Denoising Forecasts: Leveraging Diffusion Models for Predicting Renewable Energy Supply

Swadesh Jana
Student ID: 6661704
Master Machine Learning

Abstract

Weather-dependent energy output forecasting systems have suffered due to the inherent noise in weather elements. Diffusion models can improve such forecasting by learning these noise patterns. In this work, we show that diffusion models can be used on top of any regression-based model for better forecasting. The quality of forecasting is improved by the denoising diffusion-based conditional generative model over a pre-trained conditional mean estimator. The conditioning prior model and the diffusion model are trained and tested using the RMSE metric on a dataset containing 4 years of hourly weather data with corresponding solar and wind energy supply values. It is observed that the diffusion mechanism consistently outperforms the initial conditioning prior by 4-5% on average. On the solar 1-hour forecast, the diffusion model improves the RMSE from 86.11 to 81.4 while for the corresponding wind data, it improves from 188.57 to 178.35. By carefully tuning the hyperparameters, it is possible to obtain higher improvements as well without much bells-and-whistles. The code is available at `https://github.com/Swadesh13/Renewable-CARD`.

1 Introduction

In the realm of renewable energy supply forecasting, accurate predictions are important for efficient grid management and resource allocation. However, traditional forecasting models often grapple with inherent noise and uncertainty, leading to less reliable predictions due to the variability and intermittency of natural phenomena.

A diverse array of methods for renewable energy supply forecasting have been successfully published as outlined in Section 2. However, most of these methods suffer from being unable to utilize either the underlying noisy variations in the data or the temporal information effectively. A compelling alternative arising recently are diffusion models. Diffusion models have become increasingly popular following their success in generative tasks, particularly image synthesis (Dhariwal and Nichol, 2021; Ramesh et al., 2022). The models effectively manage and propagate uncertainties through stochastic differential equations, capturing the random evolution of variables over time, thus generating high-quality, diverse samples. In this work, the following contributions have been made:

1. Diffusion-based regression models have been used for forecasting the solar and wind energy supplies in Germany. Given the weather data at any timestep, the models predict the energy output in a 1-hour and a 24-hour interval.
2. Due to the probabilistic output of the models, they are evaluated based on the root-mean-square error (RMSE) along with graphs that help to provide a clear discussion of the results.
3. Various neural network architectures along with hyperparameter tuning have been performed to find the most suitable model for the forecasting task.

Seminar Machine Learning for Renewable Energy (Summer Term 2024), Tübingen.

In the following sections, the related works, methodology used for the modeling, the training strategies and the results have been discussed.

2 Related Works

A number of models including statistical time series analysis (Bellinguer et al., 2020), machine learning models, artificial neural networks, particularly feedforward and recurrent neural networks (Hossain and Mahmood, 2020; Lim et al., 2022), and also physical simulation models have been used for solar and wind power forecasting. A comprehensive study has been provided by Tawn and Browell (2022). Diffusion methods have been extensively discussed in Section 3. Although there has not been a comprehensive study on diffusion models in renewable power supply forecasting, Hatanaka et al. (2023) discusses solar forecasting using diffusion models. The implementation of diffusion models for this paper has been derived from the CARD models (Han et al., 2022).

3 Methodology

Forecasting of solar and wind energy outputs at any point of time can be considered a regression task where the inputs are weather conditions on a temporal domain and the output is the energy output or, in this case, a probabilistic distribution of the possible values. In this section, the general diffusion process along with its implementation in a regression problem have been discussed.

