19. Symposium Energieinnovation, 11.–13.02.2026, Graz/Österreich

# Residential Photovoltaic Generation Forecast via Long Short-Term Memory and Transformer

Marcel Lüdecke[1]* , Elias Oppermann[1], Michel Meinert[1], Bernd Engel[1]

[1]*Technische Universität Braunschweig, elenia Institute for High Voltage Technology and Power Systems, Braunschweig, Germany*
[*]*m.luedecke@tu-braunschweig.de*

**Keywords:** Forecasting, Prosumer, Energy Management, Recurrent Neural Network, Forecasting Error

## Abstract

Accurate short-term photovoltaic (PV) power forecasting is increasingly important for maximizing on-site self-consumption and ensuring reliable grid integration as decentralized PV deployment grows. This paper presents a systematic comparison of Long Short-Term Memory (LSTM) and Transformer-based architectures for deterministic short-term PV power forecasting using exclusively publicly available data from multiple climatic regions. The dataset combines multi-year 15-minute PV power measurements from nine plants with corresponding meteorological variables, and it is followed by a unified preprocessing pipeline that includes outlier treatment, interpolation, feature scaling, and correlation-based feature selection. Several feature subsets and input window lengths are evaluated, and Bayesian hyperparameter optimization is employed to refine model configurations for both architectures. The results indicate that using all meteorological variables except cloud coverage with a 2-day input window yields the best performance. Under this configuration, the Transformer model outperforms the LSTM model, achieving on average test errors of MSE = 0.0038 and MAE = 0.0265, compared to 0.0055 and 0.0343 for the LSTM, respectively. An analysis of time-resolved residuals shows that both models exhibit the largest errors around noon, while the Transformer provides a consistently narrower error distribution over the diurnal cycle. These findings highlight the advantages of attention-based sequence modeling for PV applications and offer practical guidance on feature design, input horizon selection, and hyperparameter ranges for future data-driven PV forecasting studies.

## 1 Introduction

### *1.1 Motivation*

The deployment of decentralized photovoltaic (PV) systems continues to progress rapidly. Installed system numbers have risen markedly in recent years, primarily driven by the uptake of small-scale plug-in balcony PV units [1]. This development indicates that PV systems are both economically attractive and widely accepted by the public. However, the ongoing capacity expansion is accompanied by tightening legal and regulatory constraints. For instance, feed-in remuneration is suspended during periods when day-ahead electricity spot prices are negative [2]. Consequently, the efficient on-site use of self-generated PV energy is increasingly important [3]. Moreover, grid operators face increasing challenges in balancing supply and demand as intermittent renewable sources constitute a larger share of the energy mix. Accurate short-term PV forecasts can mitigate voltage fluctuations, reduce curtailment losses, and improve the integration of distributed energy resources into distribution networks [3, 4]. From a prosumer perspective, precise forecasting enables optimized battery storage scheduling, peak-shaving strategies, and enhanced participation in local energy markets [5, 6]. To enable these operational and economic benefits, accurate and reliable short-term forecasts of PV generation are essential.

### *1.2 Contribution*

This work compares two distinct forecasting methods and relies on publicly available datasets. Using separate model training runs, the influence of external variables on forecast accuracy is investigated. The two selected architectures are then compared based on their forecast errors over the day, as this temporal error structure significantly affects the efficient utilization of PV generation.

### *1.3 Paper Structure*

This work is structured as follows: Sec. 2 utilizes related work to discuss the choice of the forecasting method. Sec. 3 as the main part describes the data preprocessing, feature selection as well as the training process. The forecast assessment is done in Sec. 4. A critical evaluation then takes place in Sec. 5, after which the work concludes with a summary in Sec. 6.

## 2 Selection of the Forecasting Method

The choice of forecasting method is driven by the objective of generating high-accuracy, robust short-term PV power forecasts. Against this background, a recurrent Long Short-Term Memory (LSTM) network and a Transformer-based model were selected, as both approaches have been identified in recent studies, e.g., Husein [3], Abdelsattar [7], and Zhou [8], as particularly effective for complex time series forecasting.

Strongly nonlinear relationships characterized the PV time series, with pronounced seasonal and diurnal patterns and high sensitivity to meteorological parameters. These properties require models that can reliably capture both local patterns (e.g., morning ramp-up, cloud passages) and longer-term dependencies (e.g., seasonal changes in irradiance) [3, 6]. Recent reviews on data-driven PV forecasting indicate that classical statistical methods and flat machine learning models (e.g., ARIMA, SVR, random forests) are increasingly being displaced by deep neural networks, which systematically achieve lower forecast errors across different horizons [3, 9].

Multiple recent review articles identify recurrent architectures, particularly LSTM and Gated Recurrent Unit (GRU), as the dominant class in PV forecasting research over the last few years [3, 6]. LSTM networks explicitly address the vanishing-gradient problem and can model long-range temporal dependencies, which are central to capturing irradiance and power profiles under variable weather conditions [3]. Comparative studies investigating various deep learning architectures for solar or PV power forecasting report that LSTM models deliver significantly reduced error metrics (e.g., Root Mean Squraed Error (RMSE) or Mean Absolute Error (MAE)) compared to simple (Multilayer Perceptrons) MLPs and conventional Recurrent Neural Networks (RNNs) in many settings, especially serving as robust baselines for short-term to day-ahead forecasts [3, 7]. GRU models, with a simpler gating mechanism, often provide comparable accuracy, faster training, and lower computational cost, making them practical alternatives depending on application requirements [3]. Both architectures remain widely used and effective for modeling the complex temporal dynamics of PV generation data [3].

Current research demonstrates that LSTM-based models perform reliably even when incorporating meteorological forecast data and can be effectively combined with feature extractors such as convolutional neural networks (CNNs) [3, 6, 10]. Methodologically, LSTM architectures represent an established and extensively evaluated reference approach in PV forecasting literature, serving as a robust benchmark for newer architectures such as Transformers [3, 10]. This integration of LSTM with meteorological data enhances the model's ability to capture environmental influences on PV power, thereby improving forecast accuracy and operational reliability in real-world applications [10, 11].

In parallel to the widespread adoption of LSTM networks, a clear trend toward attention-based architectures, especially Transformers, has emerged over the past two to three years for time series forecasting [12–14]. Transformer models utilize self-attention to capture global dependencies in sequences without explicit recurrence, enabling simultaneous consideration of distant time points and accelerating training through full parallelization of sequence processing [13, 14]. Recent overviews of deep learning methods for PV power forecasting highlight that attention-based and Transformer variants are increasingly regarded as promising extensions of recurrent architectures, particularly for multi-step, medium- to long-term forecasts and for handling high-dimensional input vectors [3, 5, 13].

