A TCN-based Spatial-Temporal PV Forecasting Framework with Automated Detector Network Selection

Yiyan Li, Member, IEEE, Lidong Song, Student Member, IEEE, Si Zhang, Student Member, IEEE, Laura Kraus, Taylor Adcox, Roger Willardson, Abhishek Komandur, and Ning Lu, Fellow, IEEE

Abstract—This paper proposes a two-stage PV forecasting framework for MW-level PV farms based on Temporal Convolutional Network (TCN). In the day-ahead stage, inverter-level physics-based model is built to convert Numerical Weather Prediction (NWP) to hourly power forecasts. TCN works as the NWP blender to merge different NWP sources to improve the forecasting accuracy. In the real-time stage, TCN can leverage the spatial-temporal correlations between the target site and its neighbors to achieve intra-hour power forecasts. A scenario-based correlation analysis method is proposed to automatically identify the most contributive neighbors. Simulation results based on 95 PV farms in North Carolina demonstrate the accuracy and efficiency of the proposed method.

Index Terms—Neighbor selection, NWP blending, physics-based model, spatial-temporal PV forecasting, temporal convolutional network.

NOMENCLATURE

Scalar

| Symbol | Description |
|--------|-------------|
| $d$    | Dilation rate |
| $D$    | Number of days in the historical data |
| $E_{bias}$ | Bias in the physics-based model |
| $h$    | Index of the historical days |
| $K$    | Filter size |
| $M$    | Length of the daily irradiance profile |
| $N$    | Number of PV sites |
| $N_{stack}$ | Number of module stacks in the TCN model |
| $P_{real}$ | Actual power output from field measurements |
| $P_{lima}$ | Power output of the physics-based model at time $t$ |
| $P_{cc}$ | Pearson Correlation Coefficient |
| $P_{cc,max}$ | $P_{cc}$ value with optimal time shift, $\Delta t_{max}$ |
| $R_{field}$ | Receptive field of the TCN model |
| $S_h$ | Index of the correlation scenarios on $h^{th}$ day |
| $\Delta t$ | Time shift between two time series |
| $\Delta t_{max}$ | Optimal time shift that leads to $P_{cc,max}$ |
| $T_{shift}$ | Threshold of time-lagged correlation analysis |
| $T_{theta}$ | Threshold of $\Delta t_{max}$ to determine successful detection |
| $\Delta x$ | Irradiance change threshold to detect cloud event |
| $\phi$ | Successful detection rate |
| $\phi_{max}$ | Maximum successful detection rate |

Function/Matrix

| Symbol | Description |
|--------|-------------|
| $F$ | Filter in TCN, $F = [f_0, f_1, \ldots, f_K]$ |
| $X$ | Normalized irradiance vector, $X = [x^1, x^2, \ldots, x^M]$ |
| $X_T$ | X of the target site, $X_T = [x^1_T, x^2_T, \ldots, x^M_T]$ |
| $X_D$ | X of the detector site, $X_D = [x^1_D, x^2_D, \ldots, x^M_D]$ |
| $\mathbf{X}_T$ | Differential vector of $X_T$, $\mathbf{X}_T = \left[ \hat{x}^1_T, \hat{x}^2_T, \ldots, \hat{x}^M_T \right]$ |
| $\mathbf{X}_D$ | Differential vector of $X_D$, $\mathbf{X}_D = \left[ \hat{x}^1_D, \hat{x}^2_D, \ldots, \hat{x}^M_D \right]$ |
| $\mathcal{F}$ | Vector of the neighboring sites |
| $\mathcal{F}_{opt}$ | Vector of the selected optimal detector network |
| $\mathcal{T}$ | Matrix of $\Delta t_{max}, Dx(N-1)$ |
| $\Phi$ | Matrix of $P_{cc,max}$ |

1. INTRODUCTION

PV forecasting methods can be categorized into two main classes: physics-based model and data-driven model [1]-[3]. The physics-based model converts the irradiance forecasts (usually obtained from NWP, satellite images, total sky imagers, etc.) to power forecasts. As a result, once the physics-based model is well calibrated, its forecasting performance completely relies on the irradiance data source. Meanwhile, because the physics-based model requires detailed parameters for the PV module/inverter and considerable model maintenance efforts, it is usually built for MW-level PV farm and rarely applied to the residential rooftop PV systems. Data-driven models such as statistical models and machine learning models, are more flexible and can be built for any PV site with enough historical data. The underlying assumption of the data-driven model is that the power output in the future will follow similar patterns as the past. However, when weather conditions change rapidly, such assumption may only hold for a short
time horizon. A qualitative comparison between the physics-based model and data-driven model is given in Table I.

| Physical parameter requirement | NWP requirement | Model adaptability | Forecasting granularity | Effective forecasting horizon |
|-------------------------------|-----------------|---------------------|-------------------------|-------------------------------|
| Historical data requirement  | Low             | Low                 | Same as NWP             | Days-ahead                    |
| Data-driven model             | High            | High                | Same as historical data | Hours-ahead                   |

In recent years, as an emerging branch of data-driven methods, leveraging the spatial-temporal correlations among neighbor sites to improve the short-term forecasting performance has drawn increasing attention. By analyzing the power outputs of a group of PV sites or the field measurements from adjacent meteorological stations, the cloud movement patterns can be modeled implicitly or explicitly to further improve the forecasting accuracy. Compared with the forecasting methods using total sky imagers or satellite images, spatial-temporal forecasting methods are usually cheaper and easier to implement while achieving comparable performance without requiring extensive hardware or external data supports.

