Forecasting photovoltaic production with neural networks and weather features

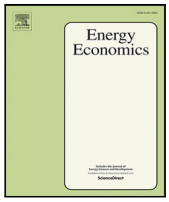
Stéphane Goutte, Klemens Klotzner, Hoang Viet Le, Hans Jörg von Mettenheim

Energy Economics 139 (2024) 107884

Contents lists available at ScienceDirect

Energy Economics

journal homepage: www.elsevier.com/locate/eneeco



Forecasting photovoltaic production with neural networks and weather features

Stéphane Goutte [b,e], Klemens Klotzner [d], Hoang-Viet Le [a,b,*], Hans-Jörg von Mettenheim [c]

[a] Keynum Investments, France
[b] Université Paris Saclay, UMI SOURCE, IRD, UVSQ, France
[c] IPAG Business School, France
[d] European Energy Market Makers, Luxembourg
[e] Paris School of Business, PSB, 59 Rue Nationale, 75013, Paris, France

ARTICLE INFO

Keywords:
Solar energy
Time series forecasting
Machine learning
Neural networks
Entity embedding

ABSTRACT

In this paper, we address the refinement of solar energy forecasting within a 2-day window by integrating weather forecast data and strategically employing entity embedding, with a specific focus on the Multilayer Perceptron (MLP) algorithm. Through the analysis of two years of hourly solar energy production data from 16 power plants in Northern Italy (2020–2021), our research underscores the substantial impact of weather variables on solar energy production. Notably, we explore the augmentation of forecasting models by incorporating entity embedding, with a particular emphasis on embedding techniques for both general weather descriptors and individual power plants. By highlighting the nuanced integration of entity embedding within the MLP algorithm, our study reveals a significant enhancement in forecasting accuracy compared to popular machine learning algorithms like XGBoost and LGBM, showcasing the potential of this approach for more precise solar energy forecasts.

1. Introduction

The transformative impact of machine learning on data analysis and prediction has been pivotal across industries, especially in the domain of energy production. As technological advancements unfold, solar energy has emerged as a significant player in the global photovoltaic market, experiencing impressive growth over the past decade. Italy, ranked sixth in the world for installed photovoltaic capacity in 2020 according to Eni's World Energy Review 2021, has solidified its position as a key player in the solar energy landscape. Solar energy is notably the most widely used renewable source in Italy after hydroelectricity, underscoring its pivotal role in the national energy portfolio.

However, the growth of large-scale photovoltaic (PV) systems in Italy has introduced challenges related to intermittent power production, as highlighted by Fonseca et al. (2012). The fluctuating power production poses operational challenges for grid users and administrators, necessitating frequent adjustments to contend with sudden surpluses or drops in power production. Forecasting the power produced by PV plants becomes crucial for various reasons, including plant performance monitoring, anomaly detection, fault diagnosis, dispatching plans for grid operators, and optimizing operation and maintenance schedules. The inherent influence of weather conditions, particularly solar irradiance and air temperature, on PV systems necessitates accurate models for reliably predicting their performance.

The recent spikes in energy prices and the role of green investment, as discussed in Belaïd et al. (2023), further underline the importance of accurate energy forecasting for enhancing energy security and accelerating the transition towards a sustainable energy future. Moreover, the challenges posed by energy price booms on energy poverty in Europe, as highlighted in Belaïd (2022), emphasize the critical need for effective energy forecasting models to mitigate such impacts.

Despite significant advancements in machine learning techniques, there remains a considerable gap in the application of these models to solar energy forecasting. Specifically, current models often fail to fully integrate weather forecast data and exploit the potential of entity embedding techniques. Our study aims to fill this gap by leveraging entity embedding within the Multilayer Perceptron (MLP) algorithm to enhance the accuracy of solar energy forecasting up to 2 days ahead. By addressing the limitations of existing models and focusing on the specific needs of solar energy forecasting, our research provides a novel contribution to the field.

* Corresponding author at: Université Paris Saclay, UMI SOURCE, IRD, UVSQ, France.
E-mail address: viet.le@keynum.fr (H.-V. Le).

https://doi.org/10.1016/j.eneco.2024.107884
Received 24 July 2023; Received in revised form 25 June 2024; Accepted 28 August 2024
Available online 6 September 2024

Moreover, solar energy forecasting plays a critical role in the broader context of climate change mitigation and renewable energy integration. Accurate forecasts are essential not only for operational efficiency but also for strategic planning and policy development. As renewable energy sources like solar power become more prominent, reliable forecasting models are necessary to ensure grid stability and optimize energy resource management. The following sections will delve deeper into the literature on machine learning applications in energy forecasting, highlighting the unique contributions of our study and situating it within the existing body of research.

Furthermore, accurate and timely prediction of solar energy generation is essential for mitigating the effects of climate change and promoting the integration of renewable energy sources into the power grid. Solar energy forecasting not only supports grid stability and efficiency but also plays a critical role in guiding policy development and strategic planning for renewable energy adoption. This study, therefore, contributes to both the academic understanding and practical application of machine learning in renewable energy forecasting, with significant implications for future research and policy-making.

2. Literature review

The landscape of machine learning applications in solar energy forecasting has undergone significant transformations, addressing the challenges posed by the intermittent nature of renewable energy sources. Early research predominantly relied on traditional machine learning methods, including support vector machines (SVMs) and artificial neural networks (ANNs) (Fonseca et al., 2012; İzgi et al., 2012; Dumitru et al., 2016; Son et al., 2018). However, advancements in the field have led to the adoption of more sophisticated techniques, such as hybrid machine learning, ensemble learning, and deep learning methods (Dou et al., 2023).

Hybrid machine learning methods, integrating various approaches, have shown promise in improving prediction results. Intelligent optimization algorithms have been incorporated into traditional machine learning, with hybrid methods based on support vector machines (SVMs) and extreme learning machines (ELMs) gaining prominence (Olatomiwa et al., 2015; VanDeventer et al., 2019). Genetic algorithms (GA) introduced into SVM by VanDeventer et al. (2019) significantly reduced the root mean square error (RMSE) in short-term PV power prediction compared to conventional SVM models. Similarly, Xiao et al. (2022) proposed an SVM model based on gray wolf optimization (GWO-SVM), leading to a substantial decrease in RMSE in PV power prediction.

Ensemble learning models, such as Random Forest and XGBoost, have also gained prominence by combining multiple machine learning models to enhance accuracy and generalization. The bagging strategy, represented by Random Forest, focuses on reducing variance, while the boosting strategy, exemplified by XGBoost, aims to reduce bias through continuous training of new learners. Additionally, the stacking strategy, leveraging heterogeneous base learners, employs meta-learning to effectively combine them for predictions. These ensemble learning strategies have demonstrated significant improvements in the accuracy of short-term PV power predictions (Mahmud et al., 2021; Ziane et al., 2021; Fan et al., 2022).

The realm of deep learning, a burgeoning avenue in machine learning research, has ushered in novel methodologies for predicting power generation. Past investigations predominantly employed models based on artificial neural networks (ANNs) (İzgi et al., 2012; Dumitru et al., 2016; Son et al., 2018). In contrast to ANNs with independent inputs, recurrent neural networks (RNNs) have demonstrated a superior ability to exploit dependencies within time series data. Pang et al. (2020) showcased that the RNN method significantly enhances the normalized mean bias error (NMBE) and reduces the root mean square error (RMSE) compared to the ANN method for short-term solar radiation prediction. Despite the traditional RNN's effectiveness in leveraging data information, it grapples with issues such as short-term memory limitations and gradient instability.

The effectiveness of LSTM-RNN for long-term prediction was demonstrated by Jung et al. (2019), who employed an RNN model containing LSTM units and analyzed data spanning over 63 months from multiple PV plants. Park and Ahn (2019) introduced LSTM units into a deep RNN for ultrashort-term and short-term prediction, achieving prediction accuracies surpassing 92% in all cases. Mellit et al. (2020) conducted experiments over four distinct short-term time horizons, revealing that LSTM outperformed other models in PV power prediction. Additionally, Carrera et al. (2020) applied deep feedforward networks (DFN) and RNN trained on historical weather forecast data, noting improvements in predicting PV generation 1 day ahead compared to a single machine learning method. Finally, the work of Luo et al. (2021) found a way to enhance the reliability of predictions by imposing constraints on LSTM.

