**Table 1**). The difference between NN6 and NN7 for instance is the number of units in each of the dense layers (**Table 1**). We used the freely-available Ausgrid dataset online (the same dataset used by Ordiano *et al.*), which included power production data at 30-minute temporal-resolution from the rooftop PV systems of 300 customers in Australia (hence, we had 48 data values for every 24 hours of data) (3).

We had two approaches in this study, finding *f(P)* (where the function *f*( ) of the past power production data input *P* represents the predicted value of future power production), and finding $f(P,M_{sin},M_{cos},D_{sin},D_{cos})$ (where *P* is the past power production data input, M_{sin} is the cosine of the numerical month value, M_{cos} is the sine of the numerical month value, D_{sin}

Model	Dense Layers	Number of units in the dense layers (between input and output layers)	Number of epochs
NN1	3	500 - 8	10
NN2	3	1000 - 8	10
NN3	3	1000 - 8	15
NN4	2	100 - 48	10
NN5	3	1000 - 8	20
NN6	3	100 - 1000	30
NN7	3	1000 - 8	30

Table 1: Neural Networks (NN) Model Architecture. This table explains the specific architecture of each of the seven models that this study tested. All of the models had 96 input units and 48 output units. All of the models used Mean Squared Error (MSE) as the loss function used to compile the model, and all the models used the ReLu activation function for all layers.

	f(P) *Without cyclically encoded time features*			f(P, Msin, Mcos, Dsin, Dcos) *With cyclically encoded time features*		
Model	**RMSE**	**MAE**	**PCC**	**RMSE**	**MAE**	**PCC**
LR	0.0964	0.0369	0.9277	0.0942	0.0358	0.9552
KNN	0.0937	0.0353	0.9333	0.0803	0.0269	0.9685
DT	0.1046	0.0422	0.9145	0.1132	0.0460	0.9345
MLP	0.0923	0.0374	0.9340	0.0903	0.0368	0.9590
RF	0.0922	0.0363	0.9335	0.0773	0.0266	0.9695
NN1	0.0912	0.0379	0.9365	0.0950	0.0448	0.9550
NN2	0.0914	0.0384	0.9348	0.0948	0.0453	0.9067
NN3	0.0947	0.0411	0.9349	0.0930	0.0443	0.9571
NN4	0.0913	0.0378	0.9342	0.0917	0.0390	0.9601
NN5	0.0935	0.0408	0.9584	0.0918	0.0427	0.9359
NN6	0.0888	0.0344	0.8360	0.1606	0.0562	0.9404
NN7	0.0895	0.0383	0.9662	0.0810	0.0354	0.9378
Average scores	0.0933	0.0381	0.9287	0.0969	0.0400	0.9483
Highest-scoring models	0.0888 (NN6)	0.0344 (NN6)	0.9662 (NN7)	0.0773 (RF)	0.0266 (RF)	0.9695 (RF)
Lowest-scoring models	0.1046 (DT)	0.0422 (DT)	0.836 (NN6)	0.1606 (NN6)	0.0562 (NN6)	0.9067 (NN2)

Table 2: Evaluation results for all models with *f*(*P*) and *f*(*P, M_{sin}, M_{cos}, D_{sin}, D_{cos}*). The three metrics that were used for evaluation are Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Pearson Correlation Coefficient (PCC). Other than the 7 neural networks (NN), the other models used were Linear Regression (LR), K-Nearest Neighbors (KNN), Decision Tree (DT), Multi-Layer Perceptron (MLP), and Random Forest (RF). The highest possible PCC is 1, the lowest possible is 0; the lowest possible MAE and RMSE is 0, the highest possible is infinity. PCC does not have units, while MAE and RMSE are measured in kilowatt-hours (kWh).

is the sine of the numerical day value, and D_{cos} is the cosine of the numerical day value—and the function *f*( ) represents the predicted value of future power production). For this latter approach (finding $f(P,M_{sin},M_{cos},D_{sin},D_{cos})$), we used the cyclical features encoding technique, by taking the sines and cosines of the month and day numerical values and adding it as a data input. Before taking the sine and cosine of the month and day values, we first normalized them into the pi scale (a scale from 0 to 2π). And before that, the same steps were taken to treat the data as Ordiano *et al*., including outlier detection and elimination, normalization, missing data treatment, and synchronization.