Recent studies specific to the PV domain demonstrate that Transformer models significantly outperform LSTM and BiLSTM in complex, long-sequence forecasting tasks such as multi-day or seasonal PV power prediction regarding Mean Squared Error (MSE) and MAE metrics [7, 15, 16]. Hybrid approaches combining LSTM and Transformer components highlight their complementary strengths: LSTM layers effectively model local temporal dynamics and short-term patterns, while Transformer components enhance the capture of global dependencies over longer horizons [17, 18]. This supports the view that Transformer-based models currently represent the state of the art in PV time series forecasting, particularly exhibiting strong innovation potential for challenging multi-step or long-horizon scenarios [7, 15, 16]. Empirical evaluations show both approaches achieve high accuracy, and their integration into hybrid architectures offers promising improvements in prediction precision and robustness for PV power generation forecasts [17, 18].

However, comparative studies on time series forecasting emphasize that LSTM models remain highly competitive for short- to medium-term predictions with moderate input window lengths, often matching or surpassing Transformers at significantly lower computational cost [12, 13]. Conversely, comprehensive surveys of Transformers in time series analysis reveal that pure Transformer models excel over recurrent networks on very long sequences and high-dimensional inputs but do not necessarily outperform LSTMs on smaller datasets with strongly localized patterns [12, 13]. This is a crucial factor, as open-access data is not widely available [13]. PV studies also explore alternative deep learning architectures, such as Temporal Convolutional Networks (TCN), Minimal Gated Units (MGU), and CNN-LSTM hybrids, whose performance generally falls within or slightly below that of established LSTM and modern Transformer approaches [5, 6, 8, 19]. This nuanced landscape highlights that choice of model depends on forecast horizon, input complexity, and resource constraints, with LSTMs still serving as robust, efficient baselines and Transformers representing state-of-the-art for complex, long-horizon scenarios [12, 13, 19, 20].

The choice of LSTM and Transformer models enables a systematic comparison between two representative and complementary architecture families reflecting methodological advances in recent years [7]. Comparing these approaches enables a robust evaluation of forecasting performance across diverse horizons and data conditions, providing a scientifically sound basis for further model development [7, 15]. This dual selection enables analysis of trade-offs among prediction accuracy, computational efficiency, and the ability to capture both local temporal dynamics and long-range dependencies in PV power generation data [7, 15].

## 3 Baseload forecasting

The most critical aspect of developing a machine-learning model is selecting and preparing the training dataset. The following details the data sources used in this study and the data preparation process. Subsequently, the selection of features, the

training of the models, and the tuning of their hyperparameters are described.

### *3.1 Data Selection and Preprocessing*

The data used in this research combines photovoltaic power [21–25] and weather datasets [26, 27] from various towns around the world, with a focus on representing different climate zones. The dataset consists of several years in 15-minute resolution for the towns of Gaithersburg (USA) [21], Melbourne [25], Istanbul [24], Hong Kong [22], and Bielefeld (GER) [23].

The photovoltaic power datasets comprises recorded power in kW. To enable accurate comparisons between solar power plants of different sizes, power is normalised to each plant's peak power (kWp).

The weather datasets [26, 27] contain common meteorological features for solar power forecasting, as stated in Tsai [4]: temperature, relative humidity, cloud coverage, shortwave radiation, wind speed, and angle of incidence. The significant features were determined in the subsequent feature selection.

Both the photovoltaic power and weather data used in this study were provided in CSV format. For each town, the corresponding files were merged, and subsequently stored in a SQLite database.

During preprocessing, outliers and missing values must be removed to ensure higher data quality. NAN-values were identified and replaced by zeros. Gaps occurred between two consecutive values because some weather files recorded feature certain values at 30-minute intervals instead of the required 15-minute intervals. Furthermore, gaps with zeros were filled with linear interpolation. In addition to linear interpolation, the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) is employed because it preserves the data's shape and monotonicity, yielding more accurate results [28].

Finally, the data is scaled using a min–max scaler, which normalizes each feature to the [0, 1] range. This normalization enhances the comparability of features whose value ranges differ widely. Furthermore, it helps the model's training process by limiting the range of parameter values to search over [30].

### *3.2 Feature Selection*

The most commonly used features in photovoltaic power forecasting, as employed in Tsai [4], are assessed using scatter plots and Pearson correlation analysis, as described below. For an initial overview, a scatterplot is presented in Figure 1. To assess linear relationships, an initial linear regression is performed for each feature. Figure 2 illustrates the Pearson correlation of weather-related features and the historical PV power. While the Pearson correlation coefficient provides a precise measure of the strength of a linear relationship, features may also exhibit nonlinear relationships with the target variable [30]. Therefore, in a third step, feature combinations are evaluated within distinct machine learning models.

The scatterplot in Figure 1 clearly shows that shortwave radiation exhibits the strongest linear relationship with PV power,

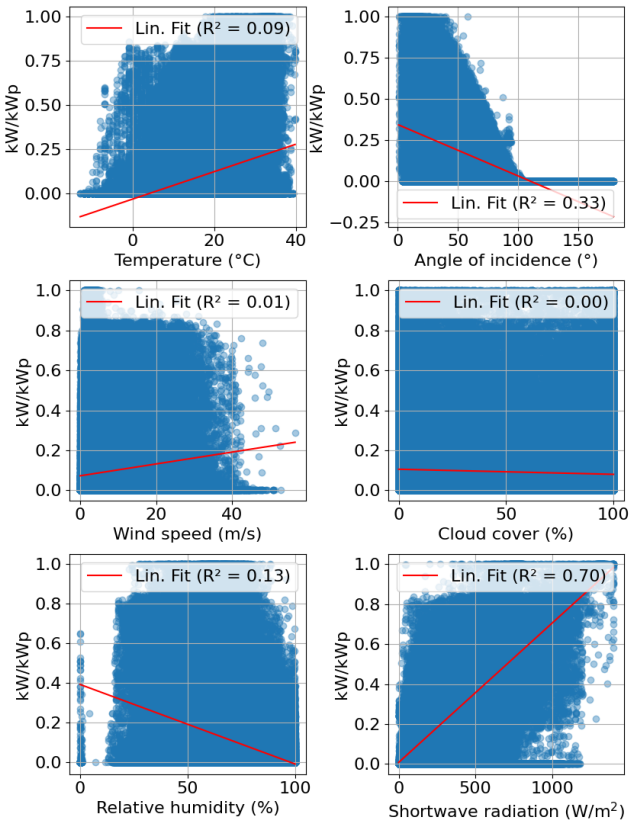


Fig. 1 Scatterplot with linear regression: target variable vs. feature

as also indicated by the highest $R^2$ value. The angle of incidence also plays a crucial role. This angle is determined by the tilt and azimuth of both the sun and the solar module and describes the angle between the direct sunlight and the line perpendicular to the PV panel's surface. Other features, in order of relevance, such as relative humidity, temperature, wind speed, and cloud cover, show a less clear relationship.

