Predicting Photovoltaic Power Production using High-Uncertainty Weather Forecasts

Tomas Polasek\textsuperscript{a,*} (Researcher), Martin Čadík\textsuperscript{a} (Researcher)

\textsuperscript{a}Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic

ABSTRACT

A growing interest in renewable power increases its impact on the energy grid, posing significant challenges to reliability, stability, and planning. Weather-based prediction methods help relieve these issues. However, their real-world accuracy is limited by weather forecast errors. To help resolve this limitation, we introduce the SolarPredictor model. Publicly available weather forecasts are used to predict solar power production by a target photovoltaic power plant. To achieve high prediction accuracy, the model is trained on genuine weather forecasts, including errors and mispredictions. Further, we introduce the SolarDB dataset, comprising one year of power production data for 16 power plants. The dataset includes hourly weather forecasts with seven days of history, allowing our model to anticipate errors in the meteorological features. The prediction accuracy is evaluated on a wide range of weather forecast ages, accurately reflecting real-world performance. The SolarPredictor model is compared against 17 techniques, reaching an average RRMSE of 6.15 for 1-day, 8.54 for 3-day, and 8.89 for 7-day predictions on the SolarDB dataset. Finally, we analyze the effects of weather forecast uncertainty on prediction accuracy, showing there is at least a 23% performance gap compared to using zero-error weather.

1. Introduction

Renewable energy utilization grows every year. Its increasing importance is notable by the average yearly growth rate of almost 0.68% within the European Union [1] and targets of reaching 32% coverage by 2030. However, many renewable energy sources are inherently dependent on meteorological factors, leading to challenging problems with energy grid balancing and stability [2]. Precise predictions are highly desirable from the point of smart power grids [3], competitive pricing [4], and energy utilization [5]. Many countries also require production forecasts [6], with deviations leading to penalty charges [7]. Predictive systems are used to alleviate these issues, thus guaranteeing a degree of certainty and allowing fallback to other power sources when needed [8, 9]. Several such systems specifically target photovoltaic power plants, predicting their power output directly from the input weather data [10]. However, while the prediction accuracy of weather forecasting techniques is constantly improving, even short-term weather forecasts are often weighed down by significant errors [11]. As a result, using error-free weather in predictive models leads to unexpected performance degradation in real-world scenarios when realistic weather forecasts are used.

Aims and Objectives: There are three primary goals we focus on in this paper:

- Predicting power production of solar power plants using localized meteorological data. We target specific locations to model their local properties, focusing on real-world performance by using weather forecast data. We solve this task with a novel SolarPredictor architecture (Sec. 2.2), taking into consideration the error-prone nature of weather forecasts.

- Creating a suitable dataset for the training of forecast-based prediction models. We propose the SolarDB dataset (Sec. 2.1) covering one year of detailed data for 16 photovoltaic power plants. We make this dataset publicly available to other researchers, facilitating further research in this area.

- Analysing of what affects the prediction accuracy and ways of improving it further. We use the SolarDB dataset to evaluate a wide range of scenarios – including weather, season, location, and forecast age – in order to determine which factors are important.

We make the dataset freely available for research purposes. For additional data, resources, and source codes, see the supplementary materials and the project website at cphoto.fit.vutbr.cz/solar.

\textsuperscript{*}Corresponding author

\textbullet\ ipolasek@fit.vutbr.cz (T. Polasek); cadik@fit.vutbr.cz (M. Čadík)
\textbullet\ cphoto.fit.vutbr.cz (T. Polasek); cphoto.fit.vutbr.cz (M. Čadík)

\textsuperscript{†}ORCID(s): 0000-0002-1441-0241 (T. Polasek); 0000-0001-7058-9912 (M. Čadík)
### Nomenclature

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD Clear Day</td>
<td>( \Delta ) Acceptance threshold</td>
</tr>
<tr>
<td>CNN Convolutional Neural Network</td>
<td>( f, \hat{f} ) Measurement frequency</td>
</tr>
<tr>
<td>DNI Direct Normal Irradiance</td>
<td>( P, P' ) Production over interval</td>
</tr>
<tr>
<td>DNN Deep Neural Network</td>
<td>( p_t ) Production at time index ( t )</td>
</tr>
<tr>
<td>GHI Global Horizontal Irradiance</td>
<td>( t ) Time of day</td>
</tr>
<tr>
<td>GPU Graphics Processing Unit</td>
<td>( w_t ) Weather features at time index ( t )</td>
</tr>
<tr>
<td>GRU Gated Recurrent Unit</td>
<td>( x_t ) Input features at time index ( t )</td>
</tr>
<tr>
<td>LSTM Long Short-Term Memory</td>
<td>( y_t/\hat{y}_t ) Ground-truth/prediction at time index ( t )</td>
</tr>
<tr>
<td>MLP Multilayer Perceptron</td>
<td></td>
</tr>
<tr>
<td>NN Neural Network</td>
<td></td>
</tr>
<tr>
<td>NWP Numerical Weather Prediction</td>
<td></td>
</tr>
<tr>
<td>OC Overcast Day</td>
<td></td>
</tr>
<tr>
<td>RF/RFor Random Forest</td>
<td></td>
</tr>
<tr>
<td>RNN Recurrent Neural Network</td>
<td></td>
</tr>
<tr>
<td>SeH Multi-output Model with History</td>
<td></td>
</tr>
<tr>
<td>Seq Multi-output Model</td>
<td></td>
</tr>
<tr>
<td>Sin Single-output Model</td>
<td></td>
</tr>
<tr>
<td>SPred Solar Predictor</td>
<td></td>
</tr>
<tr>
<td>SVM Support Vector Machine</td>
<td></td>
</tr>
<tr>
<td>TCN Temporal Convolutional Network</td>
<td></td>
</tr>
</tbody>
</table>

### 1.1. Related Work

#### Review of Solar Prediction Models

The prediction task takes a set of input values – including endogenous (internal) and exogenous (external) parameters – and calculates a power production forecast for the target time frame. The accuracy is largely determined by the precision of the input parameters and the prediction time horizon [12] – i.e., short \( t \leq 1 \) day, medium \( t \leq 14 \) days, and long-term \( t > 14 \) days. The prediction methods include physical, statistical, and machine learning models [13], their uses ranging from real-time scheduling [14] to long-term planning [15].

Physical models use the properties of the photovoltaic system without relying on historical data. The inputs include numerical weather predictions (NWP), monitoring data, and properties of the plant [16, 17], and output global horizontal irradiance (GHI) and direct normal irradiance (DNI), proportional to the plant’s production [18]. Although physically correct, these methods rely on precise meteorological data. They generalize poorly when provided with imprecise weather data [19], resulting in unreliable predictions when using real-world weather forecasts. These approaches include ASHRAE [20] and Hotell equations [21].

Statistical methods consider historical data and attempt to automatically derive a pattern – mapping the input variables to the output power production [22]. Common approaches used with solar power include Markov Chains [23], fuzzy logic [24] or, auto-regressive models [25, 26] such as the NARX [12] or NARMAX [27]. Although generally less complex than the physical models, the historical data allows them to better model the specifics of a power plant [28]. However, to do so, they require a large amount of data for each target plant [25], resulting in difficulties when rapid growth is needed.

Machine learning models use a trained mapping function. They derive the mapping through a process of training on a dataset of input and output samples [29]. Each prediction model is suited to a different use case. The models include Decision Trees [30], Random Forests (RFor) [31], and their ensembles [32], which are well suited to represent the variability of different power plant sites. Similarly, Support Vector Machines (SVM) [33] automatically determine
Photovoltaic Power Forecasting using Weather Forecasts

important input parameters and allow a degree of interpretability. Deep Networks (DNN) [34] are used sparingly since they provide a looser constraint on the time-series prediction data. Recurrent Networks (RNN) [35] model the temporal data, enhancing multi-step prediction stability [36]. Both Long Short-Term Memory (LSTM) [37] and Gated Recurrent Unit (GRU) [38] are used to further stabilize the training process. Conversely, Convolutional Networks (CNN) [39] are used to model the spatial aspect of the power production data [40] and perform localized associations [41].

