Renewable Energy 108 (2017) 274–286

Contents lists available at ScienceDirect

Renewable Energy

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

Short-term photovoltaic power forecasting using Artificial Neural Networks and an Analog Ensemble

Guido Cervone [a, b, *], Laura Clemente-Harding [c, a], Stefano Alessandrini [b], Luca Delle Monache [b]

[a] Geoinformatics and Earth Observation Laboratory, Department of Geography and Institute for CyberScience, The Pennsylvania State University, University Park, PA, United States
[b] Research Application Laboratory, National Center for Atmospheric Research, Boulder, CO, United States
[c] Geospatial Research Laboratory, Engineer Research and Development Center, Alexandria, VA, United States

ARTICLE INFO

Article history:
Received 5 October 2016
Received in revised form
14 January 2017
Accepted 18 February 2017
Available online 21 February 2017

Keywords:
Solar power
numerical weather prediction
Artificial Neural Networks
Uncertainty estimation
Ensemble modeling
Parallel computing

ABSTRACT

A methodology based on Artificial Neural Networks (ANN) and an Analog Ensemble (AnEn) is presented to generate 72 h deterministic and probabilistic forecasts of power generated by photovoltaic (PV) power plants using input from a numerical weather prediction model and computed astronomical variables. ANN and AnEn are used individually and in combination to generate forecasts for three solar power plants located in Italy. The computational scalability of the proposed solution is tested using synthetic data simulating 4450 PV power stations. The National Center for Atmospheric Research (NCAR) Yellowstone supercomputer is employed to test the parallel implementation of the proposed solution, ranging from one node (32 cores) to 4450 nodes (141,140 cores). Results show that a combined AnEn + ANN solution yields best results, and that the proposed solution is well suited for massive scale computation.

Published by Elsevier Ltd.

1. Introduction

Building a sustainable society requires providing solutions that meet societal needs and will last for generations to come. The current reliance on finite environmental resources to meet the power needs of the world's expanding population and economy is not sustainable in the long term [18]. Renewable energy sources provide a potential sustainable solution to meet societal power needs. This article describes a methodology for generating deterministic and probabilistic forecasts of photovoltaic (PV) power generation, which are specific tasks required to rely on renewable sources for a portion of the energy production requirements.

Becker et al. [6] analyzed 32 years of weather data investigating the feasibility of U.S. reliance on wind and solar power to satisfy the country's power requirements. They concluded that the U.S. has adequate meteorological and terrain characteristics to suggest that renewable sources can be successfully implemented. The Renewable Electricity Futures Study (RE Futures) used two power generation models to conclude that up to 80% of U.S. electricity demand could be met through renewable resources [12]. Arent et al. [3] drew upon RE Futures results to conclude that a high reliance on renewable sources necessitates a number of structural modifications that positively affect both supply chains and the environment. With respect to PV production in the U.S. the southwest has the highest solar radiation and most areas of the country are viable candidates, including regions like PA and NJ where the solar radiation is comparable to northern Spain.

Brazil is an example of a country with significant potential for PV penetration [19]. Distributed PV power can provide energy to mitigate peak loads when air conditioning is greatest in urban areas and minimize the energy loss caused by energy traveling longer distances. PV presents an opportunity in the Brazilian Amazon where connectivity to a main grid is not available and Diesel generators are the main power source for independent mini-grids.

Specifically, Lima et al. [19] studied Northeastern Brazil using

* Corresponding author. Geoinformatics and Earth Observation Laboratory, Department of Geography and Institute for CyberScience, The Pennsylvania State University, University Park, PA, United States.
E-mail addresses: cervone@psu.edu (G. Cervone), laura@psu.edu (L. Clemente-Harding), alessand@ucar.edu (S. Alessandrini), lucadm@ucar.edu (L. Delle Monache).

http://dx.doi.org/10.1016/j.renene.2017.02.052
0960-1481/Published by Elsevier Ltd.

numerical weather prediction (NWP) model output, ground observations, and a series of ANN to develop a methodology for improving 24 h solar irradiance forecasts in the Northeastern region of Brazil. This methodology resulted in the identification of spatial patterns in the data and an improvement in the solar irradiance forecasts when using ANN where the ANN reduces the general overforecasting of solar irradiance in the NWP output.