The Pearson correlation analysis reveals the strongest correlations between shortwave radiation and generated photovoltaic energy, as well as with the angle of incidence. Although relative humidity exhibits a weaker linear correlation with the target variable, it still significantly influences the generated solar power. The Pearson correlation reveals a minor influence of wind speed and cloud coverage.

As demonstrated in the scatter plot, in Figure 1 and the Pearson correlation, in Figure 2, some dependencies do not appear to be linear. Thus, all mentioned features, regardless of their relevance were used in the following process of model training and hyperparameter tuning. Consequently, the evaluation and assessment in Sec. 4 discuss the influence of the features on the forecasting error.

### *3.3 Model Training and Hyperparameter Tuning*

As described in Sec. 2, this work focused on training a neural network using LSTM and Transformer architectures, implemented in TensorFlow. TensorFlow, along with PyTorch, is one

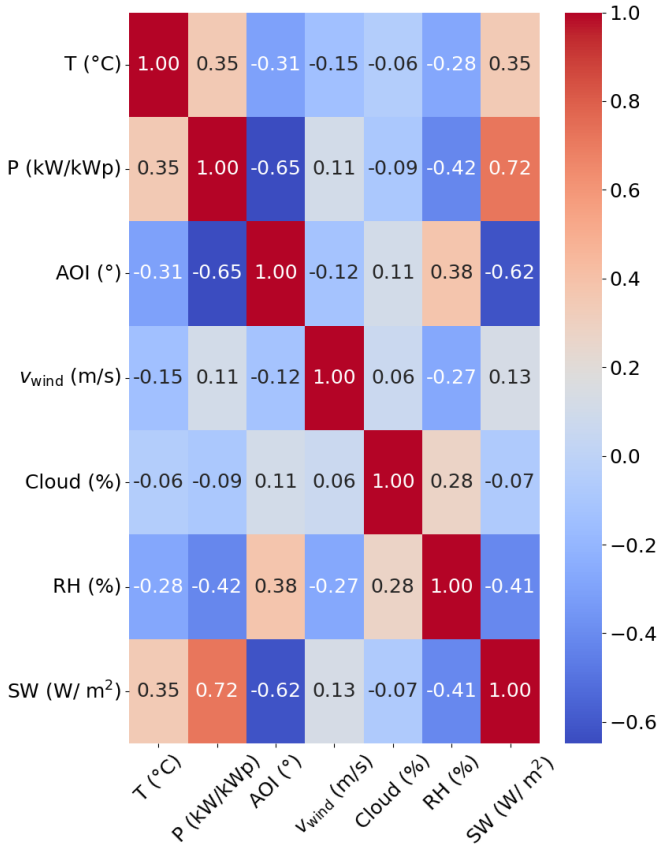


Fig. 2. Pearson correlation of features used for the model

of the most commonly used Python libraries for implementing deep learning models [29]. The dataset was divided into training, validation, and test subsets. The data for each profile was split into 75% training, 10% validation, and 15% test. The training dataset was used to train the model. In contrast, the validation dataset monitored the training process and was used to define potential termination criteria, such as early stopping [29]. The test dataset was reserved for model evaluation, as discussed in Sec. 4, ensuring that the assessment remains independent of the training process. Since the dataset contains samples from nine distinct solar power plants from five different locations, a 9-fold cross-validation scheme was employed, with each fold using data from one plant as the validation set and the remaining eight as the training set. Consequently, nine models were trained and validated. The optimal hyperparameters were determined by the configuration yielding the lowest average validation loss (MSE) across all folds. This approach ensures a robust performance assessment and supports the model's ability to generalize across different locations.

Following the feature selection in Subsec. 3.2, a simplified LSTM model was employed to identify the optimal feature combination and input-window length. In contrast to the complex LSTM model, no hyperparameter tuning and overfitting regularization techniques, such as weight decay and dropout, were applied. This simplified model serves as a baseline for determining the optimal input configuration and for assessing the impact of hyperparameter tuning. Table 1 presents the architecture and hyperparameters used in the simplified LSTM model.

Table 1 Architecture and hyperparameters - simplified LSTM

| Hyperparameter | Interval | Reference |
|---|---|---|
| No. of Layers LSTM | 2 | [31] |
| No. of Neurons first LSTM Layer | 32 | [31, 32] |
| No. of Neurons second LSTM Layer | 64 | [31, 32] |
| AF of LSTM-Layers | (tanh) | [32] |
| AF of Dense Layer | (sigmoid) | [34] |
| Batchsize | 32 | [30, 31] |
| Learning rate | $10^{-3}$ | [31] |
| Optimization function | (Adam) | [30] |

Table 2 Intervals of the optimized hyperparameters - Complex LSTM

| Hyperparameter | Interval | Reference |
|---|---|---|
| No. of Layers LSTM | (1 - 3) | [31] |
| No. of Neurons LSTM | (32 - 128) | [31, 32] |
| AF of LSTM-Layers | (tanh) | [32] |
| AF of Dense Layer | (sigmoid) | [34] |
| Batchsize | (32 - 256) | [30, 31] |
| Learning rate | ($10^{-5}$ - 0.01) | [31] |
| Dropout rate | (0 - 0.5) | [31] |
| Weight Decay | ($10^{-6}$ - $10^{-3}$) | [33] |
| Optimization function | (Adam) | [30] |

The other models were optimized through hyperparameter tuning to reduce the risk of overfitting and underfitting. Bayesian optimization was performed using the Python library Optuna, which is particularly effective for computationally intensive optimization tasks. The mean squared error (MSE) was used as the error metric, as it assigns greater weight to larger prediction errors.

The ranges of variation for the hyperparameters of the LSTM architecture are listed in Table 2. The range was derived from values reported in various reviews of LSTM architectures for solar power forecasting [30–33]. The tanh activation function was used because it has been shown to yield the most promising results for LSTM architectures in solar power forecasting models, according to Liu et al. [32]. To ensure that the output values range from 0 to 1, since the target represents a relative value within this range (unit: kW/kWp), a sigmoid activation function was used in the dense output layer [34]. The Adam optimization algorithm was chosen to estimate the network's weights and biases, as it ensures stable gradient descent and provides an efficient adaptive learning-rate mechanism [30].