Hybrid architectures combine both temporal and spatial constraints of the models [42]. The ConvLSTM [43] uses convolutional layers within the LSTM unit to provide translation invariance and a wider receptive field. The WaveNet [44] and Temporal Convolutional Network (TCN) [45] employ causal dilated convolutions to better model the temporal nature of the predictions. The Transformer [46] utilizes an attention mechanism, selectively choosing which time steps to use in the prediction. Ghimire et al. [39] and Zang et al. [47] use a CNN-LSTM hybrid for pattern recognition and time-series analysis to lower the data requirements. This is further enhanced when using attention to model the short-term and long-term temporal modes [48]. Each architecture has its respective merits and shortcomings. The RNNs are challenging to train due to vanishing gradients [49] and numerical instability [50]. CNNs can reach wide receptive fields but require deep architectures and large amounts of training data [41].

This study introduces a hybrid predictor motivated by the spatio-temporal nature of the power production signal. We name the model SolarPredictor. The model is designed to predict power production from low-reliability weather forecasts by adapting the UNet [51] to use residual connections [52]. The model performs a spatial analysis with 1D causal dilated convolutions [53], modeling the short and long-term relations. Then, the signal is passed through an aggregator to reconstruct the temporal nature of the prediction [44]. Details of this architecture are found in Sec. 2.2.

Review of Renewable Energy Datasets

A robust dataset is essential for the training of machine learning models, which require a large amount of data. Based on their coverage, solar power datasets can be divided into [54]: large-scale, planet-covering datasets with lower spatial resolution; regional, providing added detail and granularity, although at reduced scale; and site-specific, a set of locations, usually including power production values along with meteorological features.

The large-scale datasets include the Retrospective Analysis datasets – MERRA [55] and MERRA-2 [56] – focusing on climate data spanning the Earth from 1980. They provide hourly meteorological data with a spatial resolution of 50 km. The Derived Renewable Energy (EMHIRES) datasets comprise solar [57] and wind [58] power generation data. Both datasets cover Central Europe, providing 30 years of hourly data without weather features. Finally, the Solar Radiation Database (NSRDB) [59] is a solar-specific dataset with an hourly resolution for specific locations or a 4 km grid. The NSRDB contains solar irradiance information without direct power production data.

The OpenSolar [60] initiative provides tools for solar-specific regional datasets. The Solar Power Data (SPDIS) [61, 62] dataset consists of one year of 5-minute data covering 5,020 locations, including simulated power production and weather data based on the NSRDB [59]. The Solar Radiation Research Laboratory (SRRL) [63] dataset includes data from a single site dating from 1981, comprising 75 instruments at a 1–10-minute resolution. However, the site is located at 1829 m above sea level. The Microgen Database [64] dataset includes data from over 7,000 installations with only power generation data available. Lastly, the Dataport [65] dataset consists of production data of over 318 installations at a 1-minute resolution with no weather features.

The Comprehensive dataset [66] consists of 3 years of data for a single installation, providing production data at a 1-minute resolution along with seven weather variables. Data is sourced from on-site measurements and the Mesoscale Forecast System. However, the dataset does not include any weather forecast data. Finally, the DKA Solar [67] dataset includes 62 installations in central-northern Australia, consisting of production and weather features at 1-minute to 5-minute intervals, although no weather forecasts are provided.

Under optimal conditions, decades of historical data are needed for each power plant site to reach reliable predictions [54]. However, a dataset of this magnitude is currently not available and collecting years of data for each new installation is untenable. Moreover, to the best of our knowledge, all sufficiently sized datasets provide only zero-error weather, which is not representative of the real-world weather forecast used during inference.

Considering the outlined limitations, we present the SolarDB dataset (Sec. 2.1). It comprises one year of data for 16 solar power plants, providing power production at 5-minute intervals. Weather features include 12 meteorological variables available at an hourly resolution. Crucially, the dataset includes weather forecast features from commodity sources, providing seven days of historical forecasts with a 1-hour spacing between the data points.
Table 1

1-column Power Plant Sites: A list of photovoltaic installations included in the SolarDB dataset. Their reporting frequencies, nominal power, and inverter counts are provided. Each site is represented by 1-year of data starting with the Interval. The number of days without missing values is specified; other days contain gaps not exceeding four samples.

### 2. Method

To resolve the conflicting requirements of precise power predictions on one side and utilization of error-prone weather forecasts on the other, we propose the SolarPredictor system. It uses a hybrid neural network model, combining multi-scale spatiotemporal analysis with residual connections. The input consists of meteorological features for the target time frame and near historical production data. In contrast to other contemporary methods, the prediction is based on realistic weather forecasts and limited historical data for each site. This results in a more practical prediction task, taking into account the inherent uncertainty present in weather forecasts [68] and reflecting its real-world performance.

To train the SolarPredictor system, we introduce the SolarDB dataset covering production and weather data. Please see the supplementary materials and resources available at cphoto.fit.vutbr.cz/solar for further details and data.

### 2.1. The SolarDB Dataset

The dataset consists of 1 year of data from 16 photovoltaic installations. It includes power production, on-site weather forecasts, and additional meta information. The data spans 1.46 M power records, 16.3 M inverter power records, 139 K weather records, and 22.8 M forecast records. Through careful curation and pre-processing, it is further expanded into the overall 40 M records and made publicly available through the provided Python SolarDB API.

#### 2.1.1. Data Acquisition

A total of 16 installations spanning 258 power inverters are included in the dataset, as detailed in Tab. 1. The data was collected over the interval of two years, after which the continuous intervals of the highest quality were selected on a per-site basis. The SolarDB dataset includes four types of data: power, weather, exogenous, and meta-data. For a full description, please see App. A.

**Power** data is provided separately for the complete power plant and each of its inverters. Both pure production ($\text{Power}_{\text{DC}}$) and power after inversion ($\text{Power}_{\text{AC}}$) are provided in 5-minute intervals. The values are collected from various reporting systems, resulting in a non-uniform frequency (Tab. 1, Freq.). They are unified to the 5-minute frequency using linear interpolation for the in-between values as detailed in Sec. 2.1.2. We also provide status data for each installation and inverter, aggregating the status and error codes from reporting systems into a simplified codebook detailed in the supplementary materials.

**Weather** data consists of measured weather and weather forecasts with 12 features each, provided at a 1-hour resolution. Weather records are annotated with their age in hours. Measured weather represents the actual weather situation, while the forecasts may contain errors based on their age. Seven days of past forecasts are provided with a 1-hour step, resulting in 1 measured weather and $7 \times 24 = 168$ forecasts for each hourly time-point. Meteorological forecasts are collected from the Dark Sky API [69].
2.1.2. Data Pre-processing

The data pre-processing consists of filtration, frequency unification, interpolation, and extrapolation. First, erroneous measurements and inconsistent data are removed, converting both power and weather features into standard units and ranges. For details, see App. A and the extended quantities table in the supplementary materials.

Next, the frequency of the data points is unified to the common 5-minute interval. This step simplifies downstream tasks and generalized predictions. The missing values are calculated through linear interpolation and marked with a special Interpolated flag. However, only values between two valid records are interpolated to ensure high quality.

Finally, missing Power and Weather data is completed through a sequence of interpolation and extrapolation steps. This ensures data availability for all time points. In contrast to the frequency unification, the weather features are kept at an hourly resolution to reduce the size of the dataset. Linear interpolation is applied to intervals of \(1\) to \(2\) missing records. Any remaining missing time steps are then extrapolated by using the next closest day. Records created in this way are marked as Interpolated or Extrapolated, respectively.

2.1.3. Data Augmentation

The dataset is augmented with complementary Exogenous features, including temporal, solar, and clarity components. Temporal features focus on the periodic properties of the production. Importantly, each Weather record is provided with its Age implying its veracity. The solar features comprise simulated properties of the Sun – altitude, azimuth, and irradiance [18] – calculated with the PySolar library [70] using the precise position of the power plant.