Despite these advancements, there remains a significant gap in the literature regarding the application of entity embedding in energy production forecasting. Entity embedding, popularized in natural language processing (NLP) as word embedding, represents categorical variables in lower-dimensional spaces. Techniques like Word2vec, GloVe, and fastText have successfully applied entity embedding in various fields, including NLP and large language models (LLMs) like BERT, RoBERTa, ChatGPT, and GPT-4 (OpenAI, 2023). However, there is a noticeable absence of literature on the utilization of entity embedding in energy production forecasting. Entity embedding, as a technique adept at handling categorical variables, plays a crucial role in capturing meaningful features of discrete data. Originally employed in natural language processing (NLP) under the term "word embedding", entity embedding represents categorical variables in high-dimensional spaces as continuous vectors in lower-dimensional spaces, aligning each dimension with a meaningful feature of the variable (Guo and Berkhahn, 2016).

In the realm of NLP, this technique has found widespread application, as seen in the success of Word2vec, GloVe, and fastText. Moreover, it constitutes a foundational element in transformer architectures such as BERT, RoBERTa, ChatGPT, and GPT-4 (OpenAI, 2023). Despite its extensive use in various domains, including traffic prediction (Wang et al., 2021) and health machine indicators prediction (HealthMachine), the application of entity embedding in energy production forecasting remains relatively unexplored.

To date, the existing literature lacks comprehensive studies on incorporating entity embedding into the context of renewable energy forecasting. Notable works, such as that of Wagner et al. (2022), have touched upon the usage of embedding for electricity price prediction but did not delve into the realm of energy production. Similarly, the work by Rosato et al. (2016) applied embedding solely to time series features, overlooking the potential of utilizing entity embedding for weather forecast categorical features in the context of energy production forecasting.

This gap in the literature underscores the need for investigations into the potential of entity embedding in the domain of solar energy forecasting. By applying entity embedding to handle categorical variables related to weather forecasts, it may be possible to unlock new avenues for enhancing the accuracy of machine learning models in predicting solar energy production. Our study seeks to fill this void by exploring the impact of entity embedding on solar energy forecasting, leveraging weather forecast data, and drawing insights from its successful applications in diverse domains.

This exploration aims to contribute valuable insights to the broader research on renewable energy integration into the power grid, shedding light on the untapped potential of entity embedding in improving the accuracy of solar energy forecasts. The incorporation of this technique may pave the way for more sophisticated machine learning models capable of handling intricate data structures, particularly categorical variables, and ultimately advancing the field of renewable energy forecasting.

Table 1
Merged weather and energy production data.

	Field name	Type	Description
1	timestamp	datetime	The hour that energy production was recorded.
2	PlantID	integer	The integer that represents each energy plant.
3	Day–Night	string	The daylight status
4	Sky descriptor	integer	The integer that represents each type of sky condition
5	Precipitation descriptor	integer	The integer that represents each type of precipitation
6	Temperature	float	The forecasted temperature in Celsius
7	Wind speed	float	The speed of wind in km/h
8	Wind direction	integer	The direction of the wind by compass degrees (0–359, with 0 equals to North)
9	Humidity	integer	The forecasted humidity level in percentage
10	Energy Production	float	The amount of energy generated by each energy plant during that specific hour in megawatts (MW).

3. Methodology

3.1. Data source

3.1.1. Solar energy production data

The solar energy production data used in this study consists of hourly production measurements from 16 photovoltaic plants located in different regions of Northern Italy. The data span a period of two years from 2020 to 2021 and is provided directly by a solar production company from Italy.

3.1.2. Weather forecast data

The weather forecast data used in this study corresponds to the same regions as the 16 photovoltaic plants. The weather forecast data is obtained from publicly available sources and includes hourly forecasts of temperature, humidity, wind speed, and cloud cover. The weather forecast data is provided in a separate set of files and is also organized by month and region.

Given the geographical proximity of the 16 photovoltaic plants, we consider the possibility of using all the available solar energy production data together as inputs to machine learning models. The inclusion of weather forecast data in the models is expected to improve the accuracy of solar energy forecasting.

3.2. Data preprocessing

3.2.1. Data construction

Data preprocessing is a crucial step in the process of data analysis, which involves cleaning and organizing data in a way that makes it usable for analysis. In this study, the first step of data preprocessing was to gather all the necessary data values from different data sources into a single collection of databases. As there were two different sources of data — one reflecting the data production of each specific plant and the other containing weather forecasts, mapping between the two datasets based on the time stamps was required. Since each day's forecasting needed to be made by 11 am of the previous day, the weather forecasting information nearest before the forecasting took place was used. After the mapping, our unified dataset will be in the form described by Table 1.

3.2.2. Data imputation

The next step is to find and process possible missing values. Sometimes, there can be missing values for some features in the dataset. Some specific machine learning models can ignore or even learn from missing data. However, several of our models such as the neural networks cannot handle missing data. As a result, it is essential to be able to detect and fill these values so that there will not be any errors during the training as well as not losing some information. Several approaches can be used in this step. The two approaches that are more commonly used are to fill the missing values with average, median, zero, or the value of the previous observation or to use an interpolation technique, such as linear, time, quadratic, or cubic interpolation. In this case, the energy production dataset that we received contains three months from May 2021 to August 2021 where there were problems with the production recording procedure of the energy company. Therefore, the production data we received during this period was daily average instead of hourly, which is not suitable for our hourly forecasting model (see Fig. 1).

To deal with this problem, we decided to remove this period from our study and the dataset that followed the incident are used as the test dataset for our study. The period before the incident is used as the training dataset. In short, after data cleaning, we have 178,056 samples for the training set which ranges from January 2020 to April 2021, and 59,376 samples for the test set from August 2021 to January 2022. To ensure that our models are well-tuned and not overfitting the training data, we further divided the training set into a validation set, which contains data from January 2021 to April 2021. This way, the training set can span a full year from January 2020 to December 2020. The validation set will be used during the training phase to evaluate the performance of the models and adjust the hyperparameters accordingly. The final models will be evaluated using the test set, which contains data from August 2021 to January 2022.

3.3. Feature engineering

3.3.1. Cyclical data

As our study focuses on time series analysis, it is crucial to incorporate the time aspect of the data into our features. However, time-related data is often represented in the "datetime" format (YYYY-MM-DD HH:MM:SS), which makes it difficult to extract information beyond the ascending order of data points. This format does not reveal cyclical patterns such as hours of the day, days of the week, months, seasons, etc., which are important for our analysis.

Several approaches have been proposed to address this issue. The simplest approach is to use the number of minutes, hours, months, or weekdays as features. However, this method fails to account for the cyclical nature of the data, such as the difference between 24:00 and 01:00 h. Another approach is to use dummy variables for each hour, but this increases the number of variables and neglects the aspect of consecutive hours.

To address both of these problems, a popular method is to apply sine and cosine transformations to the cyclical data. This method preserves the cyclical nature of the data and accounts for the difference between hours. In our study, we will apply this transformation to the month and hour features as follows:

$$month\_cos = \cos(2\pi \frac{month}{12}) \qquad (1)$$
$$month\_sin = \sin(2\pi \frac{month}{12})$$

$$hour\_cos = \cos(2\pi \frac{hour}{24}) \qquad (2)$$
$$hour\_sin = \sin(2\pi \frac{hour}{24})$$

Here, Eq. (1) was used to add information regarding the months of a year, while Eq. (2) provided knowledge regarding the hours of a day.

Finally, the method of entity embedding which has been mentioned previously is also fully capable of dealing with these problems by representing the relationships among different categorical variables in the latent space.