The ranges of the hyperparameters for the Transformer architecture are listed in Table 3. Similarly to LSTM, the range of the Transformer hyperparameters was derived from values reported in various reviews [33, 35–38]. The sigmoid activation function was also applied to ensure output values between zero and one [34]. ReLU was used as the activation function in the

Table 3 Intervals of the optimized hyperparameter - Transformer

| Hyperparameter | Interval | Reference |
|---|---|---|
| No. of Encoder Layers | (1,2) | [35] |
| No. of Decoder Layers | (1,2) | [35] |
| Input dimension | (64, 128, 256) | [36] |
| No. of attention heads | (2, 4) | [35] |
| No. of neurons FFN Layer | (128 - 512) | [38] |
| AF of FFN-Layers | (ReLu) | [38] |
| AF of Dense Layer | (sigmoid) | [34] |
| Batchsize | (32, 64, 128) | [36] |
| Learningrate | ($10^{-5}$ - 0.01) | [37] |
| Weight Decay | ($10^{-6}$ - $10^{-4}$) | [33] |
| Dropout rate | (0 - 0.3) | [36] |
| Optimization function | (Adam) | [30] |

Feed Forward Network (FFN) block of the Transformer architecture, as it is the standard choice in the original Transformer design [38].

Each architecture was trained and evaluated for 30 epochs using Bayesian optimization, which efficiently balances exploration of uncertain regions with exploitation of promising areas in the hyperparameter search space to identify the configuration yielding the lowest MSE [39]. During this process, the algorithm progressively focused on the most effective parameter ranges, yielding optimal hyperparameters that are critical for capturing input-feature correlations with the target variable.

Table 4 summarizes the results of the hyperparameter optimization. Conclusively, configurations with up to two LSTM layers yielded the best results. Two layers appear sufficient to capture the complexity of the underlying correlations. However, more critical to overall model performance is the number of neurons in the first two layers of the LSTM architecture. Given that the original optimization range for neurons in the LSTM layers was between 32 and 128 (Table 2), a higher number of neurons in these layers, as indicated in Table 4, produced the best results. In the first LSTM layer, the data enters the network, and the model begins to identify feature correlations. A higher number of neurons in the second LSTM layer is also beneficial. Since the model's capacity is limited by its number of neurons, an architecture with too few neurons can lead to reduced performance [29]. According to the best results, the dropout rate should neither fall below 0.2 nor exceed 0.4. This indicates that a substantial proportion of neurons need to be randomly deactivated to reduce overfitting and improve generalization [31]. A dropout rate above 0.4 (the original upper optimization limit is 0.5) may deactivate too many neurons, increasing model error. A larger batch size was found to be preferable after hyperparameter optimization, as smaller batch sizes tend to result in slower convergence [30]. The optimal learning rate range also narrows, as values below $10^{-3}$ lead to poorer performance. Since a lower learning rate corresponds to smaller steps during weight optimization, the model may converge more slowly or fail to converge properly [34]. Smaller values (below $4 \cdot 10^{-5}$) for weight decay were excluded from the most promising results. A sufficient degree of weight regularization is required to counteract overfitting [33].

Table 4 Optimized Hyperparameter ranges - LSTM

| Hyperparameter | Interval |
|---|---|
| No. of Layers LSTM | (1-2) |
| No. of Neurons - first LSTM Layer | (96 - 120) |
| No. of Neurons - second LSTM Layer | (80 - 120) |
| Dropout - first LSTM layer | (0.2 - 0.4) |
| Dropout - second LSTM layer | (0.2 - 0.4) |
| Batchsize | (96 - 192) |
| Learningrate | ($10^{-3}$ - 0.01) |
| Weight Decay | ($4 \cdot 10^{-5}$ - $10^{-3}$) |

Table 5 shows the optimized hyperparameters for the Transformer models. The encoder, as the core component of the Transformer architecture, initially processes the input data and identifies correlations between features and the target variable, which is crucial to the model's functionality. A single encoder layer is sufficient to capture these correlations. Adding an additional encoder layer increases the model's capacity but may lead to overfitting, as the model begins to memorize training-specific patterns and loses its ability to generalize to unseen data [29]. The optimizations results focused on two decoder layers. A higher decoder capacity is preferable, as it must integrate information from the encoder with feature data for the forecast interval [29]. This part of the architecture, therefore, benefits from increased capacity to process and combine these inputs effectively [29]. The optimization process considered configurations with two or four attention heads, with a primary focus on four. Increasing the number of attention heads enables the model to capture a broader range of correlations and dependencies between different parts of the sequence [38]. In this case, a broader capacity to detect such correlations is required.

In contrast to the LSTM model, the optimization focused on the smallest batch size. A smaller batch size enables finer model calibration and more frequent weight adjustments. This aspect is advantageous in the present Transformer model. The smallest learning rate values were excluded during optimization. Although the overall range of learning rates is relatively small compared with the LSTM model (Table 2), the lowest values can still lead to less effective convergence. The initial learning rate range was intentionally set to relatively

Table 5 Optimized Hyperparameter ranges - Transformer

| Hyperparameter Transformer | Interval |
|---|---|
| Count of Encoder Layers | (1) |
| Count of Decoder Layers | (2) |
| Input Dimension | (64) |
| Count of Attention Heads | (4) |
| Count of Neurons FFN Layer | (128 - 384) |
| Batchsize | (32) |
| Learningrate | ($3 \cdot 10^{-4}$ - $5 \cdot 10^{-4}$) |
| Weight Decay | ($2 \cdot 10^{-5}$ - $5 \cdot 10^{-5}$) |
| Dropout rate | (0.2 - 0.3) |

small values, as higher learning rates can lead to unstable weight updates. Such instability may lead to attention entropy collapse, in which individual attention heads focus almost exclusively on a few positions, thereby neglecting relevant information from other parts of the sequence [37].

The range of weight decay values was restricted relative to the original range because both the smallest and largest values were excluded from the optimization. The optimization focused on a dropout value between 0.2 and 0.3. Since dropout is applied at several points in the Transformer architecture, including the attention heads and the FFN it is essential for promoting good generalization [38].

## 4 Forecast Assessment

Following the description of the model training procedures, the evaluation is conducted in two parts: an analysis of the performance of a simplified LSTM model under different feature-selection configurations in Subsec. 4.2, and a comprehensive assessment of the same simplified LSTM model using the most promising feature–input-window combinations, including an analysis of the resulting error distribution in Subsec. 4.3. Once the optimal combination of features and input window length has been identified, it is applied to both the hyperparameter-tuned LSTM and Transformer models.