There are two major ways of leveraging the spatial-temporal correlations: explicit modeling and implicit inferencing. The explicit modeling tries to recover the cloud condition explicitly, such as the cloud movement vector, cloud outlines, etc. In [4], Chen et al. design a two-layer concentric sensor network around the target PV site to catch the cloud movement. In [5], Bosch et al. introduce an analytical method to estimate the cloud movement vector based on the irradiance sensor measurements or inverter-level power outputs within the same PV farm. Based on the sky images, in [6], Zhen et al. propose a cloud motion speed calculation method that combines pattern classification and particle swarm optimization to assist minute-level PV forecasting. In [7], Meng et al. try to recover the “virtual cloud” based on the inverter-level power output measurements within a PV farm. Based on the virtual cloud information, the second-level forecasting can be achieved with satisfying accuracy. Once the cloud is explicitly modeled, its trajectory can be accurately predicted in the near future, leading to a significant improvement in the ultra-short term PV forecasting accuracy. However, to recover the cloud condition, a dense sensor network is usually required, making the explicit modeling method less adaptive. Meanwhile, the forecasting horizon is usually limited to minutes or up to 1 hour ahead.

The implicit inferencing uses statistical or machine learning tools to extract the spatial-temporal information among neighbors to assist forecasting. Regression models, such as AR, ARX, NARX, are firstly introduced in [8]-[11] to build the mapping from the neighboring sites to the target site. In [12], Yang et al. achieve spatial-temporal PV forecasting using time-forward kriging. Compressive sensing is used by Tascikaraoglu et al. in [13] to extract the spatial-temporal correlations between a target meteorological station and its surrounding stations to improve the short-term forecasting accuracy. In [14], Zhang et al. use Bayesian network to forecast PV generation based on spatial-temporal analysis results. Recently, deep-learning receives increasing attention because of its powerful nonlinear learning ability. In [15], Liu et al. combine CNN and GRU to extract the spatial-temporal information from multi-dimensional time series inputs from adjacent sites. Similarly, in [16], Wang et al. implement CNN to extract the spatial features across multiple sites, and then use LSTM to achieve the forecasting of the target site. In [17], Jeong et al. organize the multi-site time series data into spatial-temporal matrix, and then use CNN to achieve feature extraction and forecasting. In [18], Venugopal et al. compare different CNN structures in extracting features from heterogeneous data sources, as well as their short-term PV forecasting performances. Using implicit inferencing method is convenient and has a lower entry level, because the spatial-temporal information is assumed to be learned by machine learning models instead of requiring domain expertise knowledge. However, the results are usually not interpretable and the algorithm may be instable if the model is not properly trained, especially for deep-learning models.

There are two research gaps that have not yet been fully addressed in the existing spatial-temporal forecasting literatures.

First, there lacks of an automated, objective detector site selection mechanism. Selecting the most contributive neighbors (also called detectors in this paper) for the target site among all its neighbors is important in both improving the forecasting performance and reducing the model complexity. Conventional pairwise-correlation-based methods ignore both the temporal correlations between sites (i.e. which site engages the cloud event first and which next) and the collaboration effect of multiple neighbors. Meanwhile, conventional methods usually require a manually-defined correlation threshold to eliminate “unrelated” neighbors. This is highly subjective for guaranteeing selection optimality.

Second, there lacks of a computationally efficient deep-learning model. The combination of CNN (for spatial feature extraction) and RNN (or LSTM, GRU, etc. for temporal learning and forecasting) has become a widely-used methodology for solving the spatial-temporal PV forecasting problem. However, due to the sequential nature of RNN, its training process cannot be paralleled and is time-consuming. This makes the real-time model tuning impractical.

In this paper, a two-stage short-term PV forecasting framework is proposed for MW-level PV farms. In the day-ahead stage, the inverter-level physics-based model will take in the NWP blending results from TCN to obtain the days-ahead forecast with hourly granularity. In the real-time stage, TCN will learn from the historical data from the selected neighbors to generate hours-ahead forecasting with minute-level granularity. Contributions of this paper are two-folds.

1) We introduce TCN, a novel and efficient deep-learning model, for solving the spatial-temporal PV forecasting problem. In the day-ahead stage, TCN works as an NWP blender to improve the physics-based model performance; in the real-time stage, TCN can efficiently learn the spatial-temporal features from neighbor sites and achieve forecasting. Because TCN has
a purely convolutional structure and can be highly paralleled, it has better computational efficiency than the CNN-LSTM structure. To the best of the authors’ knowledge, this is the first time for TCN to be used for solving the spatial-temporal PV forecasting problem.

2) A scenario-based neighbor selection algorithm is novelly proposed to identify the most contributive neighbors for the target site. Compared with conventional methods, the proposed algorithm considers the temporal leading/lagging patterns between sites and the collaboration effect of multiple neighbors. Meanwhile, the proposed algorithm is fully automated and objective. It does not require any domain expertise and human intervention.