3.1 Diffusion process

A forward diffusion process (Sohl-Dickstein et al., 2015) starts at some starting point $y_0 \sim q$, where q is the probability distribution to be learned. Then, it repeatedly adds noise to y_{t-1} by

$$y_t = \sqrt{1-\beta_t}y_{t-1} + \sqrt{\beta_t}\epsilon_t, \tag{1}$$

where $\beta_1, \ldots, \beta_T \in (0,1)$ are fixed constants, and $(\epsilon_1, \ldots, \epsilon_T)$ are IID samples from $\mathcal{N}(0, I)$. Following Han et al. (2022), a prior knowledge of the relation between the original feature vector x and y_0 learned by a neural network (NN), $f_\phi(x)$ is also added. Thus, the forward process can be written as:

$$q(y_t|y_{t-1}, f_\phi(x)) = \mathcal{N}(y_t; \sqrt{\alpha_t}y_{t-1} + (1-\sqrt{\alpha_t})f_\phi(x), \beta_t I), \tag{2}$$

where $\alpha_t = 1 - \beta_t$.

The forward process is a product over multiple timesteps and thus, also a normal distribution: $\mathcal{N}(y_t; \sqrt{\bar{\alpha}_t}y_0 + (1-\sqrt{\bar{\alpha}_t})f_\phi(x), (1-\bar{\alpha}_t)I)$, where $\bar{\alpha}_t = \prod_t \alpha_t$.

From here, we can derive through standard manipulation of Gaussian process, that

$$q(y_{t-1}|y_t, y_0, f_\phi(x)) = \mathcal{N}(y_{t-1}; \tilde{\mu}_t(y_t, y_0, f_\phi(x)), \tilde{\beta}_t I), \tag{3}$$

where

$$\tilde{\mu}_t = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}y_t + \frac{\beta_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}y_0 + \left(1 + \frac{(\sqrt{\bar{\alpha}_t}-1)(\sqrt{\alpha_t}+\sqrt{\bar{\alpha}_{t-1}})}{1-\bar{\alpha}_t}\right)f_\phi(x),$$
$$\tilde{\beta}_t = \frac{1-\tilde{\alpha}_{t-1}}{1-\tilde{\alpha}_t}\beta_t.$$

Using the denoising diffusion probabilistic models (DDPM) method (Ho et al., 2020), the reverse diffusion process is characterized as the joint distribution $p_\theta(y_{0:T})$ with learnable parameters θ, and is a Markov chain with learned Gaussian transitions starting at $p(y_T|x) = \mathcal{N}(y_T; f_\phi(x), I)$. This is shown in Fig. 1. Therefore, the goal is to learn the parameters such that $p_\theta(y_{t-1}|y_t, x)$ is as close to $q(y_{t-1}|y_t, y_0, f_\phi(x))$ as possible. This is possible through variational inference, i.e. performing the negative log likelihood (NLL) and using the Evidence Lower Bound (ELBO) inequality:

$$L(\theta) = \sum_{t=1}^{T} \mathbb{E}_{y_{t-1}, y_t \sim q}\left[-\ln p_\theta(y_{t-1}|y_t, x)\right] + \mathbb{E}_{y_0 \sim q}\left[D_{KL}(q(y_T|y_0, f_\phi(x))||p_\theta(y_T|x))\right] + C, \tag{4}$$

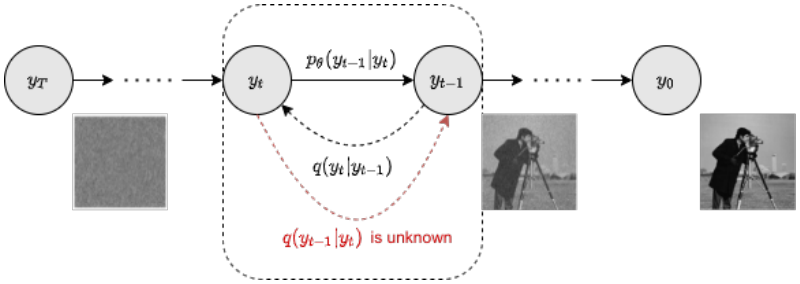


Figure 1: The diffusion process visualization with the example of denoising an image. y_T is the initial noise, p_θ is the trained posterior that tries to learn the noise to be removed from the image at some time step y_t. The forward Gaussian process is represented by the successive $q(y_t|y_{t-1})$.