### *4.1 Error Metrics*

To ensure a comprehensive assessment, several error metrics are considered. These metrics are presented in equations 1 - 4 [31]. Since the target variable (kW/kWp) is bounded between 0 and 1 and is therefore already normalized, no additional error metrics were considered.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

$$r_i = y_i - \hat{y}_i \tag{4}$$

### *4.2 Feature Relevance and Time Lag*

Several feature selections were evaluated for the simplified LSTM model in this work. As already mentioned in Subsec. 3.2, the following features were considered: temperature, relative humidity, cloud coverage, shortwave radiation, wind speed, and angle of incidence. The error metrics (test dataset) are shown in Table 6, which also lists the omitted features. The input window is set to one day by default.

The differences between the error values are relatively small. However, based on the earlier finding that shortwave radiation and the angle of incidence are the most crucial factors influencing solar power generation, while other meteorological variables, such as cloud coverage and wind speed, play a minor role, incorporating more features still yields a slightly more accurate forecast. This tendency is evident in the comparison of feature combinations 1-3 with 4 and 5.

Table 6 Average test error for different feature selections

| No. | **Feature Selection** | MSE | MAE |
|---|---|---|---|
| 1 | All features | 0.0064 | 0.0383 |
| 2 | All features except cloud coverage | 0.0062 | 0.0378 |
| 3 | All features except cloud coverage and wind speed | 0.0063 | 0.0383 |
| 4 | Shortwave radiation, temperature, angle of incidence | 0.0065 | 0.0388 |
| 5 | Shortwave radiation, angle of incidence | 0.0067 | 0.0393 |

Overall, including all features except cloud coverage yields the most accurate results for solar power forecasting. Since cloud coverage represents a general value for a given location, it does not provide information about the exact position of clouds relative to the sun or how they block sunlight. Consequently, low overall cloud coverage can still result in significant shading of the solar panels, even if only a few clouds obstruct direct radiation [40].

Si et al. [40] demonstrated that accurately capturing changes in cloud coverage requires a CNN–LSTM network to model this phenomenon both temporally and spatially. Furthermore, they proposed an approach that considers the cloud regions blocking sunlight, known as the Active Cloud Region Selection Rule, which is combined with CNN–LSTM outputs in a Sequential Cloud Region Selection algorithm.

In the present work, a simplified representation of cloud coverage was adopted, but it did not improve forecast accuracy. More detailed modeling of cloud coverage may yield greater benefits for forecast accuracy than using a single aggregated value for a given location.

Since a one-day input window was previously used, the number of input days in Table 7 is now adjusted to evaluate its impact on forecast accuracy. The results show that a two-day input window yields the highest forecast accuracy. Nevertheless, the differences among the tested input window lengths are minimal, and increasing the number of input days beyond two does not improve model performance. Using more than two days does not substantially improve forecast accuracy. Therefore, to reduce computational demands and resource usage, a two-day input window is used.

Table 7 Average test error for input-window-length

| Input window (days) | MSE | MAE |
|---|---|---|
| 1 | 0.0062 | 0.0378 |
| 2 | 0.0058 | 0.0365 |
| 3 | 0.0063 | 0.0382 |
| 4 | 0.0059 | 0.0365 |

Table 8 Average test error - simplified LSTM vs. complex LSTM

| Model type | MSE | MAE |
|---|---|---|
| Simplified LSTM | 0.0058 | 0.0363 |
| Complex LSTM | 0.0055 | 0.0343 |

In this work, the most promising results were obtained when all features except cloud coverage were included, with a two-day input window.

### *4.3 Error Distribution*

To evaluate the impact of hyperparameter optimization, the error metrics for both the simplified and complex LSTM models are shown in Table 8. The results indicate that tuning the hyperparameters slightly improves the forecast accuracy. As the observed differences in error are relatively small, it can be inferred that hyperparameter optimization improves model performance to some extent. However, data quality and characteristics likely have a greater impact on overall forecasting accuracy.

Table 9 compares the error metrics of the optimized LSTM (complex LSTM) and Transformer architectures. Both the average MSE and MAE on the test dataset indicate that the Transformer model outperforms the LSTM. This improvement can be explained by the fundamental difference in how the two models handle sequential data: the LSTM processes inputs sequentially, whereas the Transformer captures the entire sequence at once using multi-head attention. This architectural distinction accounts for the Transformer model's higher accuracy [38].

Table 9 Average test error - complex LSTM vs. Transformer

| Model type | MSE | MAE |
|---|---|---|
| LSTM | 0.0055 | 0.0343 |
| Transformer | 0.0038 | 0.0265 |

To visualize how the predicted values deviate from the true values across different folds, hexbin plots for the complex LSTM and Transformer models are shown in Figures 3 and 4, respectively. The hexbin plots illustrate the density distribution of the predicted versus true values. Fold 1 shows the lowest forecast accuracy for both models. In contrast, the remaining folds show smaller deviations from the diagonal, indicating more accurate predictions.

For a detailed performance analysis, the error distribution was examined at 15-minute intervals throughout the day. The residual error was calculated as the difference between the true values and the model predictions (see Subsec. 4.1). Figures

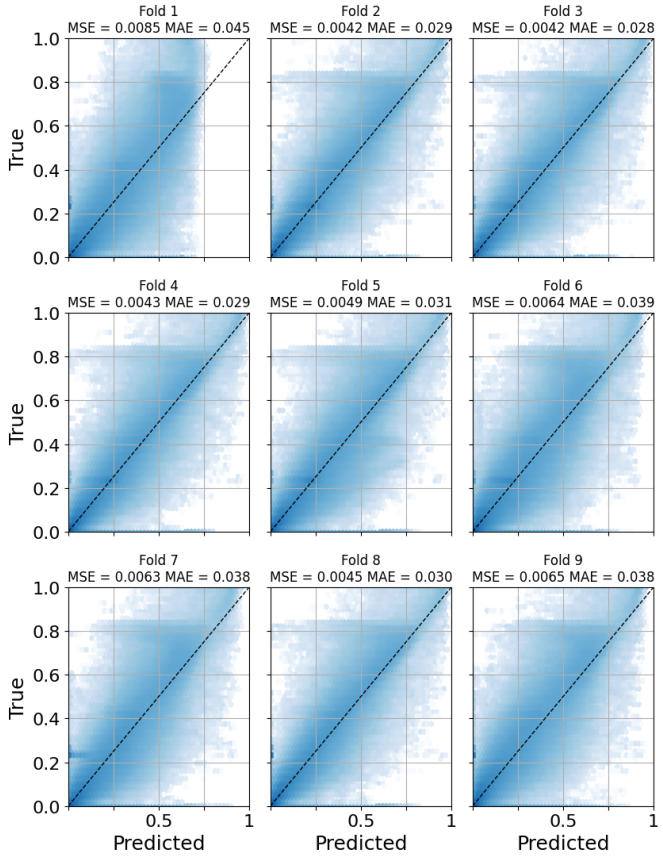


Fig. 3 Hexbin plot: True vs. Predicted Values for the complex LSTM - darker color implies higher density