To evaluate our machine learning models, we used MAE, RMSE, and PCC. MAE and RMSE are error metrics in the same units, which allowed for comparisons between models and between the two data approaches in our study; whereas PCC (which measures strength of correlation between the predicted value and the actual value) specifically also allowed us to make cross-study comparisons with state-of-the-art models in the field used for the same purpose (this was helpful to see if our models were better, worse, or equal to the accuracies of state-of-the-art models). More details on these methods can be found in the Materials and Methods section.

For *f*(*P*) without cyclically encoded time features, the highest-scoring RMSE was 0.0888 from NN6, the highest-scoring MAE was 0.0344 from NN6, and the highest-scoring PCC value was 0.9662 from NN7 (**Table 2**). On the other hand, the lowest-scoring RMSE was 0.1046 from Decision Tree (DT), the lowest-scoring MAE was 0.0422 again from DT, and the lowest-scoring PCC was 0.836 from NN6 (**Table 2**). This latter result was slightly surprising, as NN6 had the highest MAE and RMSE scores. However, NN6 was the highest-scoring model overall, as it scored the highest in two out of the three evaluation metrics: RMSE and MAE. On the other hand, DT was the lowest-scoring model overall, producing the lowest scores in these same metrics. The average scores from all the models were an RMSE of 0.0933, a MAE of 0.0381, and a PCC value of 0.9287 (Table 2).

However, for $f(P,M_{sin},M_{cos},D_{sin},D_{cos})$ with additional cyclically encoded time features, the highest-scoring model was Random Forest (RF) for all metrics, with a RMSE of 0.0773, MAE of 0.0266, and PCC value of 0.9695. Conversely, the lowest-scoring RMSE was 0.1606 from NN6, the lowest-scoring MAE was 0.0562 from NN6, and the lowest-scoring PCC value was 0.9067 from NN2. Hence, the overall highest-scoring model was RF in this case, while the overall lowest-scoring model was NN6 in this case (as it scored the lowest in two out of the three evaluation metrics—RMSE and MAE). The average scores from all the models were an RMSE of 0.0969, a MAE of 0.0400, and a PCC value of 0.9483. Overall, this means that all the evaluation metrics showed that compared to f(P), the addition of cyclically encoded time features in $f(P,M_{sin},M_{cos},D_{sin},D_{cos})$ contributed to a dramatic improvement in two of the three metrics (where a lower RMSE and MAE are more desirable, and a higher PCC is more desirable). Compared to f(P), for the predictions by $f(P,M_{sin},M_{cos},D_{sin},D_{cos})$, the highest-scoring RMSE value was 13% better, the highest-scoring MAE was 23% better, and the highest-scoring PCC value was 0.34% higher. There was a large improvement in the lowest-scoring model's PCC value, with an improvement of 8.46%, from 0.836 to 0.907.

As we considered the study by Ordiano *et al.* as a baseline, we compared the highest-scoring models from their study with the highest-scoring models from this study (**Table 3**) (3). The improvement in percentages is shown. Note that a comparison is made for both the case where only past power production data was used by both studies, as well as the case where the additional time features were also used by both studies. However, for the latter, the method of using time features was markedly different for both studies: the present study made use of cyclical features encoding the month and day of month features, while the other study made use of a linear time of day feature (3). The results show that there was significant and considerable improvement in every single

Inputs used	Metric	Highest-scoring model from our study	Highest-scoring model from Ordiano *et al.* (3).	Improvement by our study
Using only past power production data	RMSE	0.0888	0.1247	-29%
	MAE	0.0344	0.0664	-48%
	PCC	0.9662	0.8765	+10%
Using both past power production data and time features	RMSE	0.0773	0.1247	-38%
	MAE	0.0266	0.0664	-60%
	PCC	0.9695	0.8765	+11%