The rest of the paper are organized as follows: Section II introduces the proposed two-stage PV forecasting framework. Section III demonstrate the simulation results. Section IV concludes this paper.

II. METHODOLOGY

The proposed two-stage PV forecasting framework is summarized in Fig.1. In the day-ahead stage, the hourly NWP data from different sources will be fed into the TCN model. TCN will work as the NWP blender to find an optimal combination of NWPs that can maximize the irradiance prediction accuracy. Then the predicted irradiance from TCN will be converted to hourly PV forecasts by the physics-based model to achieve one to seven days ahead forecasting. In the real-time stage, the historical irradiance data from multiple PV sites will be normalized between 0 and 1. At each time step, the irradiance data of all PV sites will be coded into a 2D matrix based on their geographical location. To select the most-correlated neighbors, in this paper we propose a scenario-based neighbor selection algorithm that can automatically identify an effective detector network for the target site without requiring human assistance. Finally, the historical data of the target site together with its selected neighbors will be fed into the TCN model to extract the spatial-temporal information and achieve intra-hour forecasting. Such a two-stage forecasting framework combines the advantages of physics-based model and deep-learning model to provide full range forecasting supports for PV farm operation.

The two key algorithms, i.e., the TCN model and the scenario-based neighbor selection algorithm are detailed in Section II.A and II.B respectively.

A. Temporal Convolutional Network

TCN is a fully convolutional-based network structure [19], as shown in Fig. 2. Each convolutional layer needs to have the same length as the input layer. To meet this requirement, zero padding is applied to solve the dimension reduction issue caused by the convolution operation (see the dashed blocks in Fig. 2). After a few convolutional layers, the features of the input time series are extracted and compressed into the output layer, which can be further used for forecasting purpose.

Compared with sequential networks such as RNN and LSTM, such a purely convolutional structure of TCN can be highly paralleled in model training and therefore has better training efficiency [19]. Until now, TCN has been successfully implemented in solving time series forecasting problems, such as PM2.5 forecasting [20], load forecasting [21], etc. however, it has not been addressed in the spatial-temporal PV forecasting problem.
The most significant feature of TCN is the \textit{dilated convolution}. Assume the input time series is \( X = [x^1, x^2, \ldots, x^M] \), and we use a filter \( F = [f_0, f_1, \ldots, f_K] \) to conduct convolution. Then, the dilated convolution \( G(\cdot) \) for the element \( x^m \) in \( X \) can be calculated by
\[
G(x^m) = \sum_{i=0}^{K-1} f_i \cdot x^{m-d_i}
\]
where \( d \) is the dilation rate, and \( m-d_i \) indexes to the past historical data before \( x^m \). When \( d = 1 \), (1) reduces to a regular convolution operation (e.g. the first layer in Fig. 2). When \( d \) is larger than 1 (e.g. the second and third layer in Fig. 2), the filter will skip \((d-1)d\) of the elements in the previous hidden layer and only focus on the remaining 1/d. In this way, the whole network will have a large \textit{receptive field} that increases exponentially with the number of layers with limited model complexity. The receptive field of TCN model can be calculated by
\[
R_{\text{field}} = 1 + 2 \cdot (K_{\text{size}} - 1) \cdot N_{\text{stack}} \cdot \sum_i d_i
\]
According to (2), we have 3 ways to increase the receptive field: using larger filter size, larger dilation rate, and increase the network depth.

\textit{Causal convolution} is another feature of the TCN model. As shown in Fig.2, each convolution layer only extracts information from the past historical data. In other words, there is no “information leakage” from the future. This structure is particularly suitable in solving forecasting problems where the future information is unavailable. To further improve the model performance, residual connections [22] can be used to achieve identical mappings.

\subsection*{B. Scenario-based Neighbor Selection Algorithm}

Assume the normalized daily irradiance profile for the target site (i.e. the PV site we want to forecast) is \( X_T = [x^1_T, x^2_T, \ldots, x^M_T] \), and for one detector site is \( X_D = [x^1_D, x^2_D, \ldots, x^M_D] \), as shown in Fig.3(a). In this paper, our irradiance data is in 5-minute granularity, so we have \( M = 288 \) data points in each day. In the short-term PV forecasting problem, the power drop caused by the cloud movement is the most difficult part to forecast and will lead to significant impact to the system operation. So our analysis will mainly focus on the cloud events instead of the whole irradiance time series.

We define the cloud event as the irradiance drop greater than threshold \( \Delta x \) during two consecutive time intervals
\[
x_i - x_{i-1} \leq -\Delta x
\]
Note that \( \Delta x \) is dependent on data granularity. In this paper, we define \( \Delta x = 0.3 \), because we want to capture relatively large cloud events that cause the irradiance to drop greater than 30% of the rated power within 5 minutes. To detect the cloud event, we conduct differential operation to \( X_T \) and \( X_D \), and extract the cloud event for the target site and the detector site, \( \hat{X}_T \) and \( \hat{X}_D \), respectively, so we have
\[
\hat{X}_T = [\hat{x}^1_T, \hat{x}^2_T, \ldots, \hat{x}^M_T] \\
\hat{X}_D = [\hat{x}^1_D, \hat{x}^2_D, \ldots, \hat{x}^M_D]
\]
\[
\hat{x}^i_T = 0, \quad \text{if } \hat{x}^i_T > -\Delta x \\
\hat{x}^i_D = \text{if } \hat{x}^i_T \leq -\Delta x, \quad \forall i \in [1, 2, \ldots, M - 1]
\]
Examples of the extracted cloud event series are shown as the solid lines in Fig. 3(b) and 3(c).