Deep penetration of renewable energy in the existing power grid, especially distributed PV systems, is needed to sustain our society and its expanding economy [21]. Multiple challenges to renewable energy integration exist including but not limited to: power demand and supply fluctuations, meteorological conditions, infrastructure challenges and spatial and temporal changes on both demand/supply sides [5,19,22,33]. Several challenges of particular relevance to this research are described in further detail. One of the largest factors is the quantification of uncertainty associated with the estimation of future power output, which unlike traditional generators is variable and correlated to rapidly changing local atmospheric conditions [17,19,20,22,31].

Recent research investigates the optimization of fossil fuel power penetration by assuming an elastic production in response to the variability of renewable sources [11,17]. Estimating this uncertainty is paramount for the widespread use of PV system, especially for distributed residential PV which are operated and maintained independently [33]. This uncertainty can be quantified using ensemble simulations which provide insights to the probability of possible outcomes, and in turn quantify the uncertainty which can help decrease costs [1,30].

In the immediate term (seconds to minutes), use of uncertainty information is used to control smart inverters, which lower ramp-events that can damage the grid, and are highly taxed in some markets, thus reducing the profitability of the system [7,28,33]. In the short term (24–72 h or the day ahead market), uncertainty is used to compute the risk of over- or under- electricity production with respect to demand [23]. Under-producing electricity means that the power deficit must be satisfied by buying electricity from the grid, by generating it using non-renewable sources or storage facilities, or that the demand cannot be satisfied causing black-outs. Similarly, over-producing electricity presents an opportunity loss for electricity which could have been sold. This research addresses precisely this challenge, and creates an advanced CyberInfrastructure (CI) solution based on massive supercomputer simulations required to address the day-ahead uncertainty problem for thousands of PV stations.

The proposed methodology, based on Artificial Neural Networks (ANN) and the Analog Ensemble (AnEn), generates power forecasts for PV farms as a function of meteorological and environmental parameters. The ANN is used to generate a deterministic forecasts, whereas the AnEn is used to generate both deterministic and probabilistic forecasts. The major advantages of the AnEn method are its ability to provide reliable and bias-calibrated forecasts, and its computationally scalable algorithm is well suited for parallel processing [10].

The proposed methodology is domain independent and is not limited to the specific application presented. In general, the proposed method can be applied to all situations where a set of deterministic past forecasts and associated observations are present.

This paper addresses two main scientific goals:

1. To test AnEn, ANN, and a combined AnEn + ANN methodology to assess the 72 h deterministic and probabilistic forecasts of power generated by three PV stations.
2. To analyze and evaluate the computational efficiency of the methodology performing massive scale simulations for thousands of simulated PV stations using a supercomputer.

2. Data and infrastructure

The statistical analysis presented is based on real world observations of power generated by PV solar farms and atmospheric NWP model data. Analysis of the complexity of the algorithm is based on a synthetic data including thousands of simulated solar farms. This synthetic dataset, described in Section 2.3, is used only to test the scalability of the AnEn algorithm, and not to draw any conclusions on the performance of the methodology. The scalability of the AnEn algorithm for massively parallel applications is tested using over 140,000 cores of the Yellowstone supercomputer.

The overall goal of this line of research is to create the energy smart grid of the future that includes a significant fraction of energy generation from renewable sources [32].

2.1. Observations

In-situ observations of power generated by PV stations were collected at three plants located in different regions of Italy, Lombardy (SL), Calabria (SC) and Sicily (SS), respectively (Fig. 1). Table 1 summarizes the main characteristics of the data collected for each station in terms of length of the dataset, division of the training and testing sets, amount of missing data, days with less than 4/8 percent cloud coverage (sunny days), and the ratio of mean to nominal power.