3.3.2. Categorical data

Categorical data is a common type of data that can be found in many datasets, including the one used in our study. It is a type of

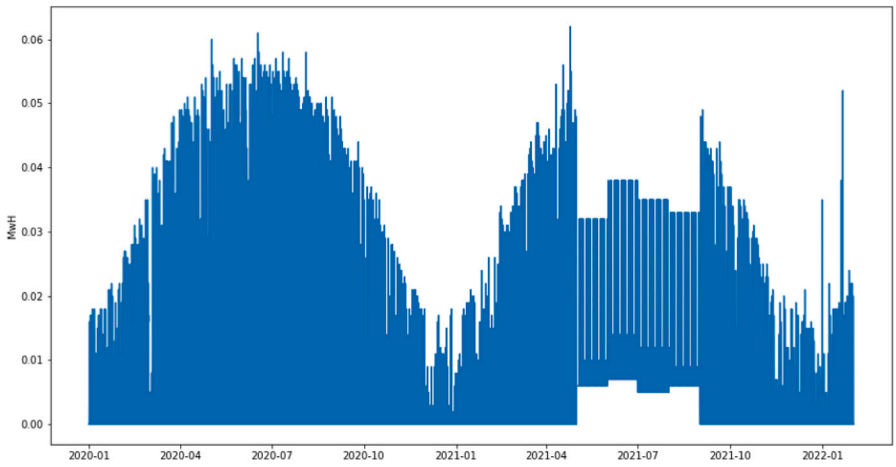


Fig. 1. Example of energy production data from a specific plant.

data that includes variables with discrete values that represent different categories or groups. This type of data can be further categorized into ordinal and nominal data. Ordinal data is data that has an inherent order, while nominal data does not. In our study, all of our categorical data (the plant id, sky descriptors, precipitation descriptors, wind direction, or event hours and months) are nominal and their labels do not imply any order. Machine learning models cannot directly work with categorical data, as they require numerical inputs. Therefore, it is crucial to transform the categorical data into a numerical format that can be used as input for the models.

One of the most common ways to transform categorical data is to use dummy variables (or one hot encoding), where each category is represented by a binary variable indicating whether the observation falls into that category or not. However, this approach has several drawbacks. It leads to a high number of input features, especially when the number of categories is large, which can cause overfitting and slow down the training process. Furthermore, dummy variables do not capture any meaningful relationship or similarity between the categories.

An alternative approach to the conventional use of dummy variables in handling categorical data is to employ entity embedding, an advanced technique widely applied in machine learning. Entity embedding transcends traditional one-hot encoding by mapping each categorical variable to a low-dimensional vector space, introducing a level of semantic richness that is particularly beneficial for capturing nuanced relationships within the data. Mathematically expressed as $f : \text{Category} \rightarrow \mathbb{R}^n$, where n signifies the embedding dimension, entity embedding involves learning embeddings during the training process.

Consider a practical example with weather types where “Sunny”, “Cloudy”, and “Rainy” are represented as entity embeddings denoted as $\mathbf{v}_{\text{Sunny}}$, $\mathbf{v}_{\text{Cloudy}}$, and $\mathbf{v}_{\text{Rainy}}$, respectively. These embeddings could be represented as vectors in $\mathbb{R}^2$, where the values indicate the position of each weather type in the embedding space. For example:

$$\mathbf{v}_{\text{Sunny}} \rightarrow [0.9, 0.1]$$
$$\mathbf{v}_{\text{Cloudy}} \rightarrow [0.3, 0.7]$$
$$\mathbf{v}_{\text{Rainy}} \rightarrow [0.2, 0.8]$$

In this scenario, similar weather types have similar vector representations, enabling the model to recognize and leverage relationships between different weather categories. Beyond language processing, entity embedding finds utility in various domains, including computer vision. The ability of entity embedding to handle both nominal and ordinal data, coupled with its capacity to capture nonlinear relationships between categories, provides distinct advantages over the use of dummy variables. Moreover, it contributes to dimensionality reduction, making it a valuable tool for enhancing the efficiency of machine learning models in tasks ranging from sentiment analysis to language translation. The versatility and effectiveness of entity embedding underscore its significance as a powerful technique for representing categorical data in a more nuanced and informative manner.

Table 2
Conversion of degrees to 8 cardinal directions.

Degree range	Cardinal direction
337–360, 0–22	North
23–67	North East
68–112	East
113–157	South East
158–202	South
203–247	South West
248–292	West
293–336	North West

In our study, we will use both dummy variables and entity embedding to transform the categorical data. We will compare the performance of these two methods to see which one performs better for our specific dataset and model architecture. By using both methods, we can explore the trade-offs between simplicity and expressiveness, and determine the best approach for our particular problem.

Another variable that needs special treatment is the wind direction as it is represented by a compass degree from 0–359. In order to make it interpretable, a conversion from degrees to 8 cardinal directions (north, northeast, east, southeast, south, southwest, west, and northwest) is commonly used. This conversion process involves mapping specific degree ranges to their corresponding cardinal directions. By employing this approach, directional data can be transformed into a categorical format that is suitable for feeding into machine learning algorithms. This conversion enables the model to comprehend and utilize the directional information in a meaningful way, contributing to improved accuracy and interpretability of the predictions generated by the model. Consequently, this degree-to-direction conversion enhances the utility of directional data in machine learning tasks, enabling applications such as weather forecasting, object tracking, and navigation systems to benefit from a more intuitive representation of directional information (see Table 2).

3.4. Model specifications

In this paper, five popular machine learning algorithms were used which is simple Linear regression, two decision tree-based gradient boosting models (XGBoost and LightGBM), and two feed-forward neural network models with one using the entity embedding for both categorical features and the other using dummy variables.

Table 3
MLP no embedding architecture.

#	Layer	Units	Activation function	Dropout
0	Input	–	–	–
1	Dense	70	LeakyReLU	0.4
2	Dense	70	LeakyReLU	0.4
3	Dense	1	Linear	–

Table 4
MLP embedding architecture.

#	Layer	Units	Activation function	Dropout	Connected from
0	Input(float)	–	–	–	–
1	Input(plantID)	–	–	–	–
2	Input(sky desc)	–	–	–	–
3	Input(pre desc)	–	–	–	–
4	Input(direction)	–	–	–	–
5	Embedding(plantID)	2	–	–	Input(plantID)
6	Embedding(sky desc)	2	–	–	Input(sky desc)
7	Embedding(pre desc)	2	–	–	Input(pre desc)
8	Embedding(direction)	2	–	–	Input(direction)
9	Flatten(plantID)	–	–	–	Embedding(plantID)
10	Flatten(sky desc)	–	–	–	Embedding(sky desc)
11	Flatten(pre desc)	–	–	–	Embedding(pre desc)
12	Flatten(direction)	–	–	–	Embedding(direction)
13	Concatenate	–	–	–	Input(float) Flatten(plantID) Flatten(sky desc)
14	Dense	80	LeakyReLU	0.4	Concatenate
15	Dense	80	LeakyReLU	0.4	Dense 8
16	Dense	1	Linear	–	Dense 9

For the Linear regression model, we choose the logistic regression algorithm provided by the Scikit-learn library.

For both XGBoost and LightGBM, we used their official Python libraries. For these two algorithms, their best hyperparameters are tuned using a randomized search on the validation dataset.

Regarding neural network models, in this study, all of our models are built and trained using the Python library Keras running on top of the TensorFlow framework. The architectures of our neural networks which only use the dummy variables are optimized using grid searches on the number of layers, units, and dropout values as follows (see Table 3):

For the neural network models that use embedding, we used up to four embedding layers (depending on if weather features are used or not). If weather features are not used, there is only one embedding layer for plantID whereas additional layers for sky descriptor, precipitation descriptor, and wind direction are applied otherwise. The optimal specification for this network is as follows (see Table 4):

All of the neural network models are trained in 750 training epochs with batch size equaling 1024. Early stopping and model checkpoints based on the validation loss are also used to prevent overfitting so that we can get the best-performing model on the validation set.

3.5. Model evaluation

3.5.1. Root mean squared error

Root Mean Squared Error (RMSE) is one of the most commonly used metrics to evaluate forecasting models. It measures the average deviation between the predicted and actual values, with larger errors being penalized more heavily than smaller ones due to the squaring. The RMSE is calculated by taking the square root of the average of the squared differences between the predicted and actual values. RMSE is sensitive to outliers, which can greatly affect the overall score.

3.5.2. Mean absolute error

Mean Absolute Error (MAE) is another widely used metric that measures the average absolute difference between the predicted and actual values. Unlike RMSE, MAE does not penalize larger errors more heavily, which can make it a better choice when outliers are present. MAE is calculated by taking the average of the absolute differences between the predicted and actual values.