Table 3: Comparison of our highest-scoring models with the highest-scoring models of Ordiano *et al.* (3). The three metrics that were used for evaluation are RMSE, MAE and PCC. -% for RMSE and MAE is an improvement, +% for PCC is an improvement.

evaluation metric from the baseline study, for all models (3). When considering the improvements between their models and our models, for using only past power production data, the RMSE was 29% lower, the MAE was 48% lower, and the PCC value was 10% higher. Furthermore, for using additional time features, an even higher improvement was found than the improvements when using only past power production data. In our study, the RMSE was 38% lower, the MAE was 60% lower, and the PCC value was 11% higher. Overall, there were dramatic improvements for all metrics.

DISCUSSION

The hypothesis of this study was that if cyclical features encoding is used for time features, then generally all models' accuracy will increase. The results of this experiment show that this approach can lead to a large decrease in errors generally in all models, and thus our hypothesis was shown to be correct. Overall, there are three main takeaways from the results. First, using additional cyclically-encoded time features in the models yielded significant improvements, versus using only past power production data. Second, our approach yielded large improvements from the baseline study by Ordiano *et al.*, for both cases of past power production data only and the additional time features, with the latter showing even more improvement than the former (3). Third, the highest accuracy achieved was 97%, from the RF model with cyclically-encoded features.

An example of one of our most high-scoring models is NN6 (**Figure 1**). One key finding from NN6 is that during the peak power production hours of each day, the model is the poorest at making predictions; on the other hand, it predicts all other times of the day relatively well. Another key finding is that the model has difficulty predicting volatile power production, as between Day 2 and 4, fluctuations in power production were not addressed by the model. This is likely due to the fact that the past power production data of 48 hours does not reflect weather volatility as well as direct weather data (**Figure 1**).

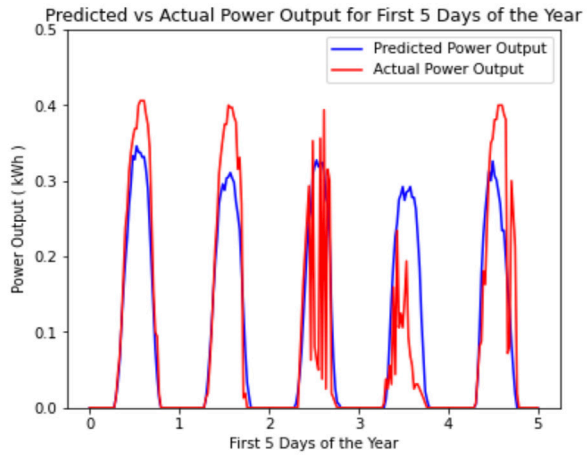

Figure 1: Example Forecast of Predicted Power Output vs. Actual Power Output. Neural Network 6 (NN6)'s predictions are shown for the first five days of the year. The model's predictions for power output (in blue) for all 24 hours of each day of the first five days of the year (from January 1 to January 5) are shown. They are compared with the actual power output values (in red). The results indicate that the model was able to capture the trends of the data during days with stable weather, while the model performed less well during days with more volatile weather.

It is important to address some additional reasons why we observed increased accuracy unrelated to the inclusion of cyclical time features. First, instead of using three years of Ausgrid data to train the model like Ordiano *et al.*, this study used only one year of data from 2012 to 2013. This was done in order to allow the model to catch more specific trends in seasonality of just the one year (3). This likely allowed for more accurate predictions, explaining the improvement without using cyclical features encoding. Second, while the study by Ordiano *et al.* used the data of 54 households from 300 due to missing values in their 3-year dataset, this study used all 300 households' data in its 1-year dataset. This allowed for our models to train on nearly six times more data, which may have contributed to these improvements (3). Third, the ML techniques used were different between the studies. Ordiano *et al.* used techniques that were simpler in nature and less diverse than the ones tested in this study (**Table 1**) (3). The techniques we used are some of the most commonly used in the field of ML and are known for their ability to produce accurate results in a very simple and efficient way. These better ML techniques are likely part of what led to the improvements from the baseline study without even considering the cyclical features encoding.