The apparent clarity is expressed by the daily clarity feature, splitting the data into clear (CD) and overcast (OC) days. The division is made on a per-day basis and is consistent with the concept of clear-day models [18]. The clarity is defined through the clear-day likelihood \(L_{CD}\) as

\[
P'(t) = \max_t \left[ P(t) \left( 1 - (2t - 1)^2 \right) \right],
\]

\[
L_{OD}(D) = \text{Var} \left[ \frac{1}{\text{Var} [P(t)]} \left( P'(t) - P(t) \right) \right],
\]

\[
L_{CD}(D) = \max \left[ 0, \min \left[ 1, 1 - L_{OD}(D) \right] \right],
\]

where \(t \in [0, 1]\) is the time of day, \(P'\) is the estimated clear-day production, \(L_{OD}\) and \(L_{CD}\) are the likelihoods of the day being overcast and clear, respectively, and \(D\) is the target day. A day is marked as clear when \(L_{CD} >= 0.5\). This results in a roughly equal split over the 16 power plants – 50.3\% of the days are marked as clear and 49.7\% as overcast. For qualitative comparison, see the random selection of clear and overcast production curves in App. A.

2.2. The SolarPredictor System

As visualized in Fig. 1, its inputs consist of historical power production and weather features for each target time step. To better reflect the weather uncertainty, forecasts aged \((1 \ldots 168)\) are used, denoting measured weather with age 0. Predictions are performed on a per-day basis, with outputs representing power production for each 5-minute step.

Figure 1: (Color, 2-column) System Overview: The SolarPredictor system is trained on the SolarDB dataset. The inputs consist of historical power (a) and weather features (b). Power predictions are estimated in daily blocks (c). The initial Power data (a) is historical, while consequent predictions recurrently use the previous outputs (d). Notably, the predictions use weather forecasts with increasing age (b, blue), which leads to progressively increasing weather error (b, orange).
2.2.1. Predictor Architecture

The SolarPredictor model architecture is visualized in Fig. 2. It uses a fully convolutional architecture consisting of two blocks: Residual UNet and Residual Composer. The upper Residual UNet performs a multi-scale analysis of the time series and produces the residual signals. While the lower Residual Composer constructs the prediction estimate by gradually adding up the residual signals. For parametrization of the architecture, see the supplementary materials.

![SolarPredictor Model Diagram](Image)

**Figure 2:** (Color, 2-column) SolarPredictor Model: The model combines a Residual UNet for time-series analysis (top) with a Residual Composer (bottom). The inputs consist of weather features and historical power data, producing the power prediction estimate. ↓, ↑ represent the down/up-sampling operations, while ⇕ is the conditional re-sampling.

Residual UNet is inspired by the UNet [51] and ResNet [52] architectures (Fig. 2, top). It contains a cascade of ResBlock units with 1d causal convolutions [53] divided into four stages. First, both Power and Weather data are pre-processed and concatenated. Next, the contractive stage on the left performs a down-sampling (↓ ResBlock) to extract spatio-temporal information, which passes through the bottleneck stage. Finally, the expansive stage performs the multi-scale analysis (↑ ResBlock), combining previous outputs with skip connections from the contractive stage.

Residual Composer is inspired by the residual modules in WaveNET [44], aggregating the individual residual signals into the estimated power prediction (Fig. 2, bottom). A pre-processing phase takes both the Power and Weather features and reshapes them into the required output dimensions. Next, a series of residual aggregation modules (ResAgg) add the residual signals from the upper part of the model to the ongoing signal. The adaptive up-down sampling module (⇕) is used to unify the dimensions between the residual tiers. Finally, the estimated power production is produced by stacking the aggregation modules in a sequence, gradually building the output signal.

2.2.2. Model Training

The proposed SolarPredictor architecture is further enhanced with data augmentations to improve predictor accuracy and training stability. A total of six techniques are used, and their efficacy is further explored in Sec. 3.

**Filtering & Scaling:** Outlier analysis and data scaling is performed to prevent anomalies and stabilize the model training. The outliers are detected using absolute z-scores calculated over the base Power and Weather features. Samples with a z-score above an empirical threshold of $\gamma = 20$ are considered anomalous. These are consequently removed (Power) or set to zero (Weather). Each Power and Weather feature is then separately processed with a Positive Robust scaler [71], defined as

$$\hat{v} = \frac{v - \mu_V}{V_{75th} - V_{25th}} - \min \left[ \frac{V}{V_{75th} - V_{25th}} \right] - \mu_V$$

where $\hat{v}$ is the scaled output, $v \in V$ is the input, $V = \{v_1, v_2 \ldots\}$ is the set of all inputs, $\mu_V$ is the mean value, and $V_{25th}, V_{75th}$ are the 25th and 75th percentiles of $V$.

**Feature augmentation:** Highly correlated features (Sec. 3.1) are removed to reduce the model size and accelerate the training procedure. Inverted variants of the Cloud Cover and Visibility features are calculated and scaled using the Positive Robust scaler (Eq. 4). Finally, cyclic time features are generated, facilitating phase detection. This approach is similar to the positional encoding in Transformer models [46]. We generate two features containing the sine and cosine transforms of selected temporal features. For details, please see App. B.
Weather Sampling: Training the model with only zero-error weather leads the model to implicit trust in the input weather, resulting in a loss of accuracy in real-world scenarios. The cause of these errors is connected to the correlation between the forecast age and prediction error (Sec. 3.7). However, directly using the weather forecasts from the beginning of the training leads to error accumulation during the recurrent predictions. Therefore, we train the model on both the measured and forecast weather features, utilizing multiple forecast ages.

Three types of sampling schemes (App. B) are used to construct the training data. The Measured weather samples use the zero-error weather features for each target time point. The Forecast samples utilize weather forecasts delayed by a constant offset \(d\). Since downstream applications receive fresh forecasts at a set frequency, the training uses multiple delays \(d \in \{1D, \cdots, 7D\}\). These delays allow the model to learn the relationship between the age of the weather and its veracity. Lastly, the Realistic samples simulate the real-world scenario by using the latest available weather forecast at the time of prediction. Thus, the age of the features increases linearly, as does its error.

Sample Weighing: The input data contains imbalanced classes with under-represented weather conditions. A sample weighing scheme is used to account for their frequency. It consists of two parts: clear-day weights \(w_{cg}\) and temporal weights \(w_t\). The clear-day weights \(w_{cg}\) balance the data based on day clarity, based on Eq. 2. The value of \(w_{cg}\) is minimized for clear days while being at its maximum for overcast days. Conversely, the temporal weight \(w_t\) is designed to favor samples based on their relative frequency. Finally, the total sample weight \(w = w_{cg} \cdot w_t\) is calculated for each sample in the training set and used as a loss multiplier. For detailed calculation, please see App. B.

Importance Pruning: Many of the samples are assigned with small weights \(w\), resulting in a low contribution to the overall training. However, completely removing them results in a degradation in prediction performance, as evaluated later in Sec. 3.5. Thus, we use stochastic importance pruning instead of simply removing the samples.

Three acceptance probabilities govern this procedure: \(\delta_d, \delta_h, \delta_s\), representing day, hour, and smooth (5-minute) samples, respectively. Depending on each sample’s prediction window, it is assigned with a threshold \(\Delta := \delta_d | \delta_h | \delta_s\). Then, a uniform random variable \(A \sim U(0, 1)\) is used to determine whether a sample is accepted (\(A \leq \Delta\)) or rejected. By choosing \(\delta_d = 1.0, \delta_h = 0.5, \) and \(\delta_s = 0.01\), we see a 95% reduction in data without noticeable degradation in performance. The stochastic nature of the pruning also provides regularization and improves training (Sec. 3.5).