3.5.3. R-squared

R-squared (R2) is a metric that measures the proportion of variance in the dependent variable (i.e., the energy production) that can be explained by the independent variables (i.e., weather and time-related features). R2 ranges from 0 to 1, with higher values indicating that the model is a better fit for the data. R2 is useful for comparing different models, but it does not provide information on the absolute error of the predictions.

3.5.4. Model Confidence Set (MCS)

The MCS, as proposed by Hansen et al. (2005), involves a series of tests aimed at identifying a set of "superior" models, where the null hypothesis of equal predictive ability (EPA) is not rejected at a predetermined confidence level. The EPA test statistic can be computed for any chosen loss function, such as the square (RMSE) or absolute loss function (MAE), as utilized in this study. The MCS procedure is a sequential testing approach that iteratively eliminates the least-performing model at each step until the hypothesis of equal predictive ability is accepted for all models within the superior set. The p-value for each model is used so that models with p-values exceeding the confidence level are included in the identified superior set.

In our study, we employed the MCS procedure on the square and absolute loss associated with the RMSE and MAE, respectively, for each model. This application aimed to ascertain whether the models are superior to others with a confidence level set at 1%. Furthermore, we sought to validate if the results align with the ranking derived from the RMSE and MAE metrics computed in the earlier part of the study. The implementation of the MCS procedure, along with the entire study, was carried out in Python using the MCS module from ARCH library.

3.5.5. Benchmarks

In addition to evaluating our models using the metrics described above, we will also use two benchmarks to assess the performance of our models. The first benchmark is the prediction from an anonymized forecasting company that currently provides services to the energy company that contributed our data. This benchmark will give us an idea of how well our models perform compared to the current forecasting methods in practice. The second benchmark is the persistence model, which is a simple forecasting model that assumes the future values will be the same as the last observed value (in this case the value 24 h ago). The persistence model is often used in energy forecasting because it provides a baseline performance that can be used to evaluate the effectiveness of more complex models.

While all three evaluation metrics (RMSE, MAE, and R-squared) are important in assessing the performance of our models, we will use the root mean squared error (RMSE) as the main metric to select the best-performing model. This is because RMSE puts more weight on larger errors, which is important in energy forecasting where large errors can have significant economic and environmental impacts. We will also use the other two metrics as secondary measures to gain a more complete understanding of the performance of our models.

4. Result and discussion

4.1. General discussion

As discussed in the previous section, the goal of our study is not only to find out the potential application of machine learning in predicting energy production but also to find out whether or not the usage of weather forecasts is necessary for the improvement of the models.

Table 5
Performance of models with weather data.

Model	Weather	RMSE	MAE	R2
MLP with embedding	Yes	**0.00459***	0.00225*	**84.06%**
MLP with dummy	Yes	0.00466	0.00234	83.56%
LightGBM	Yes	0.00499	**0.00224***	81.16%
XGBoost	Yes	0.00509	0.00256	80.40%
Linear Regression	Yes	0.00597	0.00326	73.09%
Benchmark	–	0.00652	0.00322	55.54%
Persistence	–	0.00707	0.00286	62.25%

Table 6
Performance of models without weather data.

Model	Weather	RMSE	MAE	R2
MLP with embedding	No	**0.00538***	**0.00267***	**78.14%**
MLP with dummy	No	0.00540	0.00270	77.98%
XGBoost	No	0.00565	0.00281	75.86%
LightGBM	No	0.00585	0.00270*	74.13%
Linear regression	No	0.00630	0.00329	69.95%
Benchmark	–	0.00652	0.00322	55.54%
Persistence	–	0.00707	0.00286	62.25%

Therefore, we applied a few subsets of features compared to each other to compare their usefulness.

The first dimension for the set of features is the usage of weather forecasting as inputs for the models and the second dimension is about the usage of embedding. However, as the two gradient boosting algorithms are capable of dealing with categorical data on their own, the usage of embedding only applies to the multilayer perceptron models (MLP). For linear regression, only dummy variables are used. In the end, we will have four different MLP models, two models for each of the two gradient boosting models as well as linear regression. In combination with two benchmark models, there are 12 different models in total for comparison. Their results are shown in Table 6 (see Table 5).

Table 6 shows the list of all models with their performance metrics calculated on the test set. The order in which they are listed is based on their RMSE on the test set, from the lowest to the highest, which also means the best to worst. The star next to the value of the RMSE or the MAE indicates if the model belongs to the "superior" set using the MCS procedure based on that specific type of loss function. From just a quick view of the table, we can see that the top models in RMSE are the MLP models. Next comes the two gradient-boosting models. The top 8 performing ones are all combinations of those models in general. Both two versions of the linear regression model come next. However, we can observe that all of our models no matter what specification surpassed the two benchmark models. The persistence model is the worst one and then the predictions from the forecasting company are the second worst. The pattern is quite similar regarding the R2 metrics where the ranking order is almost the same except for the two benchmark predictions. Here, the persistence model has the R2 of 62.25% whereas the benchmark prediction is only 55.54%. Regarding the MAE metrics, the ranking order is a bit different where the performance of linear regression seems to be the worst one and the best one is actually from the LightGBM. However, the MAE of the Embedding MLP is quite close to that of the LightGBM with 0.00225 compared to 0.00224. When analyzing the models using the MCS procedure, we found that the Embedding MLP is superior to all other models regarding RMSE loss. When the MAE is taken into account, the Embedding MLP model is not significantly better than the LightGBM model but both of the models are part of the superior set and are significantly better than the rest.

We can conclude that the best-performing model here in the test set is the MLP model with embedding where it has the best RMSE and R2, the second-best MAE, and is always part of the superior set. Its performance reduces the RMSE by 35.07% compared to the persistence model and 29.6% compared to the benchmark prediction and the MAE by 21.32% and 30.01% respectively. The R2 also increases from the range of 60% or less to up to 84.06%. The performance of this strategy is shown in Fig. 2 where it is compared with the true energy production for the first 1000 data points of the test set.

Table 7
Models performance with and without weather data.

Model	RMSE	MAE	R2
With weather	0.00506*	0.00253*	80.45%
Without weather	0.00572	0.00283	75.21%

4.2. The necessity of weather forecast data

Table 7 presents the average performance metrics of all machine learning models using either the weather features or not. From the table, we can see that the inclusion of the weather forecast data increases more than 5% of R2. It also reduces the RMSE by 11.53% and MAE by 10.6%. Furthermore, the result from the MCS procedure shows that models created with weather data are superior with a confidence level of 1%. We can conclude that weather forecast data is necessary to improve the performance of energy forecasting even if they are more than 24 h old.

This conclusion aligns with the expected notion that weather significantly influences photovoltaic production, even when utilizing weather forecasts from the preceding day for the entire region, as opposed to real-time data from each plant location. To delve deeper into the impact of weather data on each model, we present the feature importance for both the MLP and LightGBM models in Figs. 3 and 4, respectively.

Neural network-based machine learning models are often deemed black-box models due to their intricate internal structures, characterized by numerous layers and parameters. The complexity arises from intricate interactions between these parameters, hindering a clear understanding of their specific contributions to the model's output. Additionally, the use of activation functions and non-linear transformations makes the input–output relationship highly non-linear, further complicating interpretation. Despite these challenges, sensitivity analysis serves as a valuable tool for understanding model behavior by examining how small changes in input variables affect the output. Through the calculation of gradients, which represent the rate of change in output concerning each input, sensitivity analysis quantifies the model's responsiveness to specific features. Fig. 3 showcases the top 10 features most sensitive to the model's output. Notably, Energy Production from the previous day and the cosine transformation of the Hour ("Hour_cos" feature) are significant contributors. While Humidity is not the most critical feature, it holds greater importance compared to other features, and temperature is also among the top 10

Although sensitivity analysis effectively highlights linear relationships, it may not fully capture non-linearities or feature interactions. Fig. 4 depicts the feature importance of the LightGBM model, the second-best model. As a decision tree-based ensemble algorithm, LightGBM derives feature importance from each feature's contribution to reducing mean squared error during training, capturing non-linear relationships and interactions. Interpretation of decision tree-based algorithms like LightGBM is comparatively more straightforward than neural networks. From the Figure, the top three contributing features for the LightGBM model are all weather-related. Humidity is the most crucial weather feature, followed by temperature. Energy production from the same hour of the previous day (Production_last_day) is the most vital non-weather feature, followed by energy production from the nearest hour and PlantID. Despite differences in feature importance between the two models, both highlight the significance of weather forecasting features in their predictions, elucidating the improvement observed when incorporating weather data.