However, even with these improvements from changes in methodology, the singular, isolated effect of the novel cyclical features encoding technique can be seen in the staggeringly higher improvements from using just past power production data. The improvements from the baseline study without using cyclical features encoding were a 29% better RMSE, 48% better MAE, and 10% better PCC. However, with cyclical features encoding, the improvements from the baseline study were a 38% better RMSE, 60% better MAE, and 11% better PCC. All of these make it clear that using the novel technique of cyclical features encoding markedly improved the results. This is likely because the model was better able to interpret the cyclical nature of the time data with the engineered sine and cosine features, as opposed to the linear nature of the original time data.

There are some ways that these results could be further improved, and some considerations that need to be made about the proposed models. First, other cyclical time features could be added, like season. This has the potential to improve the results as it is another significant piece of data that could be feature encoded to be cyclical and inserted into the dataset. This is because seasons are another useful and cyclical piece of data. Second, performing other forms of feature engineering may potentially improve the results, such as adding maximum, minimum, average, median, or even the range of the power production of each day. These statistics could help the model further in interpreting the nature of the power production data. Third, it is important to understand that only for regions with extremely volatile weather conditions, these models may not be as effective, but there is a way to still use them and save the cost of purchasing expensive weather data. Other types of weather data can be obtained free of charge, such as future predicted temperatures of an entire region or forecasted probabilities of precipitation of an entire region. However, it is still likely that the proposed models of this study would work best in PV systems that have storage systems, so that prediction inaccuracies by the model can be easily compensated with energy saved in storage.

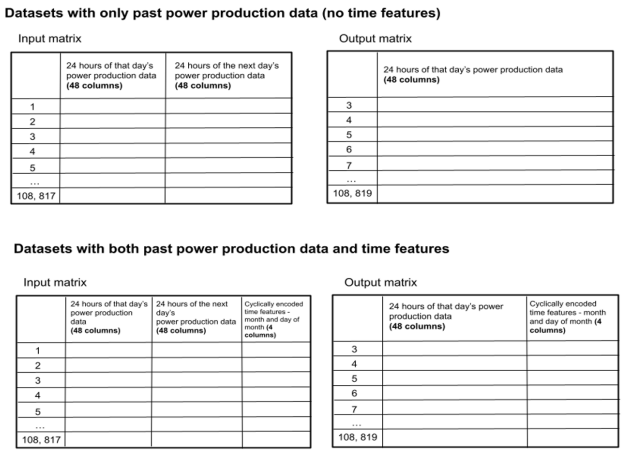

Figure 2: Visualization of the datasets. The main difference between the two datasets is that the first one does not have any cyclically encoded time features, while the second one does.

Fourth, though generally the models' predictions indicate low signs of overfitting (the models appear to be grasping the general trends of the data), future studies can improve the experimental design by including a validation split when splitting the data (**Figure 2**). Though this study used a 90:10 train/test split, a 60:20:20 or 80:10:10 train/test/validation split could also be used to be used to further improve the model's generalization abilities and to help it avoid overfitting. Fifth, another consideration this study did not make that may have an impact on the results is material sensitivity of PV panels to weather conditions—future studies could examine what role this plays in power output prediction and whether this creates a need for weather data.