SL and SS are nearly identical in terms of PV panels and electronic components and have a nominal power of 5.21 KW (roughly a typical residential roof installation), while the SC PV solar farm is

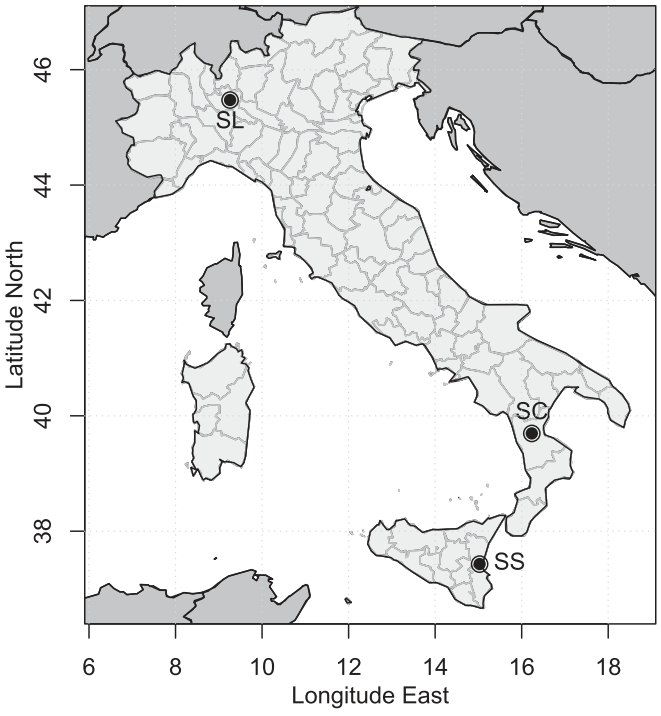

Fig. 1. Map of Italy showing the location of the three solar power stations used in this study. The SL station is located in the northern part of the country, the Lombardy region. The SC and SS stations are located in the southern part of the country, respectively in the Calabria and Sicily regions.

Table 1
Characteristics of the three stations (SL, SC, SS) used in this study. CC stands for cloud covered days lower than 4/8 and MP/NP is the ratio for mean power over nominal power.

Station	Total Days	Train Days	Test Days	Missing Data	Sunny Days %	Nominal Power	Ratio MP/NP
SL	549	365	184	9.5%	45%	5.21 KW	25%
SC	731	365	366	0.0%	60%	4.99 MW	30%
SS	730	365	365	0.0%	66%	5.21 KW	35%

much larger with a nominal power of 4.99 MW. Observations of power metered (*PM*) are available as hourly averages but have different temporal coverage at each station: SL from July 2010 to December 2011, SC from April 2011 to March 2013, and SS from January 2010 to December 2011. The SL station is missing about 9% of observations, while SC and SS have no missing data.

Fig. 2 shows a statistical summary of the power generated by SL (left), SC (center) and SS (right). While the axis for the stations are different (SL and SS are in KW, while SC is in MW), all three stations show very similar statistical behaviors. In the chart, the upper (UE) and lower (LE) extremities are defined as

$$UE = Q3 + 1.5 \times IQR \tag{1}$$

$$LE = Q1 - 1.5 \times IQR \tag{2}$$

where Q1 and Q3 are the first and third quartiles, and IQR is the interquartile range. Naturally, there is a very strong diurnal component with the majority of power generated between 06h00 and 18h00, with a peak at 12h00 (local solar noon).

The first station, SL, is located in the suburban area of Milan, in the northern Lombardy region. This plant is characterized by high concentrations of aerosol particulates due to anthropogenic emissions, a colder climate with several snow days, high cloud cover and fog. The second station, SC, is located in the southern Calabria region and is the largest of the three. This station provides the most reliable data due to its size, quality control, and for the maintenance of the panels themselves. The third station, SS, is located in Sicily, and the site is characterized by the presence of volcanic ash. Variable amounts of volcanic ash are released by the nearby Etna volcano. This variability influences the power production due to a) atmospheric dimming of solar irradiance and b) deposition of ash over the panels resulting in decreased efficiency. Numerical atmospheric weather models do not usually take volcanic ash emissions into account, making power forecasting more difficult.