where $L(\theta)$ is the loss function of the whole process, D_{KL} is the Kullback-Leibler Divergence, and C is a constant term. Since $p_\theta(y_T|x)$ is a normal distribution that does not depend on θ, we can ignore the last two terms and write the loss function as a sum of steps:

$$L(\theta) = \sum_{t=1}^{T} L_t \text{ with } L_t(x, y_t, f_\phi(x), t) = \mathbb{E}_{y_{t-1}, y_t \sim q}\left[-\ln p_\theta(y_{t-1}|y_t, x)\right] \tag{5}$$

3.2 Regression using Diffusion

The general diffusion process can be regarded as a probabilistic regression process to gradually recover the distribution of the noise term, the aleatoric or local uncertainty inherent in the observations (Kendall and Gal, 2017). Here, in addition to the original process, the learned prior f_ϕ is also used. In Eq. 5, the main term is p_θ which has mean μ_θ and variance Σ_θ. Using the method from Ho et al. (2020), Σ_θ is set to $\sigma_t^2 I$, where $\sigma_t^2 = \tilde{\beta}_t$. y_t can be written as $y_t = \sqrt{\bar{\alpha}_t} y_0 + \sqrt{1-\bar{\alpha}_t}\epsilon + (1 - \sqrt{\bar{\alpha}_t}) f_\phi(x)$ from eq. 1 to which $f_\phi(x)$ is also added. A reparameterization of the equations leads to μ_θ being computed as

$$\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(y_t - (1 - \sqrt{\bar{\alpha}_t}) f_\phi(x) - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x, y_t, f_\phi(x), t)\right), \tag{6}$$

where ϵ_θ is a function approximator intended to predict ϵ. Thus, it can be a function of any parameters that are already known at any given timestamp t, as written above. Now, the aim is to optimize ϵ_θ.

The algorithm for the training and inference process have been expressed in 1 and 2 respectively. For training, the model (i.e. the parameters θ) learns the noise at any random timestep $t \in [0, \ldots, T]$, while during inference, the model is repeatedly passed through T steps to get the predicted output, which for a diffusion model will be a distribution.

In Alg. 2, $\hat{y}_0$ is the reparameterized μ_θ value for $t = 0$. It is important to note that the inference algorithm is repeated multiple times (parallelly) to get a mean forecast. This helps to obtain a mean forecast independent of the initial noise. As will be seen further, for certain tasks, taking certain percentiles of the distribution is better than just the mean or median.

Algorithm 1 Training

1: Pre-train $f_\phi(x)$ that predicts $\mathbb{E}(y|x)$
2: **repeat**
3: Draw $y_0 \sim q(y_0|x)$
4: Draw $t \sim$ Uniform$(\{1 \ldots T\})$
5: Draw $\epsilon \sim \mathcal{N}(0, I)$
6: Perform numerical optimization on $\nabla_\theta \left|\left|\epsilon - \epsilon_\theta\left(x, y_t, f_\phi(x), t\right)\right|\right|^2$
7: **until** Convergence

Algorithm 2 Inference

1: $y_T \sim \mathcal{N}(f_\phi(x), I)$
2: **for** $t = T$ to 1 **do**
3: Draw $z \sim \mathcal{N}(0, I)$ if $t > 1$
4: Calculate reparameterized $\hat{y}_0$
5: Let $y_{t-1} = \gamma_0 \hat{y}_0 + \gamma_1 y_t + \gamma_2 f_\phi(x) + \sqrt{\tilde{\beta}_t} z$ if $t > 1$, else set $y_{t-1} = \hat{y}_0$
6: **end for**
7: **return** y_0