Further, we conduct the time-lagged correlation analysis [23] based on the extract cloud event series. More specifically, we make time shifts to the target series to find the optimal time shift \( \Delta_{\text{max}} \) where the Pearson Correlation Coefficient [24] \( P_{cc} \) between the target and the detector series is maximized.
\[
\Delta_{\text{max}} = \arg \max_{\Delta_t} P_{cc}(\hat{X}_T|\Delta_t : M + \Delta_t - 1), \hat{X}_D)
\]
The indexed values that out of range \([1; M - 1]\) will be padded by 0. Meanwhile, we set a threshold \( T_{\text{shift}} \) for the time shift to guarantee the correlation is physically meaningful. An example of \( X_T \) with \( \Delta_{\text{max}} \) time shift is shown in Fig. 3(c) as dotted lines.

After calculating the \( P_{cc} \) between the target site and detector sites, a commonly-used method is to select detectors with the highest \( P_{cc} \) values. This is straightforward as a high \( P_{cc} \) value indicates more significant correlation and therefore may contribute to the forecasting accuracy of the target site. However, we notice that such \( P_{cc} \) based selection method has the following disadvantages:

\textbf{1) Ignore the temporal correlation pattern that is crucial to detect the cloud events.} Only \textit{leading correlation} where the cloud passes the detector site earlier than the target site (\( \Delta_{\text{max}} > 0 \)) contribute to the target site forecasting. A detector site having a high \( P_{cc} \) but with \textit{lagging correlation} pattern (\( \Delta_{\text{max}} \leq 0 \)) cannot foresee the upcoming cloud event on the target site and therefore is not contributive to the target site forecasting.
2) $P_{cc}$ threshold selection is subjective and empirical. Since different neighbor sites have different $P_{cc}$ with the target site, a $P_{cc}$ threshold is needed to eliminate low-correlated neighbors. Because $P_{cc}$ values vary significantly for different cases, it is difficult to determine the optimal threshold and the number or neighbors.

3) Collaborations among neighbors are not considered.

The neighboring sites are usually selected in a pair-by-pair manner based on $P_{cc}$. However, their synergetic performance as a detector network is not considered.

In this paper, we propose a scenario-based neighbor selection algorithm to solve the aforementioned issues. First, all the 6 possible correlation scenarios between the target and the neighbor sites are summarized in Table II, based on their daily cloud conditions and $\Delta t_{max}$. If there is at least one cloud event during the day, this day is defined cloudy. Otherwise, this day is defined as clear sky.

| Scenario No. | Detector site | Target site | $\Delta t$ | Definition |
|--------------|---------------|-------------|------------|------------|
| 1            | Clear sky     | Clear sky   | \(\emptyset\) | Ignored    |
| 2            | Cloudy        | Clear sky   | \(\emptyset\) | Wrongly detect |
| 3            | Clear sky     | Cloudy     | \(0 < \Delta t_{max} \leq T_{thr}\) | Successful detection |
| 4            | Cloudy        | Cloudy     | \(T_{thr} < \Delta t_{max}\) | Irrelevant |

Scenario 1 will not be considered in this study, because our focus is to detect the cloud events that will significantly impact the PV output. Scenario 5 is defined as the successful detection where the cloud events occur earlier in the detector site than in the target site with a leading time $\Delta t_{max} \in (0, T_{thr})$. Based on Table II, we further define the successful detection rate for a detector site as

$$
\phi = \frac{\sum_{k=1}^{n} \mathcal{I}^{[2,3,4,5,6]}(S_k)}{\sum_{k=1}^{n} \mathcal{I}^{[2,3,4,5,6]}(S_k)}
$$

(7)

where $\mathcal{I}_A(x)$ is the indicator function that

$$
\mathcal{I}_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}
$$

For a given neighbor-target site pair, $\phi \in [0, 1]$ measures the successful detection rate. A larger $\phi$ means the neighbor site has a statistically significant leading correlation pattern with the target site, therefore has a better chance to provide future cloud information for the target site to assist forecasting. In real world, this means the wind direction has a higher chance to blow from the neighbor site to the target site. This reflects localized weather patterns dominated by geographical characteristics.

To maximize $\phi$, a detector network containing multiple neighbor sites are needed to have a better chance to forecast the upcoming cloud events for the target site. When calculating $\phi$ of a detector network, for each historical day we select the neighbor with the largest $\Delta t_{max}$ as a representative because it has the best prediction ability. However, selecting an optimal subset among all candidate neighbors is a typical NP-hard problem. Therefore, in this paper we design a greedy-searching algorithm to find a near-optimal solution. Key steps of the algorithm are summarized as follows:

1) **Data preparation**: calculate the maximum time-lagged correlation coefficients, $P_{cc,max}$, and the corresponding $\Delta t_{max}$ between each candidate neighbor site and the target site based on historical cloud event data.