A climatic analysis based on meteorological observations collected near the three solar power farms shows that the yearly average of the fraction of days with an average cloud cover lower than 4/8 is around 45% (SL), 60% (SC) and 66% (SS). This is reflected in the ratio between mean power produced (MP) and nominal power (NP) that equates to approximately 25% (SL), 30% (SC), and 35% (SS).

2.2. Model forecasts

The Regional Atmospheric Modeling System (RAMS), a mesoscale atmospheric model, is used to generate deterministic weather forecasts at the locations of the three solar power stations (Fig. 1). The computational domain consists of two nested grids with horizontal grids of 15×15 km^2 and 5×5 km^2. Each model run starts at 00 UTC and is 72 h long. Therefore, each time the model is run, RAMS generates forecasts for the following three days at hourly intervals. RAMS is initialized with boundary conditions from the ECMWF deterministic forecast fields starting at 00 UTC with 0.125° spatial resolution. The Harrington parameterization is used for the radiation scheme, and a bulk microphysics parameterization is implemented, accounting for full moisture complexity [13].

Forecast data for each of the three stations is composed of five predictor variables (three from RAMS and two computed), 72 forecast lead times (FLT) and a variable number of days as shown in Table 1 for 2010 and 2011. The predictor variables computed by RAMS are global horizontal irradiance (*GHI*), percent cloud cover (*CC*) and air temperature at two meters above the surface (*T2M*). Additionally, two predictor variables for solar azimuth (*AZ*) and elevation (*EL*) are computed as function of time of day and day of the year. These two predictors are fundamental for quantifying seasonal variability of solar irradiance. The 72 FLTs correspond to the forecast interval for the three day period.

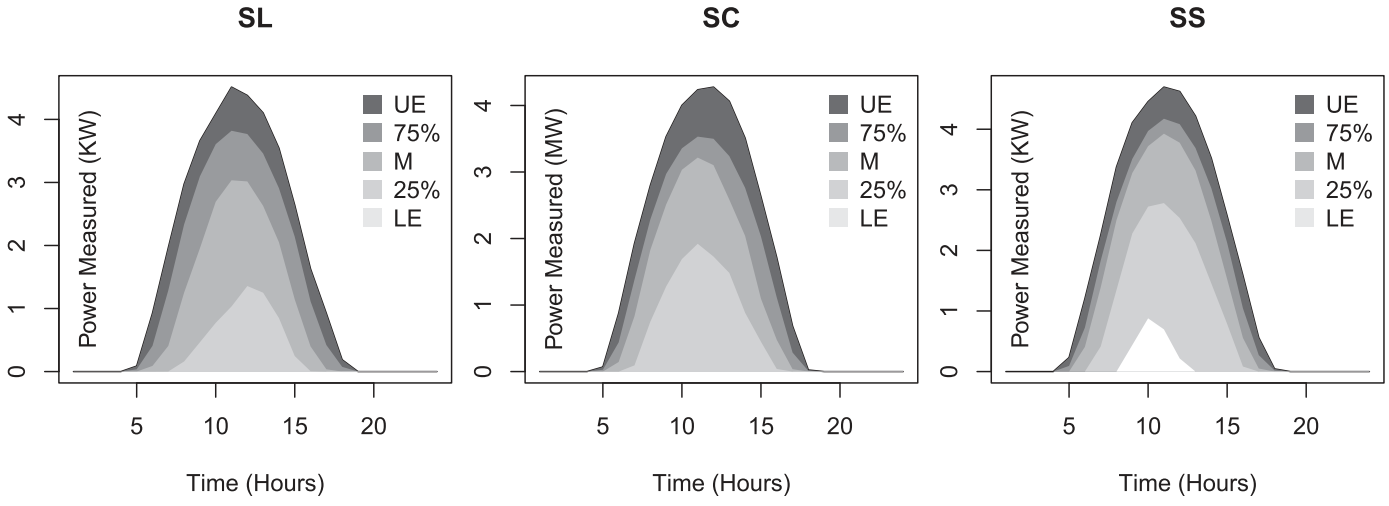

Fig. 2. Statistical summary of power generated by SL (left), SC (center) and SS (right). UE = Upper Extremity, 75% = third quartile, M = median or second quartile, 25% = first quartile, LE = Lower Extremity.