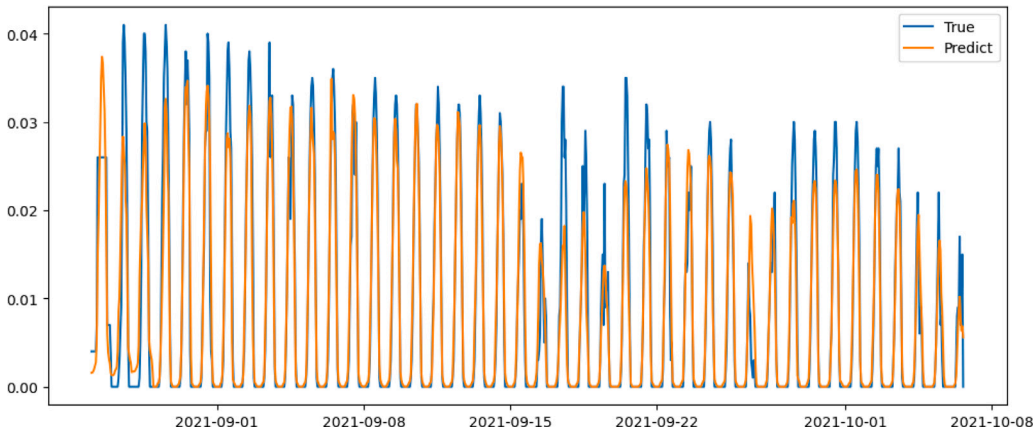


Fig. 2. Model performance in test set.

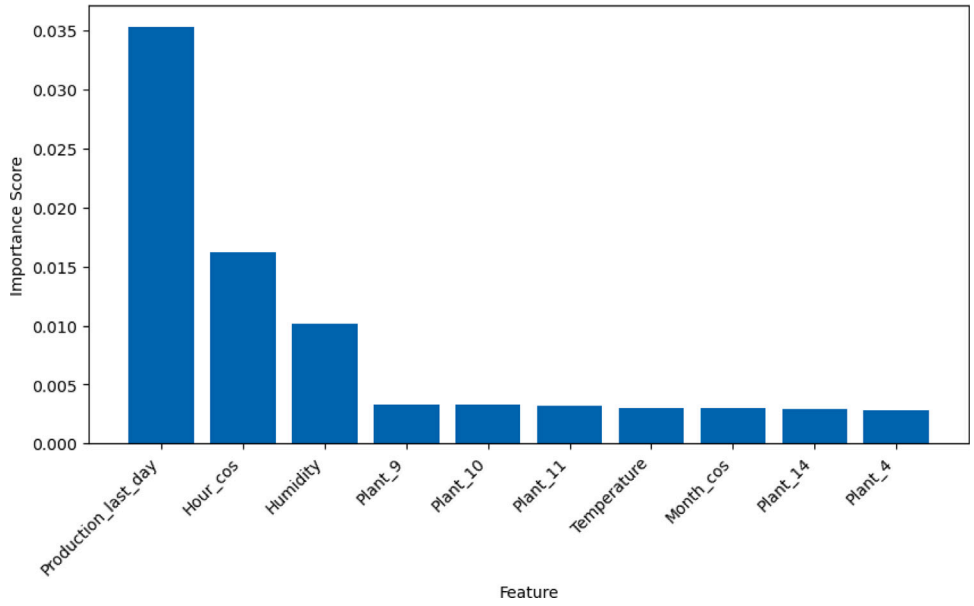


Fig. 3. Top 10 features for MLP model.

Table 8
Average performance of all types of models.

Model	RMSE	MAE	R2
MLP with embedding	**0.00499***	**0.00246***	**81.10%**
MLP with dummy	0.00503	0.00252	80.77%
XGBoost	0.00537	0.00269	78.13%
LightGBM	0.00542	0.00247*	77.64%
Linear regression	0.00614	0.00328	71.52%
Benchmark	0.00652	0.00322	55.54%
Persistence model	0.00707	0.00286	62.25%

4.3. Models comparison

Finally, we would like to compare the general performance of every machine learning model that we used. Table 8 shows the average performance metrics of all models with or without the inclusion of the weather forecasting data.

From the table, it is easy to observe that the neural networks (MLP) are better in all metrics for this specific regression task. They are ranked from the highest to the lowest in the RMSE calculated on the test set. Among them, the MLP model with entity embedding is better in almost every metric compared to other models. The MLP model with dummy variables comes second with slightly worse results. Both of them, however, are a few percent better compared to the gradient boosting algorithms (6.33% in RMSE) except for the LightGBM model with MAE. The performance of the two gradient boosting algorithms (XGBoost and LightGBM) are also quite close too. Still, the performance of those algorithms is closer to the MLP than the Linear Regression and our two benchmarks as well. The result from the MCS procedure also confirms our conclusion from the metrics that the MLP model with embedding is superior to all other models regarding RMSE and it is along with the LightGBM model superior to all other models regarding the MAE.

4.4. Entity embedding interpretation

Regarding the difference between the MLP with and without embedding, we observe a larger difference with the inclusion of weather forecasting data (1.5% reduction in RMSE compared to around 0.3%). This is reasonable considering the inclusion of the weather forecast data also comes with the incorporation of another embedding layer for the sky descriptors data. One of the advantages of using entity embedding is that we can extract the embedding vector for each categorical feature and interpret them to see if they make sense. The embedding vector represents a high-dimensional space where each dimension corresponds to a feature's weight in the embedding. By analyzing the embedding

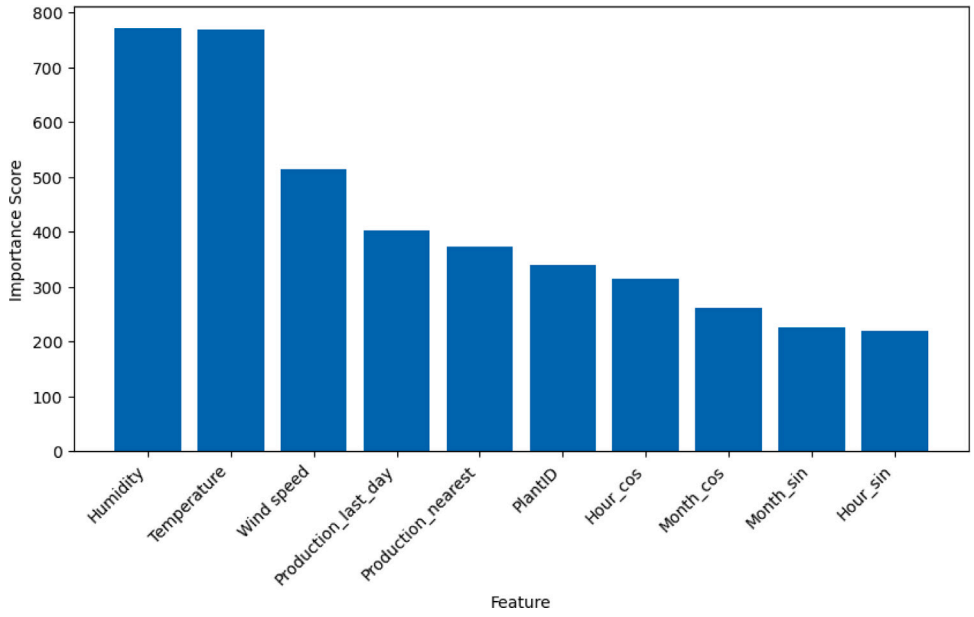


Fig. 4. Top 10 features for LightGBM model.