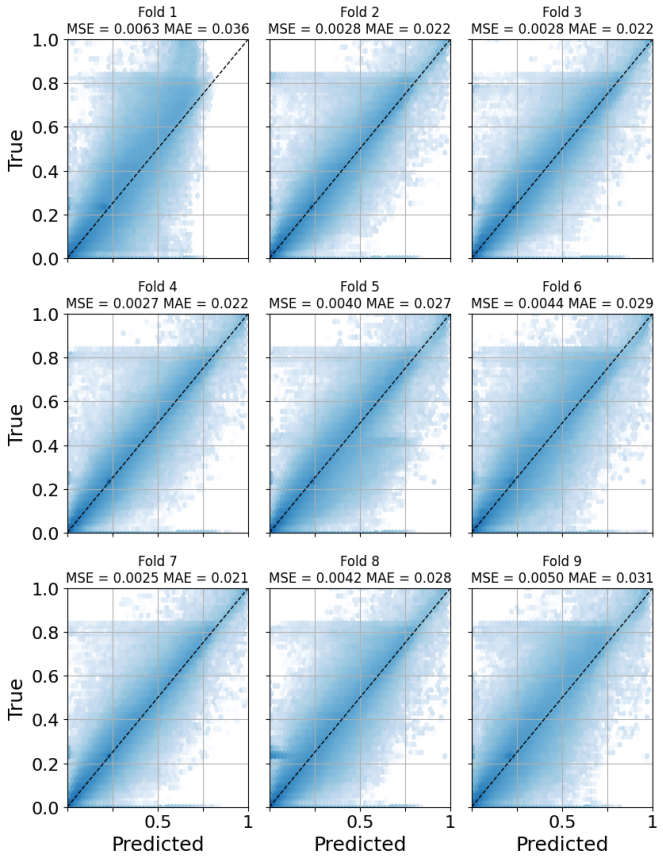


Fig. 4 Hexbin plot: True vs. Predicted Values for the Transformer

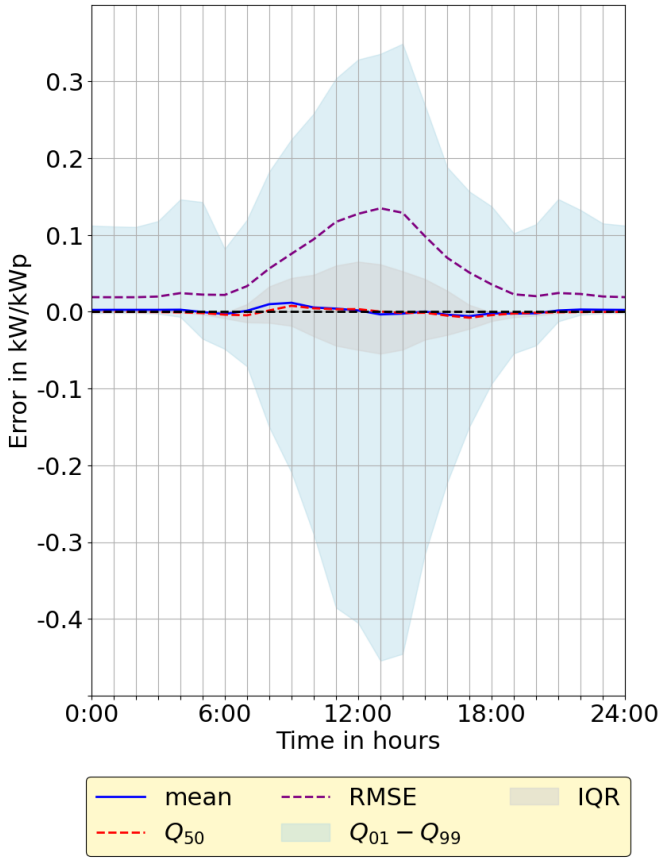


Fig. 5. Error distribution over the day - complex LSTM

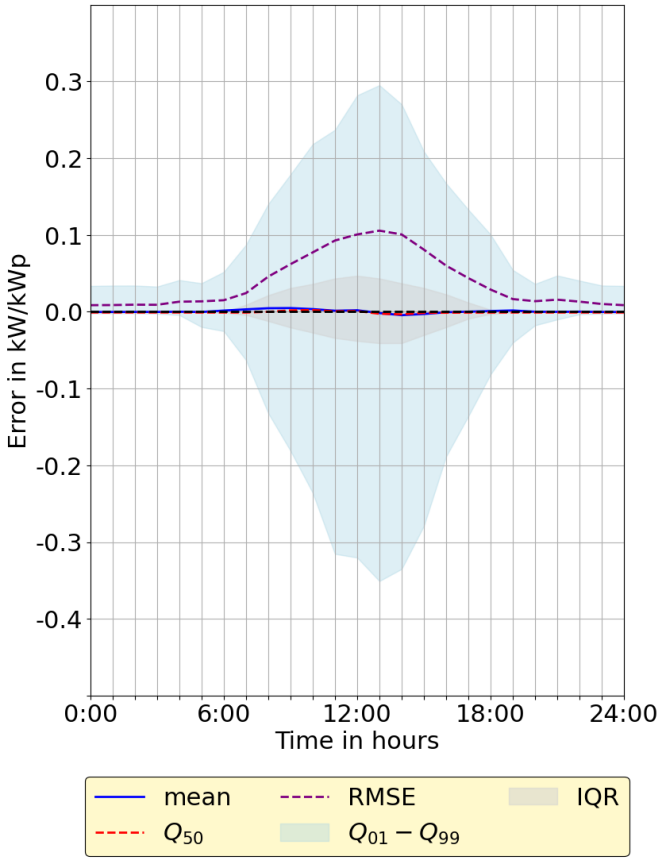


Fig. 6. Error distribution over the day - Transformer

5 and 6 display the resulting curves for the complex LSTM and Transformer architectures, showing both the interquartile range (IQR) and the range between the 1st and 99th percentiles ($Q_{01}$ and $Q_{99}$). The mean line represents the average residual error across all time intervals. In addition to the residual error quantiles, the RMSE is also presented.