Hence, there are three main notable consequences of this study. First, the proposed models are practical, cheaper, and more reliable alternatives to current state-of-the-art weather models. Second, the significant findings of this study highlight the importance of practicality in this field. If the practicality factor is not considered, such as how the data will be procured and whether or not procurement is realistic, then the models in the literature only have high accuracies in the literature—but no realistic chances for practical implementation in the real world. Thus, models that do not consider practicality are not useful to energy companies, and this study can serve as a catalyst for the consideration of the practicality factor in the field. Third, it is important to realize that weather data is used commonly not only in the forecasting of solar energy power production but also other forms of renewable energy that are dependent on weather conditions, such as wind energy. The same problems with weather-based models outlined in this paper for solar energy are true for wind energy. However, the solutions outlined in this paper, of using past power production data and a novel cyclical features encoding technique, can also be applied to create non-weather-based models for wind energy. The same premise of using cheap and readily-available forms of data like past power production data can also be applied to other forms of renewable energy, including hydro power and marine energy. The high accuracies found in this study show that there is a promising future in non-weather-based models and past power production data-dependent models for various forms of renewable energy power production forecasting that can act as a highly substantial cost-saver and a more reliable way of forecasting.

MATERIALS AND METHODS

Treatment of Ausgrid Solar Home Electricity Data

The "Ausgrid Solar Home Electricity Data" dataset was used, which is available online for free, published by the Australian energy provider Ausgrid (8). This data was used because it is also the dataset used in the study by Ordiano *et al.*, which is the baseline study that we compared our results with (3). The dataset includes power production data in kilowatts (kW) from the rooftop solar PV systems of 300 different customers in Australia. The data has a 30-minute temporal resolution, giving 48 values for every 24 hours.

Though in the study by Ordiano *et al.* data from July 1st, 2010 to July 30th, 2013 was used, our study only used data from July 1st, 2012, to June 28th, 2013 (3). We did this for two reasons: first, because the data for this period specifically did not have any missing values, thus simplifying the task and reducing the need to perform missing data treatment (unlike the 3-year dataset used by Ordiano *et al.*, which has many missing values and thus requires missing data treatment); second, the selection of this period allowed us to use the data of 300 households (more training data) instead of only 54 like Ordiano *et al.* (3).

The dataset format was changed from the original in order to better fit the structure needed for an ML model. Unnecessary features such as customer number, generator capacity, postcode, and consumption category were removed. The data was split into an input matrix and an output matrix. Two copies of the input and output matrices were created: one copy with the extra time features (month and day-of-month) and one copy without any time features. This simply means that, for example, for each row in the first copy, we included an additional column that indicates what month that particular time sample occurred in (numbers 1 through 12 to indicate January through December respectively), and also another additional column to indicate what day of month that time sample occurred in (numbers 1 through 31 to indicate the specific day of the month) (**Figure 2**).

The input matrix was then structured such that each row represented one day of power production data; the first 48 columns were that day's power production data; the second 48 columns (column 49 to 96) were the next day's power production data. For the input matrix with time features, month and day of month were also added as features, which will undergo cyclical features encoding later on. Thus, the input matrix without time features had 96 columns, and the input matrix with time features had 98 columns. The output matrix was structured such that each row represented one day of power production data; the first 48 columns were that day's power production data. For the output matrix with time features, month and day of month were also added as features. There were 108,819 total rows for both the input and output matrices. The input data was structured in this way so that 48 hours of past production data and optional time features could be used to predict the output—24 hours of that day's power production data (Figure 2). Note that the last two rows of the input matrix and the first two rows of the output matrix were removed for each customer, as there is nothing left to predict using the last 48 hours of data and there is no

available prior data to predict the first 48 hours.

Data Preprocessing (without cyclical features encoding)

The data was prepared using the same process described by Ordiano *et al.*, which is briefly summarized as follows: outlier detection and elimination, normalization, missing data treatment, and synchronization (3). We conducted everything aside from missing data treatment, which was not necessary as our period of the data did not have any missing values.

Cyclical Features Encoding

Though this is also part of the data preprocessing, this step was done after all the other preprocessing steps. From the two sets of data (with and without time features), only the data with time features went through this step. This step was not done by Ordiano *et al.*, as they only used past power production data and time inputs (but they did not feature-engineer the time inputs).