2) **Detector network formulation**: add each neighbor site successively to the detector network according to a descending order of their yearly averaged $P_{cc,max}$ values. Every time we introduce a new neighbor, we will calculate $\phi$ of the detector network. In this way, after going through all the potential neighbors, we will have a curve of $\phi$ for different number of neighbor sites. The detector network that has the maximum $\phi$ value is selected for further refinement, as shown in step 3).

3) **Detector network refinement**: In this step, we try to remove “bad” neighbors that introduced by the greedy-search algorithm in step 2). More specifically, we will successively remove the neighbors in the detector network obtained from 2) to see if $\phi$ can be further improved. If $\phi$ is improved after removing certain “bad” neighbors, we will remove those and go back to step 2). Otherwise if $\phi$ cannot be further improved in this step, we consider the detector network is converged and end the algorithm.

The pseudocode of the algorithm is shown in Algorithm 1. After selecting the detector network, we will further put the historical data of both the target site and the detector network into the TCN model to extract their spatial-temporal correlations. This will be discussed in Section III.B.

**Algorithm 1**: Scenario-based detector site selection algorithm

**Input**: Target site $i$, neighboring sites $\mathcal{J}_{nb(i)}$, cloud event sets of each site

**Output**: Selected detector network $\mathcal{F}_{opt}$ for the target site

**Initialization**: $\phi_{\text{init}}=0$, $\mathcal{F}=[]$, $T_{\text{thr}}=0$, $P_{\text{init}}=0$, flag=1

# step 1): data preparation
1: for $d=1, 2, \ldots, D$ do
2:   for $j=1, 2, \ldots, N$ and $j \neq i$ do
3:     calculate $\mathcal{T}(d,j)=\Delta t_{max} \text{ between site } i \text{ and } j \text{ according to (4)}$
4:     calculate $P(d,j)=P_{cc,max}$
5: end

# step 2): detector network formulation
6: sort $\mathcal{T}$ in descending order according to the average values of $P$
7: while(flag=0) do
8:   flag = 0
9:   for $k=1, 2, \ldots, N-1$ do
10:      add site $\mathcal{J}[k]$ to $\mathcal{F}$
11:      calculate $\phi$ of $\mathcal{F}$
12:      if $\phi > \phi_{\text{max}}$ do
13:         $\phi_{\text{max}} = \phi$, $\mathcal{F}_{\text{opt}} = \mathcal{F}$
14:      end
15: end

# step 3): detector network refinement
16: for $k=1, 2, \ldots, \text{length( } \mathcal{F}_{\text{opt}} \text{ )}$ do
17:   calculate $\phi$ of $\mathcal{F}_{\text{opt}}$ without $\mathcal{J}[k]$
18:   if $\phi > \phi_{\text{max}}$ do
19:      remove $\mathcal{J}[k]$ from $\mathcal{F}_{\text{opt}}$
20:      $\phi_{\text{max}} = \phi$
21:      flag = 1

III. CASE STUDY

A. Test Case Setup and Data Preprocessing

In this paper, field measurement data collected from 95 utility-scale PV farms in North Carolina are used to develop and verify the performance of the proposed algorithm. The PV farms range from 0.4MW up to 26.2MW. The locations of the
PV sites are shown in Fig. 4. The available measurements of each site are summarized in Table III. All the measurements are from 1/1/2020 to 11/30/2020 with 5-minute granularity.

The parameters of PV modules and inverters from the 95 PV farms are used to build the physics-based models. An example of the parameters from PV site Lenoir is shown in Table IV and Table V.

For each PV site, because multiple measurements are available, we select the irradiance data measured by the pyranometer as the representative to reflect the cloud movement. This is because the pyranometer is placed horizontally and is more comparable across different PV sites [25]. However, there are missing points in pyranometer measurements (the data completeness is about 87%). In this paper, we use the correlations between pyranometer measurements and other irradiance-related measurements (e.g. inverter-level power output) to patch the missing data, as shown in Fig. 5. All the measurements are normalized by their maximum value.

### B. Day-ahead Stage

In this stage, we build the inverter-level physics-based model for each PV site to achieve the irradiance-power conversion, based on the public PV modeling toolkit PVlib [26]. Here we take 4 inverters: Lenoir-B2, Lenoir2-C1, SnowHill2-A3 and Crockett-B1, to show the performance of the physics-based model.

### Table IV

| Parameters | Values | Parameters | Values |
|------------|--------|------------|--------|
| $STC$      | 250 (W) | $P_{B2}$  | 0.0057 (W/V) |
| $PTC$      | 230 (W) | $P_{C1}$  | 0.00074 (W/V) |
| $P_{acc}$  | 511501 (W) | $P_{max}$ | 101 (W) |
| $V_{dc}$   | 600 (V) | $I_{dc}$  | 1600 (A) |
| $I_{ref}$  | 330 (V) | $C_{dc}$  | 330 (V) |
| $C_{max}$  | 3.11x10^3 (W/V) | $MPPT_{sca}$ | 330 (V) |

Inverter type: SMA America SC 500HE-US

After the physics-based models are built, we firstly validate their irradiance-power conversion accuracy by taking the field measurement irradiance as the input to see if it can produce the correct power output. The average model bias is calculated by

$$E_{bias} = \frac{1}{M} \sum_{i=1}^{M} (P_{sim} - P_{real})$$

One month historical data in July 2020 [27] are used to validate the model performance. As shown in Fig. 8 and Table VI, we can see that the simulation results follows closely to the ground truth with the average bias less than 1%. This means the physics-based models are relatively accurate and can be used to produce forecasting results when fed with NWP data.