2.3. Synthetic data

A very large synthetic dataset is created to test the scalability of the AnEn algorithm in a massively parallel environment. The data is created using a statistical model based on observations and NWP model data. The solar elevation and azimuth are computed for the longitude and latitude location of each of the simulated station, and are used to estimate the maximum theoretical power (*MTP*) produced. The *PM* value for each of the simulated stations is computed by selecting a random day in the dataset from the closest of the three observed stations, and adjusting the observed *PM* value taking into account the difference in *MTP* at the observed and simulated station. This statistically consistent dataset is only used for scalability performance evaluation, and not to test the statistical performance of AnEn.

A total of 4450 stations were simulated at equally spaced locations throughout the Italian peninsula. Each simulated station data contains six predictor variables and an associated *PM* variable. Each station has a length of 1460 days and 72 FLTs. The simulated predictor variables include $GHI, CC, T2M$, and computed EL and AZ at each location. An additional predictor variable is created using a neural network in the same manner as is done for the real experiments (see Section 3.1). A total of over 46 million values are generated [4450 stations $\times$ 6 predictor variables $\times$ 1 output variable $\times$ 1460 days] for this synthetic dataset.

2.4. Yellowstone supercomputer

The experiments presented are performed using the NCAR Yellowstone Supercomputer, the fastest supercomputer designed primarily for Earth and atmospheric science research. The supercomputer is physically located in Wyoming and is linked to the main NCAR Boulder campus through a high speed network.

Yellowstone is a 1.5-petaflops high-performance IBM iDataPlex cluster featuring 4536 nodes comprising of a total of 72,576 Sandy Bridge cores and 144.6 TB of memory. Each node has a dual 2.6-GHz Intel Xeon E5-2670 8-core processor, resulting in 16 available cores per node. Furthermore, each core can be enabled for hyper-threading computation, which simulates twice as many cores (32) per node yielding 145,152 cores. Hyper-threading is a proprietary Intel technology allowing each core to perform multiple tasks at once, thus simulating double as many cores as is physically available. The benefit of hyper-threading varies from problem to problem and has been shown to greatly increase performance for multimedia and general purpose applications. However, it has also been shown to offer little to no improvement for intensive numerical operations.

To test the scalability of the AnEn algorithm in a real world scenario, the entire Yellowstone supercomputer was allocated to perform tests using the synthetic data generated.

3. Methodology

3.1. ANN algorithm

Artificial Neural Networks (ANN) are a family of well-established machine learning supervised classifiers inspired by biological neural networks [14]. They require labeled data, which is used to train a network that predicts the independent variable as a function of multiple dependent input variables. The independent variable is typically a continuous numerical variable. Once the network is trained, it can be used to classify an unknown independent variable as function of the input variables.

ANN are implemented as layers of interconnected nodes, also called neurons. The number of layers is highly variable, and depends on the characteristics of each problem. ANNs require a minimum of three layers, one for the input nodes, one hidden, and one for the output nodes (e.g. Fig. 3), which is also the configuration used in this study. The number of input and output nodes are defined by the problem, whereas the number of hidden nodes is a crucial parameter which must be set and can be optimized. In this study, the number of hidden nodes was optimized using a brute force algorithm to test various network sizes. Additionally, bias nodes are included in all layers, except the input layer, and are used to control the overall behavior of the layer.

Connections between the nodes contain weights that are iteratively updated during a training phase. In this study, the back-propagation (backprop) algorithm is used to train the ANN [25]. This is a stochastic algorithm which uses random initial weights for the links between the nodes, and then iteratively updates the weights to minimize the error between the input and network prediction. Each network configuration is repeated 30 times with different initial random seeds to account for the stochasticity of the backprop algorithm.