First, the data was converted to a pi scale as a form of normalization. If M is the month data, D is the day-of-month data, and the subscript n represents 'normalized data', then M_n is the 'normalized' month data, D_n is the 'normalized' day-of-month data, and hence the following functions (Equation 1 and Equation 2) were used to normalize the data (where M_{max} and D_{max} represent the maximum month and day-of-month, respectively).

$$M_n = \frac{2\pi M}{M_{max}} \quad \text{(Eqn 1)}$$

$$D_n = \frac{2\pi D}{D_{max}} \quad \text{(Eqn 2)}$$

M_n and D_n were then used to define the sine and cosine features of the month and day-of-month—M_{sin}, M_{cos}, D_{sin}, D_{cos}—as per Equations 3–6:

$$M_{sin} = sin(M_n) \quad \text{(Eqn 3)}$$

$$M_{cos} = cos(M_n) \quad \text{(Eqn 4)}$$

$$D_{sin} = sin(D_n) \quad \text{(Eqn 5)}$$

$$D_{cos} = cos(D_n) \quad \text{(Eqn 6)}$$

These four features were the only ones kept in the dataset as the time features, so that the only time data fed into the model was cyclical, and the previous linear data forms were removed. This process of cyclical features encoding was performed on both the input and output matrix of the data set with time features.

Machine Learning Algorithms

Before applying the ML algorithms to the data, it is important to realize the final, current structure of the two sets of data. In the input matrix, there are 96 features of past power production data (for the past 48 hours); in the output matrix, there are 48 features of that day's power production data (for 24 hours) (**Figure 2**). The second set of data with time features has an additional four columns for both the input and output matrices, corresponding to the four cyclically encoded time features (M_{sin}, M_{cos}, D_{sin}, D_{cos}).

If we attempt to predict a forecast horizon H of 24 hours using the past 48 hours of past power production data, then Equation 7 and 8 model this relationship, where $\hat{p}[k]$ represents the predicted value of a time series at sample number k, $f(\ )$ represents the machine learning function that takes in the data inputs, and H_1 represents a 24-hour horizon (3):

$$\hat{P}[k] = f(\ P[k-H_1],\ P[k-2H_1]\ ) \quad \text{(Eqn 7)}$$

$$\hat{P}[k] = f(\ P[k-H_1],\ P[k-2H_1],\ M_{sin}[k],\ M_{cos}[k],\ D_{sin}[k],\ D_{cos}[k]\ ) \quad \text{(Eqn 8)}$$

Before being fed into the ML models, the data was split into a training and test set with a 90:10 ratio respectively. This means that the first 90% of customers in the dataset, or 270 customers were used to train the model, and the remaining 10% were used for testing.

The ML techniques tested in this study are a mix of simple and complex ML models. Five simple, traditional models and seven neural networks are tested. The five traditional models include Linear Regression (LR), K-Nearest Neighbors (KNN), Decision Tree Regressor (DT), Multi-Layer Perceptron Regressor (MLP) and Random Forest (RF). The default parameters of each model from the scikit-learn library were used in order to simplify the task. The first model, LR, will act as the baseline model, to measure how increasingly complex models perform against it. The rectified linear unit (ReLU) (maximum) activation function is used for all the models, so that the models can learn non-linear relationships in a fast, efficient way. All the models were compiled with a MSE loss function (**Table 1**). Meanwhile, Ordiano *et al.* used two neural networks as well, four polynomial techniques, and a persistence method. Compared to ours, Ordiano *et al.* used much more simple and less diverse machine learning techniques.

Evaluation Metrics

The predictions of the models on test data were evaluated using the same metrics as the study by Ordiano *et al.*, so that results can be compared directly between both studies (3). Three key metrics were measured: MAE, RMSE, and PCC, between the actual value and the predicted value ($r_{P\hat{P}}$). The error in forecasting, $e_f[k]$, was calculated by the following Equation 9 (as per the study by Ordiano *et al.*), where P represents the actual value and $\hat{P}$ represents the predicted value (3):