Training Procedure: The SolarPredictor model is trained on the SolarDB dataset. The ADAM [72] optimizer is used with the AMSGrad [73] modification, utilizing the reduce-on-plateau technique. Through empirical observation, the initial learning rate is set to \(lr = 0.005\) while setting \(\beta_1 = 0.9\) and \(\beta_2 = 0.999\) – the values recommended by the authors [73]. The training proceeds with a mini-batch size of 256, shuffling samples at the end of each epoch. Further, Smooth Loss \(\mathcal{L}\) is used as the training objective, combining the properties of Mean Squared Error \(\mathcal{L}_M\), Huber Loss \(\mathcal{L}_H\), and Differential Smooth Loss \(\mathcal{L}_\Delta\). For details, see App. B. Lastly, Gaussian noise (\(\mu = 0\) and \(\sigma = 0.0005\)) is injected into the power history data, improving the training stability and generalization of the model.

3. Results and Discussion

The results of the proposed system are presented in this section. First, a deeper look at the SolarDB dataset is provided, focusing on its statistical properties. Second, the accuracy of the SolarPredictor model is evaluated. To better convey its properties, we compare it against other predictors from the point of accuracy, performance, and size. Next, the proposed model improvements are gauged in an ablation study and further studied in a cross-validation experiment. Finally, the relevance of the presented results is discussed, along with limitations and potential uses.

3.1. Dataset Overview

An overview of the SolarDB dataset can be found in Fig. 3. The 16 power plants show the expected sinusoidal tendency consistent with the Earth’s temperate zone. Notably, the similar precipitation and temperatures throughout the year are caused by the physical proximity of the power plants, facilitating experiments with their distance.

![Figure 3: (Color, 2-column) SolarDB Overview](image)

Each graph displays a year of data for one of the 16 power plants from January (left) to December (right). The weekly power production is marked in blue; grey dots represent daily values, along with temperature (orange) and precipitation (green). The data is aligned, starting date marked by the red vertical line.
3.2. Feature Analysis

There is a wide range of meteorological features contained in the SolarDB dataset (App. A). However, their importance is uncertain. To determine which features are essential in the prediction task, we calculate the Pearson correlation [74] between the weather features and the power production. The resulting matrix is visualized in Fig. 4. It exhibits a strong correlation between power, temperature, and sun-related features. Further, the TimeX feature corresponds with the sinusoidal tendency of the production, which usually peaks around noon. As expected, during clear days (orange bars), the correlation is higher since the pattern is not disrupted by the weather conditions. Comparatively, the overcast days (blue bars) have a higher relative correlation with the weather features and are less dependent on the time. Finally, shifted power productions (PwrHist) are highly correlated for both clear and overcast days, indicating that historical production data may lead to improved accuracy. This effect is further analyzed in Sec. 3.4.

![Feature Correlation](image)

Figure 4: (Color, 1/1.5-column) Feature Analysis: The correlation matrix (left) shows Pearson correlation [74] for weather features. Power importance (right) details how vital these features are to the prediction. The Power (1) column contains correlations of the Power to all other features – filled bars show average results for all days, while the vertical bars represent the Clear (orange) and Overcast (blue) days. The following columns contain feature analysis using single (RForSin) and multi-value (RForSeH) Random Forest prediction models (Sec. 3.4) using normalized Gini Importance [75] as the metric.

To corroborate our findings, deeper analysis is performed by utilizing a single-value (RForSin) and multi-value (RForSeH) Random Forest models, each configured as detailed in Sec. 3.4. The RForSin receives weather features for a single time point, while the RForSeH is provided with weather features for the entire day along with historical power production. To increase the significance of the results, 30 cross-validation runs are performed and averaged. The importance analysis is based on the normalized Gini Importance [75]; details can be found in App. C.

The results for both RForSin and RForSeH models (Fig. 4) show the mean ($I_f$, solid bar) and the variance ($\sigma^2_f$, tail) values. Both temperature and humidity are strongly utilized, yet precipitation and cloud cover features are less critical. Notably, the single-value predictor (RForSin) mostly depends on the humidity, time, and sun features – possibly for the day-cycle phase detection. Conversely, the multi-value predictor (RForSeH) replaces the temporal features with power history, suggesting that recurrent models may utilize past values for phase detection.

3.3. Evaluation Procedure

The accuracy of the proposed model is evaluated using the task of power prediction. Unless otherwise specified, the inputs consist of the weather forecasts. Thus, the age of the weather features increases monotonically (Fig. 1, b), resulting in progressively increasing forecast error, as it would in real-use scenarios. Further, in multi-day predictions, the predicted power is used recurrently – supplying the outputs of the previous prediction step as an input to the next step. This approach better approximates the real-world use case and shows any potential error accumulation issues.

Several quantitative metrics are used to gauge the accuracy of the models. For weather and power data, the errors are calculated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Relative RMSE (RRMSE), and Coefficient of Determination ($R^2$). Hourly aggregates of weather (WError) and power (PError) errors are also used to gauge the overall performance. For implementation details, please see App. C.
A cross-validation protocol is used in all experiments. The cross-validation procedure ensures that the results are representative of various seasons and weather conditions. Each individual experiment consists of 12 separate runs. In each run, the one year of data is split into training and test sets of 11 and 1 month, respectively. The test set is rotated so that each month is selected exactly once within the 12 runs. The prediction results for the 12 test sets are then concatenated, making up one full year of predictions, which are then used in the evaluation.

Prediction models are implemented using the Python programming language, utilizing the Tensorflow [76] framework, SciPy [77], and SciKit-Learn [71] packages. Training and inference measurements are performed on a work node equipped with an Intel Xeon 12-core @ 2.2GHz CPU, 96GiB DDR4 @ 2666GHz memory, and an NVidia GeForce RTX 2080 Ti GPU. Specific model and training parameters can be found in the supplementary materials.

### 3.4. Model Comparison

A performance baseline consisting of 24 prediction models are tested on the SolarDB dataset. The results are presented in Tab. 2. Where applicable, alternative input and output vectors are also tested, resulting in single-value (Sin), multi-value (Seq), and multi-value with history (SeH) variants. To ensure a fair comparison, all of the models utilize the same set of augmentations and weather sources.

The evaluation is performed on power plant #8, chosen as the overall average within the SolarDB dataset. Each model is first trained using the cross-validation protocol. Then, predictions for 1 to 10 days ahead are calculated for each of the 12 testing months. The months are concatenated into a single continuous array, making up a full year of predictions. Next, the predicted data is split into Clear and Overcast subsets (Sec. 2.1.3). This allows separate evaluation in each category (Clear, Overcast) as well as the overall performance (All). Finally, the performance metrics are calculated for each of the 1 to 10-day sets and then averaged into a single aggregate value.