### Table V

| Parameters | Values | Parameters | Values |
|------------|--------|------------|--------|
| $V_{dc}$   | 200 (V) | $C_{2}$    | 0.0057 (W/V) |
| $P_{acc}$  | 1879 (W) | $C_{3}$    | 0.00074 (W/V) |
| $P_{max}$  | 500000 (W) | $P_{max}$ | 101 (W) |
| $V_{dc}$   | 511501 (W) | $V_{dc}$  | 600 (V) |
| $I_{dc}$   | 371 (V) | $I_{dc}$  | 1600 (A) |
| $C_{dc}$   | -4.03x10^4 (W/V) | $C_{max}$ | 330 (V) |
| $C_{max}$  | 3.11x10^3 (W/V) | $MPPT_{sca}$ | 330 (V) |

Inverter type: SMA America SC 500HE-US

To test the forecasting performance of the physics-based model, we recorded the NWP data from May to October in 2020.
2020, and fed them into the physics-based model to generate the forecasts. The NWP data comes from 5 difference sources: High-Resolution Rapid Refresh (HRRR), Global Forecast System (GFS), National Digital Forecast Database (NDFD), Rapid Refresh (RAP) and North American Mesoscale (NAM). The features and the forecasting performance of each NWP data source are summarized in Table VII.

![Table VII](image)

We can see that different NWP data sources have different forecasting features and performances. This inspires us to combine them together to offset their modeling deficiencies and improve the overall forecasting performance. To achieve this, we select TCN as the NWP blender: the inputs are time series data from different NWP data sources, and the output is the time series of the field measurement irradiance. In short, TCN here achieves the sequence-to-sequence mapping from NWPs to the actual irradiance. Finally, the output irradiance of TCN will be fed into the physics-based model to obtain the final power forecasting results.

![Table VIII](image)

C. Real-time stage

In the real-time stage, we use the proposed TCN model to extract the spatial-temporal information between the target site and its neighbors to enhance the forecasting accuracy of the target site. The cloud event detection threshold is $\Delta t = 0.3$, and time-lagging threshold is $\Delta T = 1$h (for 1h ahead forecasting).

Firstly, we select two sites: Lenoir and Marshville, to demonstrate the proposed scenario-based neighbor selection algorithm, as shown in Fig. 8. For the site Lenoir, the successful detection rate $\phi$ firstly increases with the number of selected neighbors, and reach its maximum $\phi_{\text{max}} = 0.65$ when 7 neighbors are selected. This is because more neighbors will have a higher chance to catch the upcoming cloud event for Lenoir. However, as the number of the selected neighbors further increases, $\phi$ starts to drop because more irrelevant cloud events (i.e. scenario No.6 in Table II) start to be detected, which will pollute the detection accuracy. Similar results are observed for Marshville. In this way, we can automatically identify the optimal detector network that has the best chance to detect the upcoming cloud events for the target site automatically without human intervention.

![Table IX](image)

We use the historical data from May to September with hourly granularity to train the TCN model (for the GFS data with 3-hour granularity, we firstly up-sample them to hourly granularity by interpolation), and test its performance on October. The TCN model configuration is shown in Table VIII. The performance comparison between TCN and other benchmarking blending methods: Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR), Multilayer Perceptron (MLP), LSTM, are summarized in Table IX. After TCN blending, the forecasting bias reduces to 0.47, and the forecasting RMSE reduces to 43.17, both are significantly improved compared with any single NWP model. TCN also outperforms other benchmarking methods. The forecasting results of Lenoir-B2 inverter from Oct.1 to Oct.7 are shown in Fig. 7 as an example. After developing the physics-based models for all the inverters, site-level forecasting results can be obtained.

![Fig. 7](image)

![Fig. 8](image)
configurations of the TCN and CNN-LSTM models are shown in Table X. We keep them with similar complexity to make the results comparable.

| TABLE X | MODEL CONFIGURATIONS |
|---------|-----------------------|
| TCN     | CNN-LSTM              |
| Kernel size | 3 | 3 |
| Input data length | 24h | 24h |
| Number of filters | 64 | 64 |
| Dilation rate | [1, 3, 9] | Number of units | 96 |
| Number of stacks | 1 |  |
| Skip connection | Yes |  |
| Total parameter | 67K | 64K |

To train the model, for each PV site we split the 1 year historical data into training (70%), validation (20%), and testing (10%). We train both TCN and CNN-LSTM on the training set until they obtain the best performance on the validation set, and then test them on the testing set. The training curves of the two models on site Lenior is shown in Fig. 9 as an example. We can see that TCN has better convergence than CNN-LSTM in both training and validation. Meanwhile, the training of TCN takes about 2 minutes on Intel i9-9900K CPU with 64GB RAM, which is also considerably faster than the 8 minutes of CNN-LSTM. This means the TCN model has better learning performance and training efficiency compared with those of CNN-LSTM under the same model complexity.