$$e_f[k] = \hat{P}[k] - P[k] \quad \text{(Eqn 9)}$$

used to calculate the three evaluation metrics, using the equations below from the study by Ordiano *et al.*, where $\bar{P}$ and $\bar{\hat{P}}$ represent the averages of each respective time series and K represents the total number of data points (3):

$$MAE = \frac{1}{K}\sum_{k=1}^{K}|e_f[k]| \quad \text{(Eqn 10)}$$

$$RMSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(e_f[k])^2} \quad \text{(Eqn 11)}$$

$$r_{P\hat{P}} = \frac{\sum_{k=1}^{K}(P[k]-\bar{P})(\hat{P}[k]-\bar{\hat{P}})}{\sqrt{\sum_{k=1}^{K}(P[k]-\bar{P})^2}\ \sqrt{\sum_{k=1}^{K}(\hat{P}[k]-\bar{\hat{P}})^2}} \quad \text{(Eqn 12)}$$

All three metrics were used to evaluate each model. The MAE, RMSE, and PCC were all compared directly against the

same values of the models in the paper by Ordiano *et al.* (3). As the same dataset has been used, the MAE and RMSE are on the same scale, and thus direct comparisons were made between the values. The PCC value was used to evaluate the model accuracy by measuring just how closely correlated the actual values are with the predicted values. The PCC value was also used to compare the accuracy of this study with other state-of-the-art weather models.

ACKNOWLEDGEMENTS

We would also like to thank PhD candidate Elena Orlova from the University of Chicago for her help and support in the beginning of this project, and her advice to explore feature engineering!

Received: April 14, 2022
Accepted: August 8, 2022
Published: June 9, 2023

REFERENCES

1. Green, Martin A. "Silicon Photovoltaic Modules: A Brief History of the First 50 Years." *Progress in Photovoltaics: Research and Applications*, vol. 13, no. 5, 18 Apr. 2005, pp. 447–455, doi:10.1002/pip.612. Accessed 09 Sept. 2022.
2. Wirth, Harry. "Recent Facts about Photovoltaics in Germany." *Fraunhofer Institute for Solar Energy Systems ISE,* 2021, www.ise.fraunhofer.de/en/publications/studies/recent-facts-about-pv-in-germany.html. Accessed 09 Sept. 2022.
3. Ordiano, Jorge Ángel González, et al. "Photovoltaic Power Forecasting Using Simple Data-Driven Models without Weather Data." *Computer Science - Research and Development*, vol. 32, no. 1-2, 16 July 2016, pp. 237–246, doi: 10.1007/s00450-016-0316-5.
4. Mosavi, Amir, et al. "State of the Art of Machine Learning Models in Energy Systems, a Systematic Review." *Energies*, vol. 12, no. 7, 4 Apr. 2019, p. 1301, doi: 10.3390/en12071301
5. Waczowicz, Simon, et al. "Demand Response Clustering - How Do Dynamic Prices Affect Household Electricity Consumption?" *IEEE Xplore*, 1 June 2015, ieeexplore.ieee.org/abstract/document/7232493. Accessed 7 Jan. 2022.
6. Almonacid, F., et al. "Calculation of the Energy Provided by a PV Generator. Comparative Study: Conventional Methods vs. Artificial Neural Networks." *Energy*, vol. 36, no. 1, 1 Jan. 2011, pp. 375–384, doi:10.1016/j.energy.2010.10.028.
7. Hanae, Loufi, et al. "Generation of Horizontal Hourly Global Solar Radiation from Exogenous Variables Using an Artificial Neural Network in Fes (Morocco)." *International Journal of Renewable Energy Research-IJRER*, vol. 7, no. 3, 2017, doi:10.20508/ijrer.v7i3.5852.
8. Sharma, Ekanki. "Energy Forecasting Based on Predictive Data Mining Techniques in Smart Energy Grids." *Energy Informatics*, vol. 1, no. S1, Oct. 2018, doi:10.1186/s42162-018-0048-9.
9. Ausgrid. "Solar Home Electricity Data - Ausgrid." *Ausgrid*, 2014, www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data. Accessed 09 Sept. 2022.