<table>
<thead>
<tr>
<th>Model</th>
<th>Power</th>
<th>Weather</th>
<th>Out</th>
<th>Clear Days</th>
<th>Overcast Days</th>
<th>All Days</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RMSE</td>
<td>PError</td>
<td>RMSE</td>
</tr>
<tr>
<td>Classical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM Sin</td>
<td>N/A</td>
<td>1:5</td>
<td>1:5</td>
<td>19.09</td>
<td>70.361</td>
<td>42.17</td>
</tr>
<tr>
<td>SVM SeH</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>9.39</td>
<td>26.909</td>
<td>24.19</td>
</tr>
<tr>
<td>RF Sin</td>
<td>N/A</td>
<td>1:5</td>
<td>1:5</td>
<td>8.56</td>
<td>22.995</td>
<td>33.65</td>
</tr>
<tr>
<td>RF Seq</td>
<td>N/A</td>
<td>288:5</td>
<td>288:5</td>
<td>7.47</td>
<td>26.175</td>
<td>25.92</td>
</tr>
<tr>
<td>RF SeH</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>7.42</td>
<td>24.373</td>
<td>19.70</td>
</tr>
<tr>
<td>DNN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNN Sin</td>
<td>N/A</td>
<td>1:5</td>
<td>1:5</td>
<td>17.09</td>
<td>42.106</td>
<td>30.29</td>
</tr>
<tr>
<td>DNN Seq</td>
<td>N/A</td>
<td>288:5</td>
<td>288:5</td>
<td>12.62</td>
<td>25.500</td>
<td>28.10</td>
</tr>
<tr>
<td>DNN SeH</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>7.42</td>
<td>24.373</td>
<td>19.70</td>
</tr>
<tr>
<td>Recurrent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM Van</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>17.02</td>
<td>40.179</td>
<td>31.31</td>
</tr>
<tr>
<td>LSTM Sta</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>17.08</td>
<td>40.796</td>
<td>31.03</td>
</tr>
<tr>
<td>LSTM MCNN</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>16.97</td>
<td>36.410</td>
<td>24.78</td>
</tr>
<tr>
<td>LSTM MC</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>17.38</td>
<td>36.264</td>
<td>34.56</td>
</tr>
<tr>
<td>LSTM MBid</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>10.03</td>
<td>19.181</td>
<td>31.69</td>
</tr>
<tr>
<td>Pred SeRc</td>
<td>36:40</td>
<td>4:10</td>
<td>8:5</td>
<td>16.34</td>
<td>47.334</td>
<td>26.68</td>
</tr>
<tr>
<td>Pred PaRc</td>
<td>36:40</td>
<td>36:5</td>
<td>36:5</td>
<td>10.24</td>
<td>16.051</td>
<td>20.86</td>
</tr>
<tr>
<td>Convolutional</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pred CNN</td>
<td>288:5</td>
<td>72:20</td>
<td>288:5</td>
<td>9.18</td>
<td>38.394</td>
<td>15.85</td>
</tr>
<tr>
<td>Pred CNNNO</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>8.06</td>
<td>30.601</td>
<td>20.25</td>
</tr>
<tr>
<td>Pred UNet</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>7.85</td>
<td>28.160</td>
<td>20.64</td>
</tr>
<tr>
<td>Pred TNC</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>7.46</td>
<td>28.643</td>
<td>19.03</td>
</tr>
<tr>
<td>Pred Wave</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>7.89</td>
<td>25.623</td>
<td>24.85</td>
</tr>
<tr>
<td>Pred Trf</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>9.50</td>
<td>27.625</td>
<td>18.29</td>
</tr>
<tr>
<td>Pred</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>5.68</td>
<td>16.886</td>
<td>16.16</td>
</tr>
<tr>
<td>Enhanced</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFor Enh</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>6.90</td>
<td>19.838</td>
<td>17.34</td>
</tr>
<tr>
<td>S Pred Enh</td>
<td>288:5</td>
<td>288:5</td>
<td>288:5</td>
<td>4.50</td>
<td>15.308</td>
<td>11.29</td>
</tr>
</tbody>
</table>

Table 2 (2-column) Prediction Models: The experiments are split by model type into Classical, DNN, RNN, CNN, and Enhanced categories. The sizing of the input (Power, Weather) and output (Out) vectors are specified as count : stride, where the count is the number of values provided and stride the time step in minutes – e.g., 288:5 covers 1440 minutes, i.e., one day. The performance is evaluated separately for Clear and Overcast days, with All Days covering both categories.
The result of the experiments is visualized in Fig. 5. The best-performing model among the Classical approaches is the Random Forest. Notably, expanding the model’s range of inputs from a single value (Sin) to a day (Seq) and adding historical data (SeH) leads to an overall improvement of 36.61%. This result indicates that a broader context is critical to prediction accuracy. A similar tendency is also seen in the DNN models, further confirming this theory.

Surprisingly, most of the LSTM [49] Recurrent models perform poorly. There is notable error accumulation in the longer prediction time-frames. Experiments with GRU [78] units lead to similar results. Although they achieve comparable Clear Day performance, they fail to predict the Overcast Days. Hybrid models combining LSTMs phase detection with DNN signal processing are an attempt to balance the results – serial (SPredSeRc) and parallel (SPredPaRc) configurations. However, both variants are still worse compared to the Random Forest models.

Purely convolutional architecture (SPredCNNO) and its combination with dense layers (SPredCNN) achieve RRMSE ≈ 15.41 and 13.11, respectively, with a significant accuracy trade-off between clear and overcast days. Further, we experiment on four backbone architectures based on the UNet (SPredUNet) [51], Temporal Convolutional Network (SPredTCN) [53], WaveNet (SPredWave) [44], and the Transformer (SPredTrf) [46]. Although these models show a minor improvement (2.11% - 7.44%) to the Clear Day accuracy of SPredCNN, they are generally (9.69% - 31.42%) worse when compared to the SPredCNN.

The SolarPredictor model (SPred) utilizes the architecture proposed in Sec. 2.2.1. It achieves an average 14.76% improvement in All Day accuracy over the second-best Random Forest model (RForSeH), gaining 23.45% on clear days and 16.27% on overcast days. The performance improvement is even more notable for the enhanced version of the model (SPredEnh), which is further analyzed in Sec. 3.5.

Finally, we also compare the models from the point of time and space complexity, visualized in Fig. 6. The results show that the Random Forest model requires 340× more memory and is 2.9× slower than the SolarPredictor model. Please see the paper supplement for detailed additional comparison data.
3.5. Ablation Study

To better demonstrate the effect of the augmentations, we present an extensive ablation study covering the enhancements proposed in Sec. 2.2.2. The experiments point to their general applicability, improving the prediction accuracy for both the SolarPredictor (SPred) and the second-best Random Forest model (RForSeH). For additional experiments and complete data, please see the supplementary materials.

<table>
<thead>
<tr>
<th>Model</th>
<th>Meas</th>
<th>Fore</th>
<th>Real</th>
<th>5.68</th>
<th>16.886</th>
<th>5.54</th>
<th>17.911</th>
<th>5.42</th>
<th>18.216</th>
<th>5.41</th>
<th>18.022</th>
<th>5.54</th>
<th>17.911</th>
<th>5.42</th>
<th>18.216</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPredAugmented</td>
<td>✓</td>
<td></td>
<td></td>
<td>5.68</td>
<td>16.886</td>
<td>5.54</td>
<td>17.911</td>
<td>5.42</td>
<td>18.216</td>
<td>5.41</td>
<td>18.022</td>
<td>5.54</td>
<td>17.911</td>
<td>5.42</td>
<td>18.216</td>
</tr>
</tbody>
</table>

Table 3 (2-column) Ablation Study: Experimental results dealing with the efficacy of the proposed augmentations. The baseline model (top) is first (middle) enhanced with scaling (Scl), sample selection (Sel: Outliers, Weights), and augmentation (Aug: Cyclic, Feature selection). This improves Clear Day performance. Next, training data augmentation is introduced (bottom) with sampling (Smpl: single per Day, All), measured weather (Meas), forecasts of up to a given age (Fore), and realistic samples (Real). Using the realistic sampling scheme improves both Clear and Overcast days.

The results of the ablation study are presented in Tab. 3. The starting model (SPredBaseline) is gradually modified with feature and forecast augmentations. Note that the baseline already predicts Clear Day accurately with RRMSE ≈ 9.13, while the accuracy for overcast days is only 32.33. The overall prediction performance is improved by over 44% by gradually adding scaling, outlier removal, sample weights, cyclic, and solar features. To further improve the performance, we consider the utilization of weather forecasts during training. Notably, the augmented model limited to using only a single training vector per day (SPredMeasured) leads to degraded prediction performance. Adding intraday sampling (SPredAllData) reduces the training data requirements to only 5.23% of those used by the augmented model while improving the performance by 4.58%. Up until this point, the model was trained only on the measured weather without seeing any inaccurate weather forecasts. However, simply training the SolarPredictor on the forecasts results in lower reliance on weather features, decreasing the Clear day accuracy.