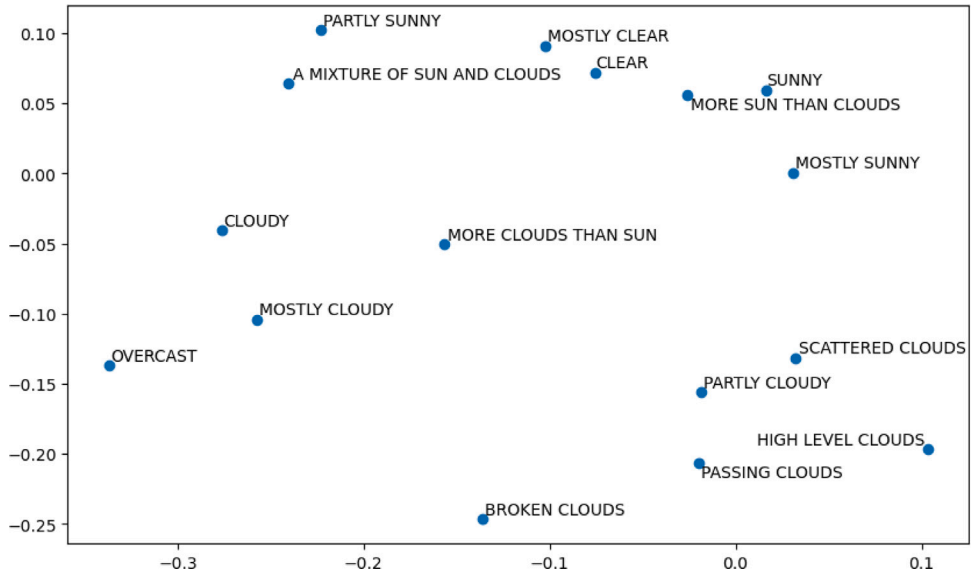


Fig. 5. Sky descriptor embedding.

vectors, we can gain insights into the categorical features and their relationships with the target variable.

Fig. 5 visualizes the relationships between different categories of sky conditions in two dimensions. The value of the 2-d vectors of each sky condition is optimized for solar energy prediction. Of course, we cannot confirm exactly the representation of the two dimensions, however, we can interpret them based on common sense to see if they are reasonable or not. From the figure, we can see that most of the sunny sky conditions stay near each other and are located on the top right of the figure. The sky conditions from the upper part are mostly related to the sunny weather. Regarding the horizontal axis, we can see that on the far left is the overcasting weather which is typically low-level clouds that are thick and often appear gray or white, covering most of the sky. Overcast clouds are often associated with rainy or stormy weather conditions. On the far right, you have high-level clouds which are typically thin and associated with fair weather. From this visualization, we can see that the embedding of those sky descriptors shows rational meaning and can also help us discover new relationships between them and the target variable.

We can also find a similar relationship regarding the embedding of precipitation descriptors. As was shown in Fig. 6, precipitation descriptors such as thunderstorms, heavy rain, and especially thundershowers stay on the upper part whereas sprinkles, a few storms, and a few showers are located on the lower part. We can infer that the embedding representation is likely capturing a gradient or scale of precipitation intensity in the vertical axis as the former group of precipitation involves significant rainfall, thunder, and potentially strong winds. The words grouped together on the upper part represent more severe or intense forms of precipitation, while the words in the opposite direction suggest milder or lighter forms. We can also observe that the precipitations related to snow weather such as snow, moderate snow, and light snow are quite near to each other and located on the right side of the figure. On the other hand, Words like no precipitation, showery, and isolated storms are positioned on the left side, suggesting different conditions or the absence of precipitation. “No precipitation” indicates a lack of

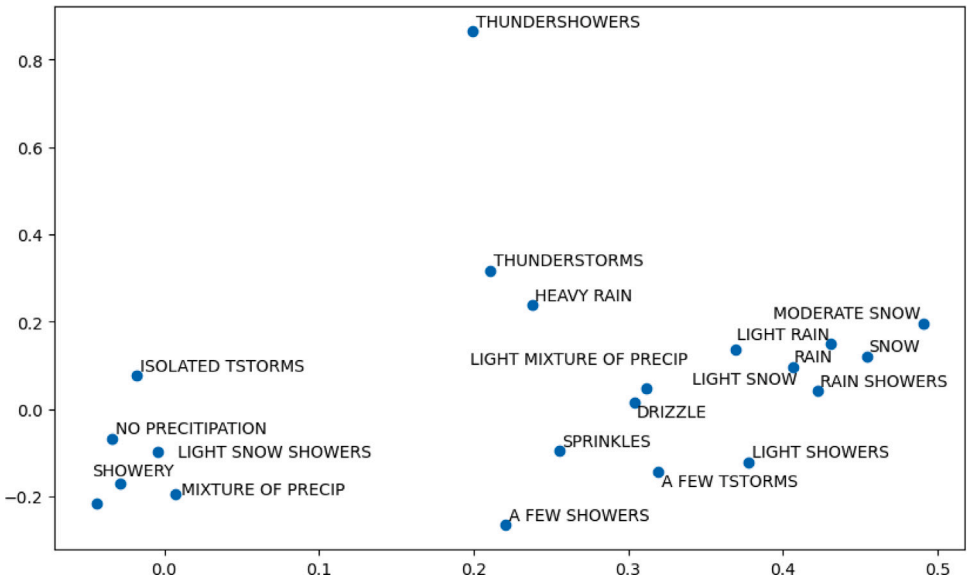


Fig. 6. Precipitation descriptor embedding.

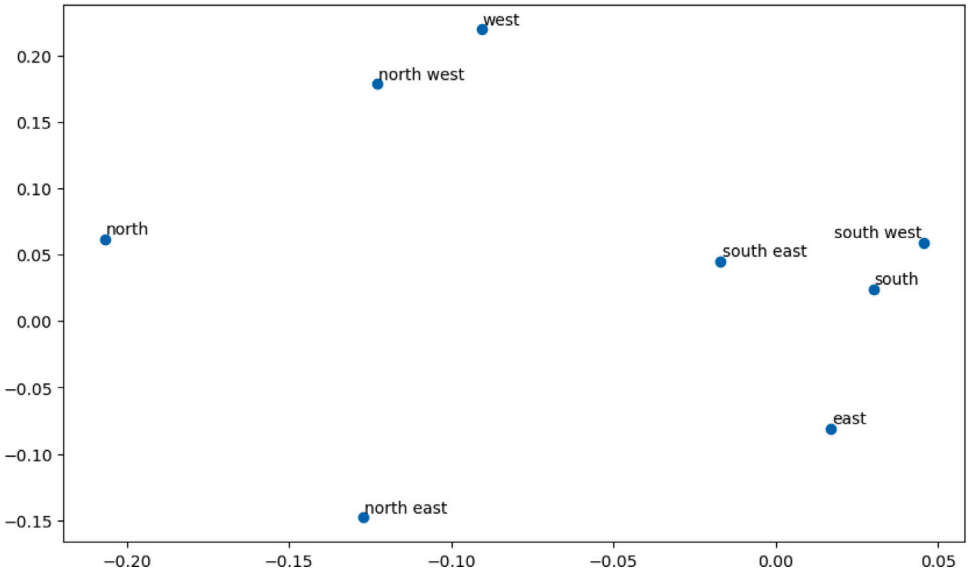


Fig. 7. Wind direction embedding.

rainfall, while "showery" implies intermittent or scattered showers. "Isolated storms" suggests the occurrence of sporadic or localized storm systems. Overall, we can see that the technique of embedding is able to capture not only the intensity but also the type and conditions associated with different forms of precipitation solely based on the energy production itself.

The study of Vasel and Iakovidis (2017) has shown a relationship between the wind direction and the production performance of photovoltaic plants. Using historical data from a solar farm in the UK, their study has concluded that the more the southerly wind occurred, the more power the solar farm in that specific region can produce and that wind direction and wind speed frequencies can be important factors for photovoltaic production forecasting. Based on their research, we also include the wind direction as an input for our models. The process of transforming the direction in degree into 8 main labels of direction is explained in Section 3.3.2. Fig. 7 illustrated the embedding of those directions in a 2-dimensional space. From the Figure, we can observe that the visualization of the directions is quite similar to expectation. The North and the South are in the opposite direction and are far from each other. The northwest and northeast are between the North and the West and the North and the East respectively. Of course, the embedding is not totally perfect as the axis created by the North and the South and the axis created by the West and the East is not quite perpendicular. Furthermore, the distances between the North and the West and the East and the South are quite different. Still, we cannot deny the interesting fact that the neural networks algorithm can "understand" the differences between directions (illustrated by the embedding visualization) solely based on the historical production data and other weather forecasting inputs.