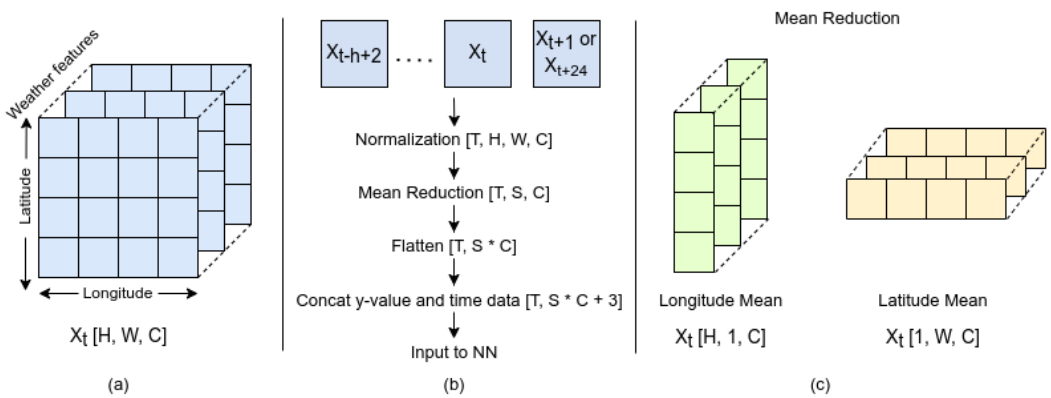


Figure 2: (a) A data matrix at any point in time, (b) T=h time frames are used including the future time step and processed as shown, (c) the two mean reduction methods viz., latitude and longitude used for the study. Note: $[\cdot]$ represents the shape of the data after applying the transformation and S is the resultant shape ($H * 1$ or $1 * W$) after the respective reductions. In general, H=8, W=10, C=8. Additionally, here T denotes the previous time steps of the data and not the diffusion process and t refers to the data point at a particular time t.

3.3 Choosing prior and guidance models

As stated above, there are two parameters that have been introduced, viz. ϕ (or f_ϕ) and θ (or ϵ_θ). The diffusion process can be considered as a secondary step over the predictions of ϕ on input features. Thus, ϕ is a regular neural network (NN) model that solves a regression model. In this paper, the NN model considered is a LSTM block followed by a simple feedforward linear neural network with activations. Theoretically, it can be any regression model.

On the other hand, θ is a guidance model that tries to learn the noise in the diffusion process. Its architecture has been derived from Han et al. (2022). The inputs $x, y_t, f_\phi(x)$ are concatenated and sent through three fully-connected layers. Hadamard product between each of the output vector with the corresponding timestep embedding is computed, followed by a softplus non-linearity. Lastly, we apply a fully-connected layer maps the vector to the output forward diffusion noise prediction.

4 Evaluation

In this section, the dataset, input and output data formats, training strategies and the evaluation metrics have been discussed.

4.1 Dataset

The dataset provided is the German weather dataset for 4 years (2019-2022) with hourly intervals. It is arranged in the form of a 3-dimensional matrix, where two dimensions are the latitudes (8) and longitudes (10), while the third is the feature axis. Alongside it, is the photovoltaic and the wind power output (cumulated over Germany) dataset with 15-minute intervals for the same time period. The 15-minute power outputs are averaged in groups of 4 to get 1-hour intervals. Since all columns are not relevant for wind and solar energy outputs, they are divided according to the power type. For wind energy, the features are geopotential (column name: z), mean sea level pressure (msl), u10, v10, u100, v100, temperature 2m above ground (t2m), and boundary layer height (blh), where u denotes the u-component of the wind speed, v is the v-component, and the numbers are the altitude at which the wind speed is measured. For solar energy, clear-sky direct solar radiation at surface (cdir), total cloud cover (tcc), t2m, surface net solar radiation (ssr), top net solar radiation (tsr), sunshine duration (ssd), forecast surface roughness (fsr) and total precipitation (tp) are used. A few additional pre-processing are also applied as follows:

1. Certain features such as ssr, tsr, fsr, ssd, tp are cumulated over the day. But it will be more useful for the model to get the values over the last hour. Therefore, they are subtracted from the previous hour value to get the density.
2. Past power output data is also made available to the model. This makes it easier to understand the range over which the future output is possible.
3. Feature scaling using mean and square root of variance to get normalized feature values.
4. Add time in the form of hours of the day, and the month of the year after passing them through a sin function to retain the periodicity.
5. The data in the original form is a 4 dimensional matrix, which when reduced to 1 dimensional vector has a dimension of $7 * 8 * 10 * 8 = 4480$. Thus, longitude and latitude mean reduction steps are employed, where for each feature the values are averaged over the longitude or the latitude axes respectively, as shown in Fig. 2(c).

The data processing steps are shown in Fig. 2. There are two forecasts to be made: 1-hour and 24-hour ahead. Let there be a window of h hours in the past that is used for the forecast at time t. Therefore, the input data is $[X_{t-h+1}, \ldots, X_t]$ where X denotes the respective features at that timestamp, to predict either the y_{t+1} or y_{t+24} energy outputs.

4.2 Training Strategy

The models are trained on four different datasets: solar with 1-hour (S-1) and 24-hour (S-24) forecast and wind with the same forecast (W-1 & W-24). The deterministic NN is first pre-trained on the datasets for 1000 epochs and then the guidance probabilistic NN is trained for 1000 epochs, always with a batch size of 256. The optimizer is Adam with initial learning rate of 10^{-3} and further stepped down by a factor of 0.1 every 400 epochs. It is observed that gradient clipping of 1 (L2 norm) helps to stabilize the training process. The loss functions used are L1 for wind prior conditioning, L2 for all other training processes.

After some hyperparameter tuning, it is observed that the best window sizes are 7 for S-1, 13 for S-24, 5 for W-1 and 7 for W-24. It is also observed that appending the original data to the prior conditioning prediction helps in the wind forecast but does not help for the solar forecasts. As shown later, the solar energy outputs have a simple distribution over time and thus, might not need a lot of data for accurate predictions.

The Root Mean Square Error (RMSE) evaluation metric has been computed on the 2022 test data both on the predicted distribution of the data and the mean of the distribution. It measures the square root of the L2 loss between the forecasted energy value and the true value. Mathematically, over N samples, it can be written as:

$$RMSE(y, \hat{y}) = \left(\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \right)^{\frac{1}{2}}, \tag{7}$$

where $\hat{y}_i$ is the i-th forecasted value and y_i is the corresponding true value.

5 Results & Discussion

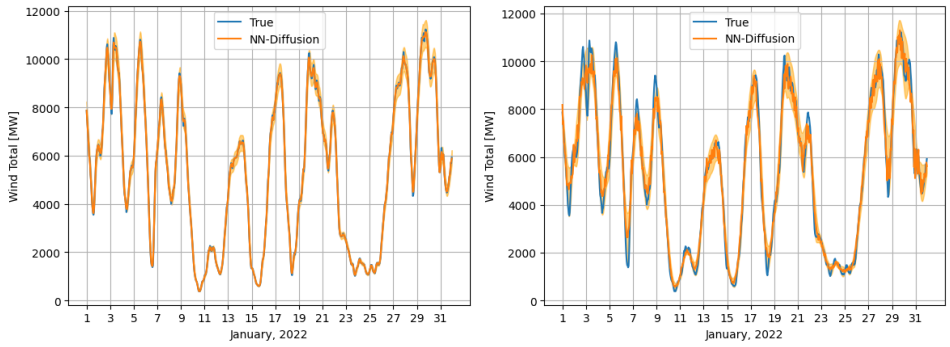
The results for each data type are provided in Table 1. Diffusion process consistently improves on the RMSE scores obtained by the original conditioning LSTM-based NN model. Longitude reduction of the data aids the solar energy output forecasting while latitude reduction can be found to be better for wind energy forecasting.