![Fig. 9. Training and validation curves of TCN and CNN-LSTM.](image)

The forecasting result statistics of TCN and CNN-LSTM on all the 95 PV sites are shown in Fig. 10. In order to validate the effectiveness of the neighbor selection algorithm, we set up 4 different scenarios: 1) Selected neighbor: the two models are trained based on the historical data of the selected neighbors. This is our target scenario. 2) Single site: the two models are trained solely based on their own historical data, so that no neighbor information is used. 3) All sites: all the historical data of the 95 PV sites are used for training the two models without detector selection process. 4) Random neighboring sites: the detector sites are randomly selected from the 95 PV sites instead of using the proposed detector selection algorithm.

From Fig. 10 we can see that the two models trained on the selected neighbors have the best performance (i.e. smallest average RMSE and smallest RMSE variance). When all sites are used, the forecasting results can have very large RMSE variance. This is because the data from uncorrelated sites will mislead the machine learning model and therefore pollute the forecasting results. Solely relying on the historical data of the target site or randomly selected neighboring sites both lead to larger forecasting errors, as the spatial-temporal correlation information cannot be properly integrated into the forecasting model. Thus, the results demonstrate the necessity for the selection process of a detector network and the effectiveness of our proposed detector selection algorithm.

![Fig. 10. Violin plots of 1-hour ahead forecasting RMSE on 95 PV sites.](image)

We further test the performance of the TCN model for different forecasting horizons, and compare its performance with CNN-LSTM, persistence model [30] and SARIMA [31]. The results as shown in Figs. 11 and 12.

![Fig. 11. RMSE for different forecasting horizons with different detector selection strategies. Figures in the first row are TCN results and in the second row are CNN-LSTM results.](image)

![Fig. 12. Comparison of different forecasting models. The black horizontal dashed line shows the forecasting accuracy achieved in day-ahead stage by the physics-based model.](image)

From Fig. 11 and 12 we have the following observations:

- The value of a detector network mainly lies in the first 2-3 hours. From the first column of Fig. 11 we can see that the forecasting errors based on the selected neighbors are relatively low in the first 2-3 hours. Then, the errors increase dramatically. Meanwhile, in Fig. 12 we compare the real-time stage forecasting error with the physics-based model in the day-ahead stage (grey dashed line). It shows that when the forecasting horizon is longer than 2-3 hours, we will need to switch to the physics-based model to get more accurate forecasting results.
- TCN has better forecasting performance compared with the benchmark methods, as shown in Fig. 12.
- Compared with the proposed neighbor selection strategy
(1st column in Fig. 11), directly using all sites for training the forecasting model will lead to a higher forecasting error in the first few hours (3rd column in Fig. 11). However, we also observe that in a longer forecasting horizon (e.g., 5-6 hours ahead), using all site strategy leads to a lower forecasting error. This inspires us to further explore the optimal number/radius of neighbors for different forecasting horizons, as shown in Figs. 13 and 14.

![Fig. 13. Forecasting RMSE of the TCN model using different numbers/distances of neighboring sites.](image)

![Fig. 14. Forecasting error comparison. (a) Distance between the target and detector sites (b) Number of detector sites](image)

We can see that when the forecasting horizon increases, we will statistically need a larger detector network to include more neighboring sites to achieve better forecasting performance. When we forecast the next 5 minutes for a PV site (blue dots/curves in Figs. 13/14), we don’t need information from neighboring sites. When forecasting 6-hour ahead PV outputs (grey dots/curves in Fig. 13/14), a larger detector network including almost all the 95 PV sites is needed. This is because as the forecasting horizon increases, the machine learning model will need to project the cloud movements for a longer period of time. This requires information from the neighboring sites far away from the target site.

IV. CONCLUSION

This paper proposes a two-stage PV forecasting framework for MV-level PV farms. In the day-ahead stage, TCN is used to blend 5 different NWP data sources before being fed into the physics-based model. Compared with single NWP data source, the forecasting accuracy has a 37% improvement after NWP blending. In the real-time stage, the proposed scenario-based neighbor selection algorithm can automatically identify the most contributive neighbors for the target site by maximizing the pre-defined successful detection rate. Based on the selected neighbors, TCN can leverage the spatial-temporal correlation to achieve intra-hour forecasting for the target site. The intra-hour forecasting results has higher accuracy and granularity than the day-ahead stage, with an effective forecasting horizon up to 3 hours. The proposed two-stage forecasting framework is built based on the most commonly-available data of a PV farm (e.g., parameters of modules/inverters, inverter-level power output, etc.) without requesting additional measurement. As a result, the model is economic, succinct and training-efficient.