The other embedding about each plantID is shown in Fig. 8. What we observed from the figure is that plants from number 8 to number 14 are quite near to each other. However, as more detail about each plant's characteristics is not revealed to us (their geographic locations, their specification, etc.), the interpretation of the embedding is not obvious in this case. We can only interpret that the vertical axis may represent the production capacity of each plant as their vertical order is the same as the order of the average production of each plant in the training dataset.

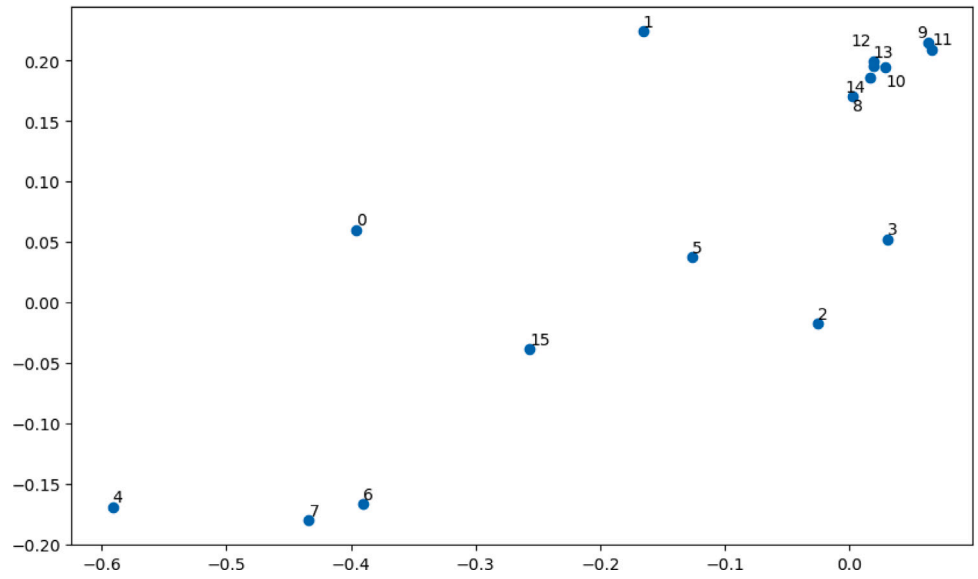
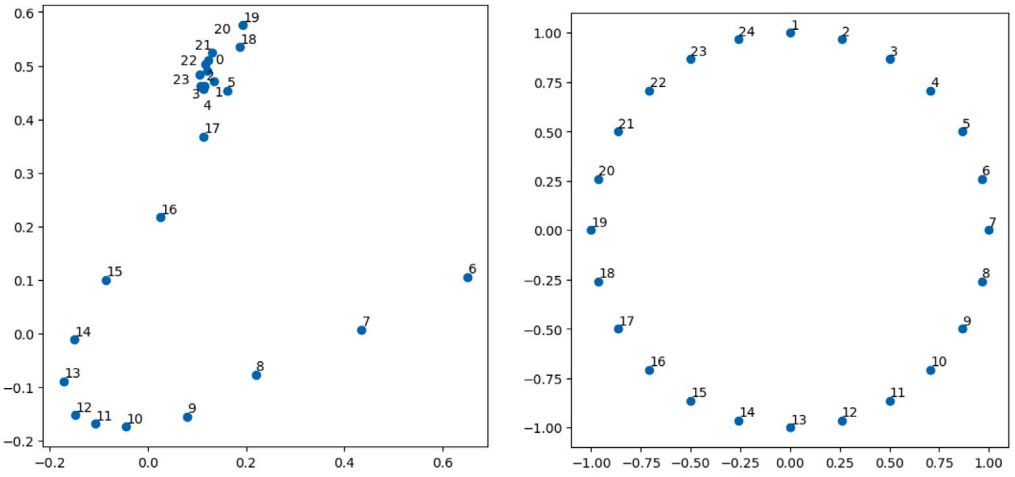


Fig. 8. PlantID embedding.



(a) Embedding **(b)** Sine Cosine Transformation

Fig. 9. Hour transformation.

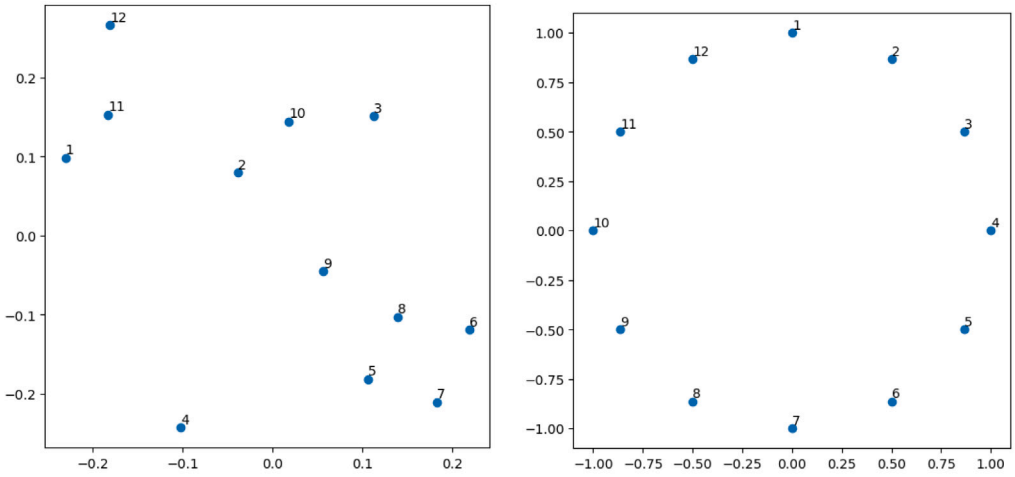
In Section 3.3.1 we did mention that for cyclical data, there can be several transformation approaches to retain their cyclical pattern. One of the popular approaches to handle this kind of feature is the sine and cosine transformation where the input is transformed into a 2-dimensional vector. The illustration of this method for the hour feature can be seen in Fig. 9(b) where the 24 h form into a circle in the 2-dimensional plane and it totally fits with our perception of time in general. However, regarding the topic of photovoltaic production, is this representation of time the most optimal way? The answer lies in Fig. 9(a) where we added the hours as an embedding feature instead of using the sine and cosine transformation. From the Figure, we can still see the cyclical patterns from 6 o'clock to 17 o'clock. However, the hours from 18 o'clock to 5 o'clock of the next day are quite close to each other and there is no specific pattern among them. This is clearly comprehensible as we know photovoltaic plants can only work when there is sun. As a result, the cyclical patterns only appear for the hours that often with the most sun. The neural networks again are capable of capturing this relationship in time features.

The embedding of the month feature, however, did not show a cyclical pattern in a similar fashion as the hour feature as it is shown in Fig. 10. Still, we need to understand that our training dataset only contains one year of production data among a cluster of power plants in the same region. With only one year of data, it can be enough for the models to learn the hourly pattern but probably not for the monthly pattern (June of this year can be unexpectedly hotter than the other June, for example). Nonetheless, we can see from the embedding that the month of November, December, and January (top left) are close to each other and are on the opposite side of the month of May, June, July, and August (bottom right). We can assume that the top left corner of the embedding represents the months of winter whereas the bottom right corner represents the month of summer.

5. Conclusion and future work

5.1. Conclusion

This research sheds light on the advancements in solar energy forecasting by integrating machine learning techniques such as neural networks and gradient-boosting trees. Our findings underscore the augmented accuracy achieved by these methods over conventional approaches, establishing a robust platform for decision-making in energy



(a) Embedding **(b)** Sine Cosine Transformation

Fig. 10. Month transformation.

markets deeply impacted by climate variables. In particular, our study has practical implications for enhancing solar energy production forecasts, crucial for policymakers grappling with the volatilities introduced by climate change.