The ANN is employed to generate deterministic forecasts of *PM* at a specific lead time as a function of the six dependent variables $GHI, CC, T2M, AZ, EL$, plus the lead time t.

$$PM_t = ANN(GHI, CC, T2M, AZ, EL, t) \tag{3}$$

The original data for each station, comprised of the observed independent variable (PM), the RAMS weather forecasts $(GHI, CC, T2M)$ and the two computed variables (AZ, EL) are represented as a two dimensional matrix with dimensions [DAYS $\times$ FLT, VARS] where DAYS vary from station to station (see Table 1), FLT = 72, and VARS = $\{GHI, CC, T2M, AZ, EL, PM, t\}$.

Therefore, for each day, the multi-variate time series of length FLT = 72 are transformed into a larger number of smaller time series of length VARS = 7. This lead time based representation of the data improves ANN results because it provides a larger training set for the ANN, and allows the output to be a single value (*PM* at a specific time) instead of a time-series 72 points long. The diurnal and seasonal signals of the data are captured by the computed

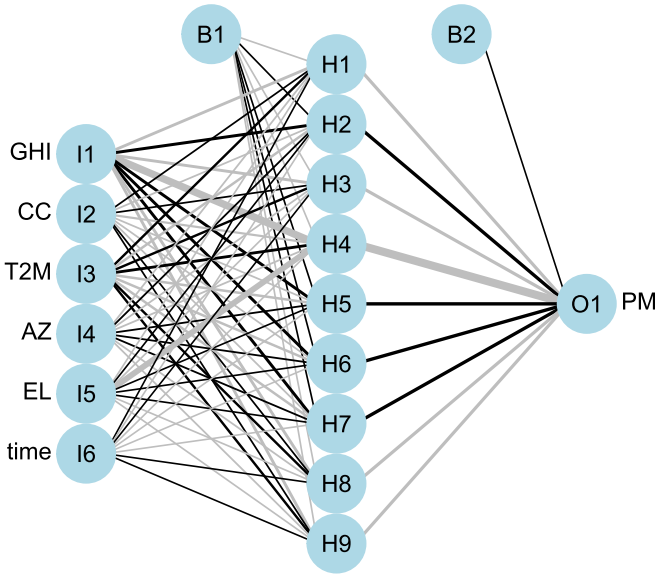

Fig. 3. ANN created for the SS station using 9 hidden layers. The left most layer of nodes are the input $(GHI, CC, T2M, AZ, EL, t)$, and the right node is the output (PM). The H nodes are the hidden nodes and B nodes are the bias for the hidden and output layer. Shading and thickness of the links are proportional to the weights learned by the backprop algorithm.

variables *AZ* and *EL*.

Fig. 3 shows the ANN generated for the SS station using an input layer of six nodes (I1-I6), a hidden layer composed of nine nodes (H1-H9), an output layer of one node (O1) and two bias nodes (B1,B2). The grey shades and thickness of the links are proportional to the weights assigned by the backprop algorithm. It is not possible to analytically evaluate ANN accuracy just by inspecting the links and weights, and it is in fact usually referred to as a 'black box'. Therefore, it is critical to test the predictive accuracy of the network.

Fig. 4 shows a sample prediction made by the network in Fig. 3 for a specific day. The solid line indicates the ANN prediction and the dots indicate the actual observations. The grey bars indicate the foretasted cloud cover in percentage. This example shows how the neural network grossly underestimates the *PM* in the first 24 h, slightly underestimates between 24 and 48 h, and slightly overestimates between 48 and 72 h. The *PM* underestimation in the first 24 h can be easily explained by RAMS overestimating the *CC* (grey bars). In fact, in correspondence of high foretasted *CC*, the neural network consistently show a decreased *PM* estimate.