For wind energy forecasting, the model is able to accurately predict the energy output patterns for the 1-hour forecast as shown in Fig. 3a. The accuracy decreases for the 24-hour forecast (Fig. 3b), but is still able to predict the average pattern except the sharp fluctuations that might be due to other external reasons as well. The corresponding values in Table 1 show that the best possible RMSE obtained are 178.35 and 963.43 respectively for the W-1 and W-24 forecasts.

The solar energy output forecasting, on the other hand, requires some more analysis before making final conclusions. As seen in Fig. 4a and 4b, the true solar values form a periodic Gaussian-like

Model	Solar-1	Solar-24	Wind-1	Wind-24
NN-Lat	86.11	481.22	190.74	986.11
NN-Lat-Diff	**81.4** ! (5.8%)	**333.72** ! (30.7%)	180.95 (5.1%)	931.21 (2.3%)
NN-Lon	95.28	505.31	188.57	953.06
NN-Lon-Diff	90.75 ! (4.8%)	390.21 ! (23%)	**178.35** (5.3%)	**963.43** (2.3%)

Table 1: Results for each dataset type. Diff means diffusion aided model. Lat means longitude reduction and thus latitude information is still kept, while Lon is vice versa. (%) refers to the improvement over the NN model. Note: ! - means the 70-th (S-1) and the 100-th (S-24) percentiles are taken instead of the mean over the predicted forecasts.



(a) (b)

Figure 3: (a) W-1 forecasts for January 2022 after the diffusion process. (b) W-24 forecasts

shape, that is much easier to predict. However, the forecasts seem to mostly miss the peaks. Fig. 4c shows how the overall diffusion forecast distribution is able to cover the maxima for each day. Thus, by taking a higher percentile of all the predictions at each point of time, it is possible to increase the scores. As seen in Table 1, the RMSE for S-1 and S-24 are 81.4 (70-th percentile) and 333.72 (100-th percentile) respectively. For the 24-hour forecasts, as observed in Fig. 4d, even the 100-th percentile (the highest point of the distribution), is well below the true energy output peaks. The difference is found to be maximum for the higher peaks. This can be attributed to the fact that the model has no information on the last 24 hours energy outputs.

6 Conclusion

The diffusion process enhances predictions over the initial conditioning model. However, this method can be computationally intensive due to the numerous timesteps required, and it necessitates the optimization of additional hyperparameters, alongside nuisance parameters such as deciding which percentile to use and whether to include the original input data. The degree of improvement from diffusion models varies depending on the complexity or difficulty of the task, suggesting that these models may not always offer consistent benefits.

Another consideration is the role of probabilistic forecasts, which, while not always essential, can provide a valuable range of possible outputs. An open question remains whether an optimal prior conditioning model exists that could negate the need for diffusion, raising concerns about the potential limit of improvement achievable through this method. Moreover, the importance of noise, as observed in multiple distributions per sample, suggests that fine-tuning the initial noise could enhance predictions or reduce the computational burden of generating multiple predictions per sample. These aspects highlight the need for further research to fully exploit the benefits of diffusion models.