Neural networks with entity embedding have emerged as frontrunners, offering two-fold benefits: handling categorical weather data with finesse and providing interpretability that aligns with our domain knowledge. With the climate crisis imposing unpredictable alterations in weather patterns, our model's ability to incorporate and analyze such categorical weather data becomes invaluable for policymakers. By focusing on Italy, a country with substantial investment in solar energy infrastructure and unique climatic conditions, we have shown that the methodologies applied here can be adapted to other regions with similar renewable energy landscapes. This enhances the generalizability and applicability of our findings beyond the Italian context, offering valuable insights for countries aiming to integrate more renewable energy into their grids.

Furthermore, our study highlights the broader significance of accurate solar energy forecasting in the context of global energy challenges. Reliable forecasts are essential for strategic planning and policy development. Our research underscores the importance of advanced forecasting techniques in enhancing energy security and stabilizing energy prices, thereby supporting the transition to a sustainable energy future.

Adapting our model can afford policymakers a more granular understanding of the energy potential, enabling optimized allocation of resources and effective grid management subject to climate variability. Accurate and timely predictions enable policymakers to make informed decisions about energy resource management, infrastructure investments, and the integration of renewable energy sources. By providing detailed insights into the performance of solar PV systems under varying weather conditions, our findings support the development of more resilient and efficient energy systems.

These insights are particularly relevant for regions like Italy, where solar energy is a growing facet of the energy mix. As policies evolve to support sustainable energy, the precision of forecasts as facilitated by our model could influence a range of policy decisions—from energy trading strategies to the adaptation of energy infrastructures for resilience against climate-induced disruptions. This study contributes to the limited literature on the application of machine learning in solar energy forecasting, providing a nuanced understanding of the trade-offs involved in energy choices and the optimization of renewable energy resources.

In conclusion, this study offers a novel approach to solar energy forecasting using advanced machine learning techniques. The insights gained from our research are not only applicable to Italy but also provide a framework for other regions facing similar energy challenges. By bridging the gap between theoretical research and practical application, our work contributes to the ongoing efforts to integrate renewable energy sources into the power grid, enhance energy security, and develop effective climate policies. Future research should continue to explore the potential of advanced machine learning models and expand the scope of data integration to further improve the accuracy and reliability of renewable energy forecasts.

5.2. *Future work*

Despite the contributions of our model, the study acknowledges its limitations regarding data granularity. Presently, the weather forecast data reflects regional trends without addressing the micro-climatic conditions encountered by each solar power plant. This discrepancy suggests that policy directives drawn from our findings should be considered with caution, particularly when geographically targeted forecasts are essential for decision-making.

Looking ahead, there is the ground for refinement. Our future research could pivot towards integrating more precise weather forecast data, tailored to the specific locations of power plants. This would undoubtedly bolster the applicability of our model for policymakers and industry stakeholders.

Additionally, the integration of more sophisticated models offers promising avenues for enhancing forecast accuracy. Recent literature has highlighted the potential of improved Recurrent Neural Network (RNN) architectures in energy forecasting. The exploration of model ensembles, combining different machine-learning approaches, may also yield significant improvements. Such methodological enhancements have the potential not only to refine the predictive accuracy of solar energy forecasts but also to broaden the scope for real-time data assimilation and adaptive policymaking in the face of climate change.

CRediT authorship contribution statement

Stéphane Goutte: Writing – review & editing, Validation, Supervision, Methodology, Investigation, Conceptualization. **Klemens Klotzner:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Conceptualization. **Hoang-Viet Le:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Data curation, Conceptualization. **Hans-Jörg von Mettenheim:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT in order to check grammar and improve the readability of the paper. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eneco.2024.107884.

References

Belaïd, F., 2022. Implications of poorly designed climate policy on energy poverty: Global reflections on the current surge in energy prices. Energy Res. Soc. Sci. 92, 102790.

Belaïd, F., Al-Sarihi, A., Al-Mestneer, R., 2023. Balancing climate mitigation and energy security goals amid converging global energy crises: The role of green investments. Renew. Energy 205, 534–542.

Carrera, B., Sim, M.-K., Jung, J.-Y., 2020. PVHybNet: A hybrid framework for predicting photovoltaic power generation using both weather forecast and observation data. IET Renew. Power Gener. 14.

Dou, Y., Tan, S., Xie, D., 2023. Comparison of machine learning and statistical methods in the field of renewable energy power generation forecasting: a mini review. Front. Energy Res. 11.

Dumitru, C., Gligor, A., Enachescu, C., 2016. Solar photovoltaic energy production forecast using neural networks. Proc. Technol. 22, 808–815.

Fan, L., Wang, Y., Fang, X., Jiang, J., 2022. To predict the power generation based on machine learning method. J. Phys. Conf. Ser. 2310 (1), 012084.

Fonseca, J., Oozeki, T., Takashima, T., Koshimizu, G., Uchida, Y., Ogimoto, K., 2012. Use of support vector regression and numerically predicted cloudiness to forecast power output of a photovoltaic power plant in Kitakyushu, Japan. Prog. Photovolt., Res. Appl. 20.

Guo, C., Berkhahn, F., 2016. Entity embeddings of categorical variables.

Hansen, P., Lunde, A., Nason, J., 2005. Model confidence sets for forecasting models.

İzgi, E., Öztopal, A., Yerli, B., Kaymak, M.K., Şahin, A.D., 2012. Short–mid-term solar power prediction by using artificial neural networks. Sol. Energy 86 (2), 725–733.

Jung, Y., Jung, J., Kim, B., Han, S., 2019. Long short-term memory recurrent neural network for modeling temporal patterns in long-term power forecasting for solar PV facilities: Case study of South Korea. J. Clean. Prod. 250, 119476.

Luo, X., Zhang, D., Zhu, X., 2021. Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge. Energy 225, 120240.

Mahmud, K., Azam, S., Karim, A., Zobaed, S., Shanmugam, B., Mathur, D., 2021. Machine learning based PV power generation forecasting in alice springs. IEEE Access PP, 1.

Mellit, A., Pavan, M., Ogliari, E., Leva, S., Lughi, V., 2020. Advanced methods for photovoltaic output power forecasting: A review. Appl. Sci. 10, 487.

Olatomiwa, L., Mekhilef, S., Shamshirband, S., Mohammadi, K., Petković, D., Chintalapati, D.S., 2015. A support vector machine-firefly algorithm-based model for global solar radiation prediction. Sol. Energy.

OpenAI, 2023. GPT-4 technical report.

Pang, Z., Niu, F., O'Neill, Z., 2020. Solar radiation prediction using recurrent neural network and artificial neural network: A case study with comparisons. Renew. Energy 156.

Park, N., Ahn, H., 2019. Multi-layer RNN-based short-term photovoltaic power forecasting using IoT dataset. pp. 1–5.

Rosato, A., Rosa, A., Araneo, R., Panella, M., 2016. Embedding of time series for the prediction in photovoltaic power plants. pp. 1–4.

Son, J., Park, Y., Lee, J., Kim, H., 2018. Sensorless PV power forecasting in grid-connected buildings through deep learning. Sensors 18, 2529.

VanDeventer, W., Jamei, E., Thirunavukkarasu, G., Seyedmahmoudian, M., Tey, K.S., Horan, B., Mekhilef, S., Stojcevski, A., 2019. Short-term PV power forecasting using hybrid GASVM technique. Renew. Energy 140.

Vasel, A., Iakovidis, F., 2017. The effect of wind direction on the performance of solar PV plants. Energy Convers. Manage. 153, 455–461.

Wagner, A., Ramentol, E., Schirra, F., Michaeli, H., 2022. Short- and long-term forecasting of electricity prices using embedding of calendar information in neural networks. J. Commod. Mark. 28, 100246.

Wang, B., Shaaban, K., Kim, I., 2021. Revealing the hidden features in traffic prediction via entity embedding. Pers. Ubiquitous Comput. 25, 21–31.

Xiao, B., Zhu, H., Zhang, S., OuYang, Z., Wang, T., Sarvazizi, S., 2022. Gray-related support vector machine optimization strategy and its implementation in forecasting photovoltaic output power. Int. J. Photoenergy 2022, 1–9.

Ziane, A., NECAIBIA, A., Nordine, S., Dabou, R., Mohammed, M., Bouraiou, A., Khelifi, S., Rouabhia, A., Blal, M., 2021. Photovoltaic output power performance assessment and forecasting: Impact of meteorological variables. Sol. Energy 220, 745–757.