<table>
<thead>
<tr>
<th>Model</th>
<th>Meas</th>
<th>Fore</th>
<th>Real</th>
<th>11.29</th>
<th>32.480</th>
<th>15.63</th>
<th>42.795</th>
<th>16.16</th>
<th>44.351</th>
<th>26.04</th>
<th>56.386</th>
<th>16.16</th>
<th>44.351</th>
<th>26.04</th>
<th>56.386</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPredAugmented</td>
<td>✓</td>
<td></td>
<td></td>
<td>5.68</td>
<td>16.886</td>
<td>5.54</td>
<td>17.911</td>
<td>5.42</td>
<td>18.216</td>
<td>5.41</td>
<td>18.022</td>
<td>5.54</td>
<td>17.911</td>
<td>5.42</td>
<td>18.216</td>
</tr>
</tbody>
</table>

Table 4 (2-column) Weather Ablation: Experiments with training on weather forecast data, using the SPredAugmented (Tab. 3). Starting with measured weather (Meas), additional forecast ages (Fore) are added in the middle section. Diminishing effects are notable at around five days of data. The deterioration is corrected by using the realistic sampling (Sec. 2.2.2).

The study presented in Tab. 4 shows the effect of training the model on different forecast ages and quantities. They confirm that multiple forecast ages improve the overall accuracy. However, diminishing returns are seen at five days of data. This problem is solved by using the realistic sampling (Sec. 2.2.2) – training the model on linearly aging forecast data. The final model reaches improvements of 3.23% and 27.77% for Clear and Overcast days, respectively. This disparity indicates that the sampling allows the model to better gauge the forecast uncertainty. The importance of training on forecast data is also visible when comparing it against the original SPredAugmented model. Its overall results are improved by 28.27%. Finally, the experiment concerning vector sizing, presented in the supplementary materials, shows that providing the model with historical data is highly beneficial, leading to a 15.86% improvement.

Polášek, Čadík: Manuscript submitted to Elsevier
3.6. Cross-Validation Results

The SolarPredictor model, along with the proposed augmentation, was shown to improve the prediction accuracy. However, the generalization of this approach is uncertain. Thus, we perform a cross-validation experiment spanning all 16 power plants from the SolarDB dataset to confirm the general validity of these conclusions. The two best-performing models were chosen – RFEnh and SPredEnh (Tab. 2) – both with the same set of proposed augmentations, improving their accuracy from the base models by 9.84% and 28.27%, respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>#1 RRM</th>
<th>#2 RRM</th>
<th>#3 RRM</th>
<th>#4 RRM</th>
<th>#5 RRM</th>
<th>#6 RRM</th>
<th>#7 RRM</th>
<th>#8 RRM</th>
<th>#9 RRM</th>
<th>#10 RRM</th>
<th>#11 RRM</th>
<th>#12 RRM</th>
<th>#13 RRM</th>
<th>#14 RRM</th>
<th>#15 RRM</th>
<th>#16 RRM</th>
<th>All Power Plants</th>
</tr>
</thead>
</table>

Table 5: (2-column) Cross-Validation Experiments: Evaluation of the two best-performing models – RFEnh and SPredEnh – for each of the 16 power plants. Each column represents a single power plant with the prediction RRMSE (RRM). The final three columns show the overall results, calculated as an average over all power plants.

The results of this experiment are presented in Tab. 5. The SolarPredictor model (SPredEnh) provides consistently higher accuracy than the second-best model (RFEnh), ranging from 32.18% to 57.81%. On average, the SolarPredictor model is 43.73% more accurate. A detailed overview of the performance is visualized in Fig. 7. As expected, the Clear days are overall easier to predict (D). Interestingly, the relative difficulty of Clear and Overcast days is different for each power plant, which is possibly a combination of site-specific environmental and meteorological factors. Finally, a calendar view of the predictions is provided in Fig. 8. The results for all 16 power plants are overlaid, averaging their daily metrics. Overall, the calendar for RFEnh (A) is darker in hue, corresponding to its lower prediction accuracy.

3.7. Prediction Difficulty

This section considers the cross-validation data to better understand which factors determine the difficulty of prediction. Essentially, the goal is to analyze the results detailed in Sec. 3.4 and 3.6 from the point of conditions leading to inferior prediction performance. This experiment allows deeper understanding of various modes of failure the prediction model exhibits and sets an expectation for the feasibility of future prediction accuracy. Weather Accuracy: Aspects contributing to the prediction difficulty are evaluated in Fig. 9. First is the length of the prediction horizon (A). As expected, the further into the future a model predicts, the lower its average accuracy. The cause of this drop-off is probably caused by the increasing uncertainty in weather forecasts (bottom inset) since all types of models are affected. This hypothesis is further supported by Fig. 9 (B), correlating the increased weather error with the lower prediction accuracy.
Photovoltaic Power Forecasting using Weather Forecasts

Figure 8: (Color, 2-column) Cross-Validation Calendar: Evaluation of predictions spanning the 16 available power plants for the RForEnh (A) and SPredEnh (B) models. Individual cells represent aggregate daily metrics, which are calculated as a mean over all power plants. Months are bracketed by vertical lines, while rows represent days of the week. Each cell conveys three types of information, from outside to inside: RRMSE, prediction error, and clarity – light/dark for clear/overcast days. Notably, the overcast days are more challenging to predict, corresponding with lower accuracy.

Figure 9: (Color, 2-column) Prediction Difficulty: The effect of various factors on the prediction accuracy for models from Tab. 2. (A) examines the relationship between prediction length and accuracy – the bottom inset displays the growing weather forecast error. (B) shows the detrimental effect of weather error on the prediction error, while (C) relates the daily prediction error and the RRMSE. Finally, (D) investigates the correlation between ground truth and predicted values – the diagonal line represents a perfect prediction accuracy. For the complete data, please see the supplementary materials.

Under optimal conditions, the curves in Fig. 9 (A) and (B) would be low and as flat as possible. However, this is an ill-posed problem, implying that the model can completely compensate for the forecast error. Nevertheless, the prediction model can be expected to utilize up-to-date forecasts more efficiently. This can be quantified by the relative accuracy decrease over the 1 – 10 day predictions. Using the data available in the supplementary materials, this results in an average RRMSE increase of 32.35% (RForSeH), 34.65% (RForEnh), 14.84% (SPred), and 51.06% (SPredEnh). The increased percentage indicates that the model enhancements lead to better utilization of forecast uncertainty.

Fig. 9 (C) analyzes the modes of model accuracy by plotting the relationship between RRMSE and PError. Thus, the models can be split into three categories. The smoothing models predict correctly on average, matching the lower frequency parts of the signal. Conversely, the pinpoint models try to match the high-frequency changes. Finally, balanced models cover both, leading to models with curves close to the diagonal – e.g., RForEnh and SPredEnh.

Fig. 9 (D) presents a correlation between ground truth and predicted power production. The results show that extreme values are more challenging to predict. This is caused by the rarity of anomalously low or high production values in the training data. The probability of their occurrence is low. Thus, models avoid them in deference to more probable results. The problem is improved by sample weighing, as seen by the SPredEnh being closer to the diagonal.
Seasonality and Climate: Seasonality presents a noticeable effect on the difficulty of prediction, as seen by the results displayed in Fig. 8. The days categorized as Overcast are concentrated during the winter months, directly corresponding to increased prediction error and weather forecast error (see supplementary materials). Some power plants are also inherently more difficult to predict, as seen in Fig. 7. The causes include varied climates and location-specific properties, such as the position of the plant and its orientation.

Table 6

(2-column) Forecast Difficulty: A comparison of training/testing on forecasts (RForEnh, SPredEnh) and measured weather (RForEnhFW, SPredEnhFW). The results show that using zero-error weather is 23% easier compared to weather forecasts.

Zero-Error Weather: Tab. 6 analyzes the disparity in prediction accuracy between weather forecasts and measured weather. The same models are trained and evaluated on weather forecasts (RForEnh, SPredEnh) and measured weather (RForEnhFW, SPredEnhFW). Notably, the comparison shows a 23.03% and 23.90% difference in accuracy for RForEnh and SPredEnh, respectively. This confirms that predictions made using weather forecasts are considerably more difficult. In conclusion, we note that with improved weather forecasts, the prediction accuracy can potentially improve by a further 23%, thus emphasizing the need for high-quality weather forecasts.