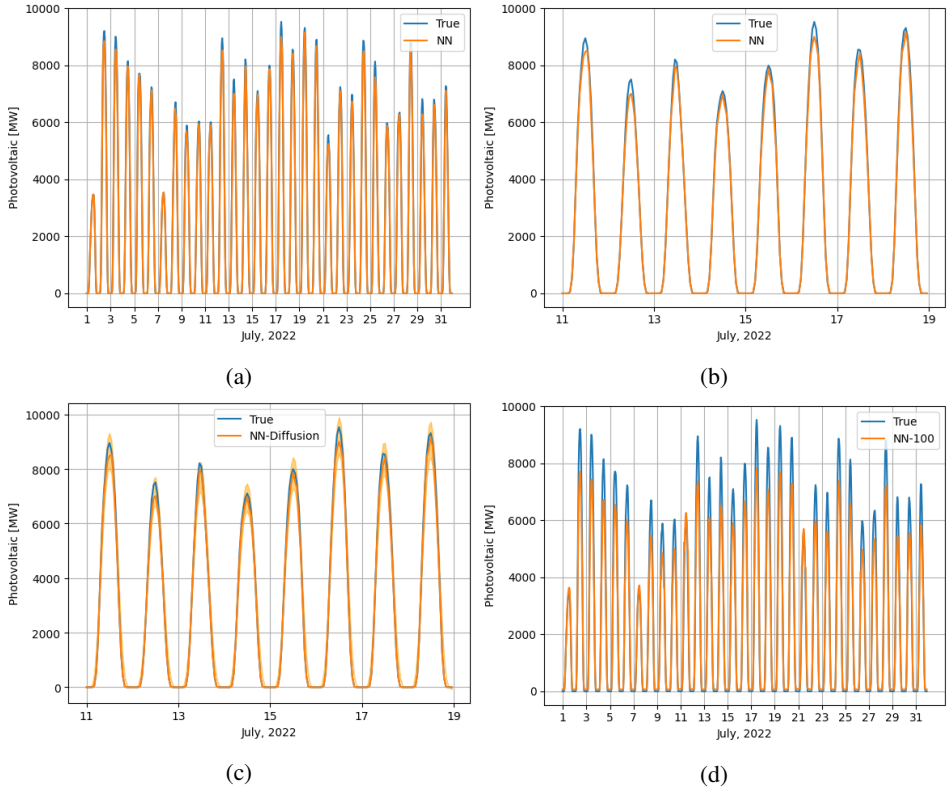


Figure 4: (a) S-1 forecasts for July 2022 by the prior conditioning model. (b) A lateral zoomed-in version of (a). (c) The forecasts after the diffusion process. Since the inference covers 100 different initial noise, they form a wide forecast distribution. (d) S-24 forecasts for July 2022.

References

Bellinguer, K., R. Girard, G. Bontron, and G. Kariniotakis (2020). "Short-term Forecasting of Photovoltaic Generation based on Conditioned Learning of Geopotential Fields". In: *2020 55th International Universities Power Engineering Conference (UPEC)*, pp. 1–6. DOI: 10.1109/UPEC49904.2020.9209858.

Dhariwal, P. and A. Nichol (2021). "Diffusion models beat gans on image synthesis". In: *Advances in neural information processing systems* 34, pp. 8780–8794.

Han, X., H. Zheng, and M. Zhou (2022). "Card: Classification and regression diffusion models". In: *Advances in Neural Information Processing Systems* 35, pp. 18100–18115.

Hatanaka, Y., Y. Glaser, G. Galgon, G. Torri, and P. Sadowski (2023). "Diffusion models for high-resolution solar forecasts". In: *arXiv preprint arXiv:2302.00170*.

Ho, J., A. Jain, and P. Abbeel (2020). "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33, pp. 6840–6851.

Hossain, M. S. and H. Mahmood (2020). "Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast". In: *Ieee Access* 8, pp. 172524–172533.

Kendall, A. and Y. Gal (2017). "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems* 30.

Lim, S.-C., J.-H. Huh, S.-H. Hong, C.-Y. Park, and J.-C. Kim (2022). "Solar power forecasting using CNN-LSTM hybrid model". In: *Energies* 15.21, p. 8233.

Ramesh, A., P. Dhariwal, A. Nichol, C. Chu, and M. Chen (2022). "Hierarchical text-conditional image generation with clip latents". In: *arXiv preprint arXiv:2204.06125* 1.2, p. 3.

Sohl-Dickstein, J., E. Weiss, N. Maheswaranathan, and S. Ganguli (2015). "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International conference on machine learning*. PMLR, pp. 2256–2265.

Tawn, R. and J. Browell (2022). "A review of very short-term wind and solar power forecasting". In: *Renewable and Sustainable Energy Reviews* 153, p. 111758.