As shown in Figure 5, the complex LSTM model exhibits the highest variability in residual error and RMSE around noon, where solar power is predominantly underestimated, with residuals down to roughly -0.4 and occasionally overestimated, with values up to 0.35. The peak RMSE at this time is about 0.13, corresponding to a relative RMSE of 13% under the normalization of the target variable to the interval. In contrast, during night and early morning, the residuals remain close to zero, with only slight overestimations observed. The interquartile range (IQR) between $Q_{25}$ and $Q_{75}$ shows that the majority of forecast errors fall within -0.05 to 0.05, which is approximately 5% of the predicted value.

Figure 6 displays the residual errors and RMSE of the Transformer model in comparison to the LSTM model. The Transformer exhibits a more compact error distribution, with residuals roughly between -0.35 and 0.30. Moreover, the errors during the early morning and late evening hours are smaller, and the IQR indicates a tighter concentration of residuals. The maximum RMSE of 0.11 (11%) confirms that the Transformer model operates within a narrower error range than the LSTM.

## 5 Limitations and Future Work

Different limitations and assumptions in this study should be taken into account. The forecasting model learns correlations from historical data and uses measured meteorological data corresponding to the forecast day. In real-world applications, however, solar power forecasts are usually based on predicted rather than observed meteorological inputs, which introduces additional uncertainty. Consequently, the errors reported in this study are relatively small. They would likely increase when using weather forecast data rather than measurements, consistent with findings [4] that typical RMSE values often lie between 15–30% (RMSE: 6-8% for LSTM and Transformer) in operational settings. Therefore, a performance degradation can be expected when applying the model with realistic weather forecast inputs.

Another limitation is the relatively small number of 30 trials used during hyperparameter tuning, which limits exploration of the hyperparameter space. More trials could identify better configurations and further reduce forecast error [41]. Developing a truly robust solar power forecasting model would require data from many solar power plants across different regions. Since this study is limited to nine plants, its results cannot be directly compared with those obtained from much larger datasets, for example with data from one hundred plants. Future work should therefore expand the hyperparameter search, for

example by increasing the number of trials to 100 or more, to explore a wider range of configurations systematically [41]. New hyperparameters could also be introduced, such as different activation functions and, for the Transformer, additional architectural options including varying numbers of encoder and decoder layers or attention heads.

Because machine learning models strongly depend on the underlying data, improving dataset quality and diversity is essential for better forecast accuracy. Incorporating measurements from more solar plants in different regions would likely enhance the model's robustness and generalization. Regarding cloud coverage, including information about the spatial relationship between clouds and the sun could further improve the predictive value of this feature. Finally, future studies could explore advanced or hybrid deep learning architectures, for example, CNN–LSTM or other combined models, which have been reported to improve performance in related forecasting applications.

## 6 Conclusion

This study examined two general types of machine learning architectures: LSTM and Transformer. The optimal configuration was first identified using a simplified LSTM baseline model, in which different feature sets and input window sizes were evaluated. The best setup used all features except cloud coverage and a two-day input window, which provided a good balance between information and model complexity.

Hyperparameter tuning was employed for both the LSTM and Transformer models, with the search space restricted to the most promising parameter ranges. In addition to architectural hyperparameters, such as the number of encoder and decoder layers in the Transformer and the number of LSTM layers and neurons, regularization-related parameters, including dropout and weight decay, were also optimized within narrow ranges to reduce overfitting.

In direct comparison, the Transformer model outperformed the LSTM model, achieving a test MSE of 0.0038 versus 0.0055 and a test MAE of 0.0265 versus 0.0343. Residual analysis and RMSE analysis showed that both models produced the largest forecast errors around noon and the smallest errors in the early morning, but the Transformer consistently exhibited lower residuals, consistent with its lower MSE and MAE values.

The superior performance of the Transformer architecture can be attributed to its fundamentally different approach to modeling temporal information and dependencies. Through its multi-head attention mechanism, the Transformer can simultaneously capture correlations across multiple time steps, which is advantageous over the strictly sequential, step-by-step processing of the LSTM and has been reported as a key benefit of Transformer-based models in time series forecasting.

## 7 Acknowledgements

This research was funded by "zukunft.niedersachsen", the joint science funding program of the Lower Saxony Ministry of Science and Culture and the Volkswagen Foundation, under project grant number ZN4462. The content of this paper reflects the authors' views and responsibilities solely.

## 8 References

[1] Wirth, H. Recent Facts about Photovoltaics in Germany. , https://www.pv-fakten.de/

[2] Scheer, J. Solarspitzengesetz: How it affects solar and negative pricing. , https://ratedpower.com/blog/how-solarspitzengesetz-affects-solar/

[3] Husein, M., Gago, E., Hasan, B. & Pegalajar, M. Towards energy efficiency: A comprehensive review of deep learning-based photovoltaic power forecasting strategies. *Heliyon*. **10**, e33419 (2024)

[4] Tsai W-C, Tu C-S, Hong C-M, Lin W-M: A Review of State-of-the-Art and Short-Term Forecasting Models for Solar PV Power Generation https://doi.org/10.3390/en16145436 (2023)

[5] Dimitriadis, C., Passalis, N. & Georgiadis, M. A deep learning framework for photovoltaic power forecasting in multiple interconnected countries. *Sustainable Energy Technologies And Assessments*. **77** pp. 104330 (2025)

[6] Al-Dahidi, S., Madhiarasan, M., Al-Ghussain, L., Abubaker, A., Ahmad, A., Alrbai, M., Aghaei, M., Alahmer, H., Alahmer, A., Baraldi, P. & Zio, E. Forecasting Solar Photovoltaic Power Production: A Comprehensive Review and Innovative Data-Driven Modeling Framework. *Energies*. **17**, 4145 (2024)

[7] Abdelsattar, M., Azim, M., AbdelMoety, A. & Emad-Eldeen, A. Comparative analysis of deep learning architectures in solar power prediction. *Scientific Reports*. **15**, 31729 (2025)

[8] Zhou, N., Shang, B., Zhang, J. & Xu, M. Research on prediction method of photovoltaic power generation based on transformer model. *Frontiers In Energy Research*. **12** (2024)

[9] Jadhav, V. & Tiwari, K. A systematic research survey on solar power forecasting using machine learning techniques. *ANNUAL SYMPOSIUM ON APPLIED AND INNOVATION TECHNOLOGICAL ENVIRONMENT 2023 (ASAITE2023): Smart Technology Based On Revolution Industry 4.0 And Society 5.0*. pp. 020001 (2024)