3.8. Limitations and Discussion

The SolarPredictor system has several limitations. First is the apparent dependence of the models on weather forecast data. Although the models expect unreliable forecast data – due to the training regime – their performance is limited by the veracity of the weather forecast. For the SolarDB dataset, we quantify the gap between the performance possible with completely accurate weather at around 23%. Further, the model is also dependant on training data. While the requirements are lower compared to other models, the SolarPredictor still needs at least a month of historical data for reliable predictions. For a wider deployment, a transfer learning scheme could be used to expedite the process. A base model can be trained on the SolarDB corpus and then fine-tuned on several days of data for the target power plant. The resulting model can then be used for production and re-trained as more data becomes available.

To test our approach in practice, we integrated the SolarPredictor into a commercial solar management system for private residences. The resulting software allowed the user to monitor their photovoltaic panels and use our predictions to gauge how much power they can expect in the horizon of 1 – 10 days. To allow quick deployment, we utilized the transfer learning approach outlined above. A base model was trained on the SolarDB dataset. Then, adding a new power plant consisted of the following steps. First, a ramp-up phase included re-training the model once a day. The pre-trained base model was fine-tuned on all up-to-date available data. Thus, as the power plant collected more data, the training corpus grew. Finally, the second phase began when a month of historical production data was available – the re-training frequency was reduced to a monthly basis. The resulting system was able to quickly adapt to new power plants while also allowing the expansion of the pre-training dataset.

However, there are other uses for SolarPredictor, specifically in the power grid management sector. Transmission system operators (TSO) have various means of controlling the power grid to ensure stable and reliable operation [79]. One includes the control reserves – backup energy sources that are employed when power consumption fluctuates or due to a power plant outage. Another way is to dis/connect interruptible loads – large consumers which allow outages. Finally, redispatch measures allow the operators to adjust the feed from connected power plants to reduce power congestion and restore balance. However, a growing percentage of renewable energy sources – such as photovoltaic or wind power plants – introduces inherent instability into the grid. These sources are heavily dependent on weather conditions and, as such, are difficult to predict. With the SolarPredictor system, we provide a step towards increasing power grid stability. If employed, the system allows the TSOs to expect excess or lack of energy in advance. Therefore, interruptible loads can be notified, and an early plan for control reserve activation can be put in place.

The system is also useful concerning the requirements of the growing electromobility industry. Longer journeys utilizing electric vehicles (EV) require careful route planning to account for the shorter drive distance. Similar concerns will be even more prevalent once EVs are used for logistics – i.e., automated trucks and deliveries. The recharge stations require a stable energy supply, and long-term storage is currently not available. However, the SolarPredictor system can be used to predict power availability along the route, thus allowing automated routing based on energy availability.
4. Conclusions

In this study, we proposed a deep-learning system called SolarPredictor, facilitating the prediction of solar power plant production. The proposed SolarPredictor model is novel in two respects. First, its architecture combines elements of UNet for spatio-temporal analysis with residual aggregation modules composing the prediction signal. Second, the training and augmentation regime utilizes weather forecasts to allow the model to better estimate real-world forecast errors and automatically adapt to specific power plant sites. Further, we presented the SolarDB dataset, covering a year of data for 16 power plants along with seven days of hourly weather forecast data. We make this dataset freely available to the research community, facilitating further research. The SolarDB dataset is used to evaluate the prediction performance of various models. The novel SolarPredictor architecture results in a 14.76% improvement against the second-best model. The proposed training regime is shown to effectively allow the model to gauge uncertainty in weather forecasts, resulting in an additional 28.27% improvement over the base model trained only on current weather. Notably, the presented results are consistent with real-world performance since weather forecast data is used during the evaluation. Finally, we present an empirical analysis of the factors determining the prediction difficulty, showing the importance of considering power plant location, seasonality, and quality of weather forecasts.

Similarly to other machine learning approaches, the proposed SolarPredictor model requires a considerable amount of training data for each power plant, slowing down its deployment. Our preliminary experiments indicate that transfer learning is highly viable. By training a baseline model on the SolarDB dataset and fine-tuning it to a target power plant, we are able to provide predictions with only several days of data instead of months. Our future work will focus on amending this issue, quantifying how well the models generalize between power plants.

Declaration of Competing Interest

The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The SolarDB dataset was created in collaboration with Forenas s.r.o., who provided access to the power plant data.

A. Dataset Details

For an overview of quantities contained within the SolarDB dataset, see Tab. 7. An random selection of Clear and Overcast production curves can be seen in Fig. 10.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>PowerAC</td>
<td>Final output after conversion</td>
<td>Meta</td>
<td>Freq</td>
<td>Frequency of power observation</td>
</tr>
<tr>
<td></td>
<td>EnergyDC</td>
<td>Energy produced over interval</td>
<td></td>
<td>Capacity</td>
<td>Installed capacity and number of inverters</td>
</tr>
<tr>
<td></td>
<td>PowerDC</td>
<td>Pure output of the panels</td>
<td></td>
<td>Location</td>
<td>Anonymized latitude and longitude</td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td>String summary of weather</td>
<td>Exogenous</td>
<td>Interpolated</td>
<td>Flag for original (0), interpolated (1)</td>
</tr>
<tr>
<td></td>
<td>PrecipInt</td>
<td>Precipitation intensity</td>
<td></td>
<td>Extrapolated</td>
<td>Flag for original (0), extrapolated (1)</td>
</tr>
<tr>
<td></td>
<td>PrecipProb</td>
<td>Precipitation probability</td>
<td></td>
<td>Day</td>
<td>Observation Day of the year</td>
</tr>
<tr>
<td>Temp</td>
<td></td>
<td>Measured temperature</td>
<td></td>
<td>Year</td>
<td>Year of observation</td>
</tr>
<tr>
<td></td>
<td>ApparentTemp</td>
<td>Perceived temperature</td>
<td></td>
<td>Time</td>
<td>Observation second of the day</td>
</tr>
<tr>
<td></td>
<td>DewPoint</td>
<td>Dew point temperature</td>
<td></td>
<td>Age</td>
<td>Age of forecast, 0 for measured</td>
</tr>
<tr>
<td>Humidity</td>
<td>Humidity from dry (0) to humid (1)</td>
<td></td>
<td></td>
<td>SunAltitude</td>
<td>Sun altitude from ground plane</td>
</tr>
<tr>
<td>Pressure</td>
<td></td>
<td>Pressure at ground level</td>
<td></td>
<td>SunAzimuth</td>
<td>Sun azimuth, north at 0, clockwise</td>
</tr>
<tr>
<td>WindSpeed</td>
<td>Average wind speed</td>
<td></td>
<td></td>
<td>SunIrradiance</td>
<td>Estimated clear sky irradiance</td>
</tr>
<tr>
<td>WindBearing</td>
<td>Wind bearing, north at 0°, clockwise</td>
<td></td>
<td></td>
<td>Status</td>
<td>Status code for power record</td>
</tr>
<tr>
<td>CloudCover</td>
<td>Cloud cover, clear (0) to overcast (1)</td>
<td></td>
<td></td>
<td>Error</td>
<td>Error code for power record</td>
</tr>
<tr>
<td>Visibility</td>
<td>Visibility up to 16 km</td>
<td></td>
<td></td>
<td>Clear</td>
<td>Daily clarity, clear (0) or overcast (1)</td>
</tr>
</tbody>
</table>

Table 7
(2-column) Dataset Quantities: Categorization of data included in the SolarDB dataset. Power production and exogenous variables are provided in 5-minute intervals for plants and inverters separately. Weather variables include on-site weather and weather forecasts, both at a 1-hour resolution. Meta-data is provided for each site and inverter.
Photovoltaic Power Forecasting using Weather Forecasts

Figure 10: (Color, 1-column) Day Clarity: Examples of clear (left) and overcast (right) productions from the SolarDB dataset. Each graph contains the power production (blue), temperature (orange), precipitation (green), wind speed (red), and humidity (purple). Generally, prediction for clear days is less complicated due to reduced weather-related effects.