### References

[1] Antonanzas, Javier, et al. "Review of photovoltaic power forecasting." Solar energy 136 (2016): 78-111.
[2] Ahmed, Razin, et al. "A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization." Renewable and Sustainable Energy Reviews, 124 (2020): 109792.
[3] Das, Utpal Kumar, et al. "Forecasting of photovoltaic power generation and model optimization: A review." Renewable and Sustainable Energy Reviews, 81 (2018): 912-928.
[4] Chen, Xiaoyang, et al. "Sensor network based PV power nowcasting with spatio-temporal preselection for grid-friendly control." Applied Energy 255 (2019): 113760.
[5] Bosch, Juan L., and Jan Kleissl. "Cloud motion vectors from a network of ground sensors in a solar power plant." Solar Energy 95 (2013): 13-20.
[6] Zhou, Zhao, et al. "Pattern classification and PSO optimal weights based sky images cloud motion speed calculation method for solar PV power forecasting." IEEE Transactions on Industry Applications, 55.4 (2019): 3331-3342.
[7] Meng, Xiangjian, et al. "Inverter-Data-Driven Second-Level Power Forecasting for Photovoltaic Power Plant." IEEE Transactions on Industrial Electronics, 68.8 (2020): 7034-7044.
[8] Yang, Chen, Anupam A. Thatte, and Le Xie. "Multimime-scale data-driven spatio-temporal forecast of photovoltaic generation." IEEE Transactions on Sustainable Energy 6.1 (2014): 104-112.
[9] Agoua, Xwégnon Ghislain, Robin Girard, and George Kariniotakis. "Short-term spatio-temporal forecasting of photovoltaic power production." IEEE Transactions on Sustainable Energy 9.2 (2017): 538-546.
[10] Bessa, Ricardo Jorge, et al. "Probabilistic solar power forecasting in smart grids using distributed information." International Journal of Electrical Power & Energy Systems 72 (2015): 16-23.
[11] Vaz, A. G. R., et al. "An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in Utrecht, the Netherlands." Renewable Energy 85 (2016): 631-641.
[12] Yang, Dazhi, et al. "Solar irradiance forecasting using spatial-temporal covariance structures and time-forward kriging." Renewable Energy, 60 (2013): 235-245.
[13] Tascikaraoglu, Akin, et al. "Compressive spatio-temporal forecasting of meteorological quantities and photovoltaic power." IEEE Transactions on Sustainable Energy, 7.3 (2016): 1295-1305.
[14] Zhang, Ruiyuan, et al. "Data-driven photovoltaic generation forecasting based on a Bayesian network with spatial-temporal correlation analysis." IEEE Transactions on Industrial Informatics, 16.3 (2019): 1635-1644.
[15] Liu, Yongqi, et al. "Ensemble spatiotemporal forecasting of solar irradiation using variational Bayesian convolutional gate recurrent unit network." Applied Energy 253 (2019): 113596.
[16] Wang, Kejun, Xiaoxia Qi, and Hongda Liu. "A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network." Applied Energy 251 (2019): 113315.
[17] Jeong, Jaek, and Hongseok Kim. "Multi-site photovoltaic forecasting exploiting space-time convolutional neural network." Energies 12.23 (2019): 4490.
[18] Venugopal, Vignesh, Yuchi Sun, and Adam R. Brandt. "Short-term solar PV forecasting using computer vision: The search for optimal CNN architectures for incorporating sky images and PV generation
history." Journal of Renewable and Sustainable Energy, 11.6 (2019): 066102.

[19] Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arXiv preprint arXiv:1803.01271 (2018).

[20] Wan, Renzhuo, et al. "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting." Electronics, 8.8 (2019): 876.

[21] Lara-Benítez, Pedro, et al. "Temporal convolutional networks applied to energy-related time series forecasting." applied sciences, 10.7 (2020): 2322.

[22] He, Kaiming, et al. "Identity mappings in deep residual networks." European conference on computer vision. Springer, Cham, 2016.

[23] Zagouras, Athanassios, Hugo TC Pedro, and Carlos FM Coimbra. "On the role of lagged exogenous variables and spatio–temporal correlations in improving the accuracy of solar forecasting methods." Renewable Energy 78 (2015): 203-218.

[24] Benesty, Jacob, et al. "Pearson correlation coefficient." Noise reduction in speech processing. Springer, Berlin, Heidelberg, 2009. 1-4.

[25] Levinson, Ronnen, et al. "Experimental comparison of pyranometer, reflectometer, and spectrophotometer methods for the measurement of roofing product albedo." Solar Energy, 206 (2020): 826-847.

[26] William F. Holmgren, Clifford W. Hansen, and Mark A. Mikofski. “pvlb python: a python package for modeling solar energy systems.” Journal of Open Source Software, 3(29), 884, (2018).

[27] Lorenz, Elke, et al. "Irradiance forecasting for the power prediction of grid-connected photovoltaic systems." IEEE Journal of selected topics in applied earth observations and remote sensing 2.1 (2009): 2-10.

[28] Lu, Siyuan, et al. "Machine learning based multi-physical-model blending for enhancing renewable energy forecast-improvement via situation dependent error correction." 2015 European control conference (ECC). IEEE, 2015.

[29] Zang, Haixiang, et al. "Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations." Renewable Energy 160 (2020): 26-41.

[30] Dutta, Shreya, et al. "Load and renewable energy forecasting for a microgrid using persistence technique." Energy Procedia 143 (2017): 617-622.

[31] Vagropoulos, Stylianos I., et al. "Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting." 2016 IEEE International Energy Conference (ENERGYCON). IEEE, 2016.