[10] Mbey, C., Yem Souhe, F., Foba Kakeu, V. & Boum, A. A Novel Deep Learning–Based Data Analysis Model for Solar Photovoltaic Power Generation and Electrical Consumption Forecasting in the Smart Power Grid. *Applied Computational Intelligence And Soft Computing*. **2024** (2024)

[11] Ferkous, K., Menakh, S., Guermoui, M., Bellaour, A., Bekkar, B., Rabehi, A., Agajie, T. & Benghanem, M. Optimized solar power forecasting: A multi-decomposition framework using VMD and swarm techniques. *AIP Advances*. **15** (2025)

[12] Ruiru, D., Jouandeau, N. & Odhiambo, D. LSTM versus Transformers: A Practical Comparison of Deep Learning

Models for Trading Financial Instruments. *Proceedings Of The 16th International Joint Conference On Computational Intelligence*. pp. 543-549 (2024)

[13] Kim, J., Kim, H., Kim, H., Lee, D. & Yoon, S. A Comprehensive Survey of Deep Learning for Time Series Forecasting: Architectural Diversity and Open Challenges. (arXiv,2024)

[14] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J. & Sun, L. Transformers in Time Series: A Survey. (2022)

[15] Shi, J., Wang, S., Qu, P. & Shao, J. Time series prediction model using LSTM-Transformer neural network for mine water inflow. *Scientific Reports*. **14**, 18284 (2024)

[16] Liu, X., Liu, Q., Feng, S., Ge, Y., Chen, H. & Chen, C. Novel model for medium to long term photovoltaic power prediction using interactive feature trend transformer. *Scientific Reports*. **15**, 6544 (2025)

[17] Wu, G., Wang, Y., Zhou, Q. & Zhang, Z. Enhanced Photovoltaic Power Forecasting: An iTransformer and LSTM-Based Model Integrating Temporal and Covariate Interactions. (arXiv,2024)

[18] Piantadosi, G., Dutto, S., Galli, A., Vito, S., Sansone, C. & Di Francia, G. Photovoltaic power forecasting: A Transformer based framework. *Energy And AI*. **18** pp. 100444 (2024)

[19] Sarkar, T. & Chu, H. A Comparative Study of LSTM Efficiency vs. Transformer Power for Localized Time Series Forecasting. *Proceedings Of The 20th Conference On Computer Science And Intelligence Systems (FedCSIS)*. pp. 265-276 (2025)

[20] Al-Dahidi, S., Alahmer, H., Rinchi, B., Bani-Abdullah, A., Alrbai, M., Ayadi, O. & Al-Ghussain, L. Multistep PV power forecasting using deep learning models and the reptile search algorithm. *Results In Engineering*. **27** pp. 106265 (2025)

[21] National Institute of Standards and Technology - Engineering Laboratory & Building Energy and Environment Division Roof tilted array (2016)

[22] Lin Z, Zhou Q, Wang Z, Wang C, Bookhart DB, Leung-Shea M & A high resolution three-year dataset supporting rooftop photovoltaics (PV) generation analytics. Sci Data 12(1):63. doi:10.1038/s41597-025-04397-y (2025)

[23] Lüdecke, M., Bialojahn, M., Meinert, M. & Engel, B. Residential baseload-forecasting by applying recurrent neural networks with gated recurrent units on field data. *IET Conference Proceedings*. **2024**, 781-788 (2025)

[24] Istanbul Energy Inc. Solar Power Electricity Production Dataset (2020)

[25] UNISOLAR: An Open Dataset of Photovoltaic Solar Energy Generation in a Large Multi-Campus University Setting. IEEE (2022)

[26] Visual Crossing: Weather Data and Weather API (2025)

[27] Open-Meteo: Free Weather API for non-commercial use (2025)

[28] Alfin Syarifuddin Syahab, MS Hendriyawan Achmad: Solar Radiation Prediction using Long Short-Term Memory with Handling of Missing Values and Outliers doi:10.20895/INFOTEL.V16I3.1225 (2024)

[29] Chollet F: Deep learning with Python. Manning Publications, Shelter Island (2021)

[30] Haben, S., Voss, M. & Holderbaum, W. Core Concepts and Methods in Load Forecasting: With Applications in Distribution Networks. (Springer International Publishing,2023)

[31] Sousa TC, Barbosa RS & Short-Term Forecast of Photovoltaic Solar Energy Production Using LSTM. Energies 17(11):2582. doi:10.3390/en17112582 (2024)

[32] Gu J-C, Yang M-T & A Simplified LSTM Neural Networks for One Day-Ahead Solar Power Forecasting. IEEE Access 9:17174–17195. doi:10.1109/ACCESS.2021.3053638 (2021)

[33] Smith LN & A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay (2018)

[34] Neupert T., Fischer M., Greplova E., Choo K., Denner M.: Machine Learning for the Sciences - Department of Physics, University of Zurich (2022)

[35] Kim J, Obregon J, Park H, Jung J-Y: Multi-step photovoltaic power forecasting using transformer and recurrent neural networks. Renewable and Sustainable Energy Reviews 200:114479. doi:10.1016/j.rser.2024.114479 (2024)

[36] Hu Z, Gao Y, Ji S, Mae M, Imaizumi T: Improved multistep ahead photovoltaic power prediction model based on LSTM and self-attention with weather forecast data. Applied Energy 359:122709. doi:10.1016/j.apenergy.2024.122709 (2024)

[37] Zhai S, Likhomanenko T, Littwin E, Busbridge D, Ramapuram J, Zhang Y, Gu J, Susskind J: Stabilizing Transformer Training by Preventing Attention Entropy Collapse (2023)

[38] A. Vaswani, N. Shazeer, N. Parmar: Attention is all you need (31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.)

[39] Shahriari B, Swersky K, Wang Z, Adams RP, Freitas N de: Taking the Human Out of the Loop: A Review of Bayesian Optimization. Proc. IEEE 104(1):148–175. doi:10.1109/JPROC.2015.2494218 (2016)

[40] Si Z, Yang M, Yu Y, Ding T & Photovoltaic power forecast based on satellite images considering effects of solar position. Applied Energy 302:117514. doi:10.1016/j.apenergy.2021.117514 (2021)

[41] J. A. Illemobayo, O. Durodola, O. Alade, O. J. Awotunde, A. T. Olanrewaju, O. Falana, A. Ogungbire, A. Osinuga, D. Ogunbiyi, A. Ifeanyi, I. E. Odezuligbo, O. E. Edu & Hyperparameter Tuning in Machine Learning: A Comprehensive Review. Journal of Engineering Research and Reports DOI: https://doi.org/10.9734/jerr/2024/v26i61188 (2024)