B. Predictor Details

Cyclic Time features consist of two sub-features containing the sine and cosine components, calculated as

\[ v_x = 0.5 \cos \left( 2\pi \frac{v}{v_{\text{max}}} \right) + 0.5, \]

\[ v_y = 0.5 \sin \left( 2\pi \frac{v}{v_{\text{max}}} \right) + 0.5, \]

where \( v \) is the input value and \( v_{\text{max}} \) is the maximum value. The resulting values, along with the Age and Sun features, facilitate the model’s phase detection.

Sample weights include the clear-day weights \( w_{g_{\text{cd}}} \), which balance the data based on day clarity (Sec. 2.1.3), setting \( w_{g_{\text{cd}}} = \max \left[ 0.5, \min \left[ 1.0, 2.0 - L_{OC} \right] \right] \in (0.5, 1.0) \), where \( L_{OC} \) is the clear-day likelihood defined in Eq. 2. The temporal weights \( w_{g_{\text{t}}} \) favor samples based on their frequency

\[ w_{g_{\text{t}}}(dt) = \begin{cases} 
\lambda_d & \text{if } dt \text{ is the beginning of the day} \\
\lambda_h \cdot \frac{1}{24} & \text{if } dt \text{ is the beginning of the hour} \\
\lambda_s \cdot \frac{1}{288} & \text{otherwise},
\end{cases} \]

where \( dt \) is the sample’s timestamp, \( \lambda_d = 1, \lambda_h = 12, \lambda_s = 32 \) are the importance multipliers, and \( 1/24, 1/288 \) are the relative frequencies for hourly and 5-minute samples, respectively.

Weather Sampling is used during the training to gradually teach the model the veracity of the input weather features. The three types of sampling schemes can be found in Tab. 8.

<table>
<thead>
<tr>
<th>Power</th>
<th>( t )</th>
<th>( t + 1 )</th>
<th>...</th>
<th>( t + o )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured</td>
<td>( t )</td>
<td>( t+1 )</td>
<td>( t+2 )</td>
<td>...</td>
</tr>
<tr>
<td>Forecast</td>
<td>( t-d )</td>
<td>( t )</td>
<td>( t+1 )</td>
<td>( t+2 )</td>
</tr>
<tr>
<td>Realistic</td>
<td>( t )</td>
<td>( t+1 )</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 8:

(1-column) Weather Sampling: The sampling scheme used for weather feature selection. Target power values at a given time \( t + x \) are paired with weather features \( t+ \) \( a \) from source time \( t + s \) for a target time \( t + g \), imposing age \( a = g - s \). Measured weather provides zero-error features, Forecast uses a fixed age, and Realistic increases the age linearly with time.
Loss Function used during the training is a combination of Mean Squared Error $L_M$, Huber Loss $L_H$, and Differential Smooth Loss $L_\Delta$. The Smooth Loss $L$ is defined as

$$L_M(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$  

(8)

$$L_H(y, \hat{y}) = \begin{cases} \frac{1}{2} (y_i - \hat{y}_i)^2, & \text{if } |y_i - \hat{y}_i| \leq \delta \\ \frac{1}{2} \delta^2 + \delta (|y_i - \hat{y}_i|-\delta), & \text{otherwise,} \end{cases}$$  

(9)

$$L_\Delta(y, \hat{y}) = \sum_{i=1}^{n-1} |\hat{y}_{i+1} - \hat{y}_i|,$$  

(10)

$$L(y, \hat{y}) = \omega g(y) \cdot (\lambda_M L_M(y, \hat{y}) + \lambda_H L_H(y, \hat{y}) + \lambda_\Delta L_\Delta(y, \hat{y})),$$  

(11)

with $\omega g$ being the sample weight, experimentally choosing $\delta = 1$, $\lambda_M = 0.5$, $\lambda_H = 0.7$, and $\lambda_\Delta = 0.5$.

Predictor Architecture, visualized in Fig. 2, was implemented using the TensorFlow framework [76]. The AMS-Grad [73] variant of the ADAM [72] optimizer is used in training. The Initial Predictor Architecture initialization is used for the kernel and bias, respectively. Additionally, kernel regularization is used, setting for $dilated$ 1D convolutions [53] using the LeakyReLU ($\alpha = 0.3$) activation function. Glorot normal and uniform [80] initialization is used for the kernel and bias, respectively. Additionally, kernel regularization is used, setting $l_2 = 0.005$.

The architectural details for the diagram visualized in Fig. 2 are as follows:

- $\downarrow$ ResBlock [ F ]: In $\rightarrow$ Conv1D[F, 3] $\rightarrow$ BatchNormalization $\rightarrow$ Conv1D[F, 3] $\rightarrow$ BatchNormalization $\rightarrow$ AveragePooling1D[s = 2] $\rightarrow$ Add[ In $\rightarrow$ Conv1D[F, 1, s = 2] $\rightarrow$ BatchNormalization ] $\rightarrow$ Out

- ResBlock [ F ]: In $\rightarrow$ Conv1D[F, 3] $\rightarrow$ BatchNormalization $\rightarrow$ Conv1D[F, 3] $\rightarrow$ BatchNormalization $\rightarrow$ Add[ In $\rightarrow$ Conv1D[F, 1] $\rightarrow$ BatchNormalization ] $\rightarrow$ DropoutOut

- $\uparrow$ ResBlock [ F ]: In $\rightarrow$ UpConv1D[s = 2] $\rightarrow$ Conv1D[F, 3] $\rightarrow$ BatchNormalization $\rightarrow$ Concatenate[ Skip ] $\rightarrow$ Conv1D[F, 3] $\rightarrow$ BatchNormalizationConv1D[F, 3] $\rightarrow$ BatchNormalization $\rightarrow$ Add[ In $\rightarrow$ Conv1DTranspose[F, 1, s = 2] $\rightarrow$ BatchNormalization ] $\rightarrow$ Out

- ResAgg: In $\rightarrow$ UpDownConv1D[288] $\rightarrow$ Conv1D[1, 1] $\rightarrow$ Add[ In ] $\rightarrow$ Out


- SolarPredictor: Input $\rightarrow$ RComposer[ RUNet ] $\rightarrow$ Output

C. Evaluation

Importance Analysis uses the normalized Gini Importance [75] as the metric, defined as

$$i_f = 1/n_{x}^{\text{root}} \sum (n_s^p n_s^p - n_s^l n_s^l - n_s^r n_s^r)$$  

(12)

for each feature $f$, where $n_s^x$ is the number of samples reaching node $x$ and $n_l^x$ is the Gini Impurity for node $x$. Furthermore, root represents the root node, and $p$, $l$, $r$ are the parent, left child, and right child nodes. Finally, the $i_f$ values are calculated for each ensemble and aggregated through their mean $I_f$ and variance $\sigma_f^2$ values. However, as noted by other authors [81, 82], Gini importance may lead to biased results. Thus, we confirm our results by computing the Permutation Importance [81], leading to the same relative ordering.
Quantitative Metrics used during evaluation include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Relative RMSE (RRMSE), and Coefficient of Determination ($R^2$), defined as

$$\text{MSE} = \frac{1}{n} \sum_i (y_i - \hat{y}_i)^2,$$

$$\text{RMSE} = \sqrt{\text{MSE}},$$

$$\text{RRMSE} = \sqrt{\frac{\text{MSE}}{\sum_i y_i^2}},$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2},$$

where $y_i, \hat{y}_i$ are the ground-truth and predicted values, respectively. Hourly aggregates of weather (WError) and power (PError) are defined as $(\sum_i \hat{y}_i \hat{f}) / (\sum_i y_i f)$, where $f$ and $\hat{f}$ represent the measurement frequency. Unless otherwise specified, the measurement frequency for both values is $f = \hat{f} = 5/60$ – i.e., 5-minute intervals. Data correlations are examined using Pearson ($cor_p$) [74] and Spearman ($cor_s$) [83] correlation coefficients.

References


Photovoltaic Power Forecasting using Weather Forecasts