3.2. AnEn algorithm

The AnEn generates probabilistic predictions using a single deterministic NWP, a set of past forecast predictions, and their corresponding observations. The AnEn technique compensates for the model bias by taking past errors into account. The main assumption is that if similar past forecasts can be found, the model error can be estimated. Specifically, the AnEn seeks to estimate the probability distribution of the observed future value of the predictand variable given a model prediction, which can be represented as $p(y \backsim f)$ where, at a given time and location, y is the unknown observed future value of the predictand variable and f are the values of the predictors from the deterministic model prediction at the same location and over a time window centered over the same time. Delle Monache et al. [10] describe several attractive features of the AnEn including the use of higher resolution forecasts and no need for initial condition perturbations, running multiple model instances, or post processing requirements. The AnEn is able to capture the flow-dependent error characteristics and show superior skill in predicting rare events when compared to state-of-the-art post processing methods [8–10].

Analogs are sought independently at each location and all forecasts are initialized at 00Z. The best-matching historical forecasts for the current prediction are selected as the analogs. The best match is determined by the metric described in Delle Monache et al. [10] and Delle Monache et al. [9].

$$\|F_t, A_{t'}\| = \sum_{i=1}^{N_v} \frac{w_i}{\sigma_{f_i}} \sqrt{\sum_{j=-\tilde{t}}^{\tilde{t}} \left(F_{i,t+j} - A_{i,t'+j}\right)^2} \tag{4}$$

where F_t is the forecast to be corrected at the given time t and station location A_t is an analog forecast at time t' before F_t was issued and at the same location, N_v and w_i are the number of predictors and their weights, respectively; σ_{f_i} is the standard deviation of the time series of past forecasts of a given variable at the same location; $\tilde{t}$ is an integer equal to half the width of the time window over which the metric is computed; and $F_{i,t+j}$ and $A_{i,t'+j}$ are the values of the forecast and the analog in the time window for a given variable.

The metric describes the quality of the analog chosen and is based upon the similarity of the current forecast window to the past forecast time windows available in the dataset. The top N analogs are selected from past dates within the historical dataset. Next, the verifying observation for each of the N best analogs is selected. Together, the verifying observations generate the N members of the ensemble prediction for the current forecast. In this research, the AnEn is used to generate probabilistic forecasts, both by itself and in combination with the ANN. The number of ensemble members N is 21 for each station. As a general rule, the number of ensemble members is set equal to the square root of the training data. However, this value can be optimized, globally or as function of the forecast lead time which has not been attempted in this study.

Fig. 5 shows the *PM* probabilistic forecast (shades), ensemble mean (dashed line), and observations (points) generated using AnEn for the 41st test day at the SS station. In comparison with Fig. 4, AnEn probabilistic forecasts are able to better predict the observations.

3.2.1. Predictor weighting

The weights, w_i, can all be set to 1 to weight each predictor

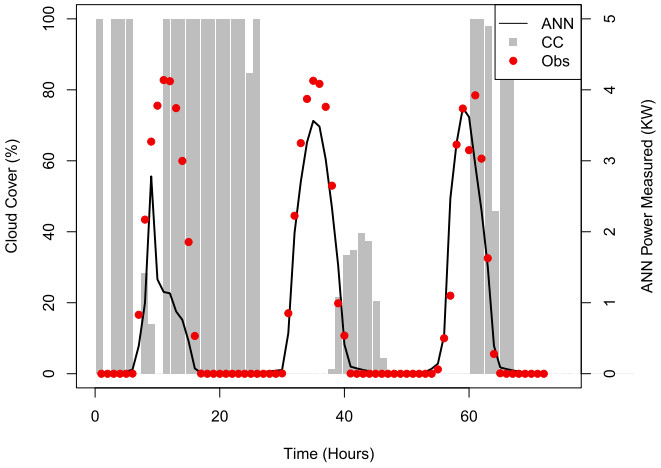

Fig. 4. PM power forecast (solid line) and observations (points) generatred using ANN for the 41st test day of the SS station. The grey bars indicate the cloud cover in percentage. The three peaks correspond to the diurnal cycles for 72 h (three days).

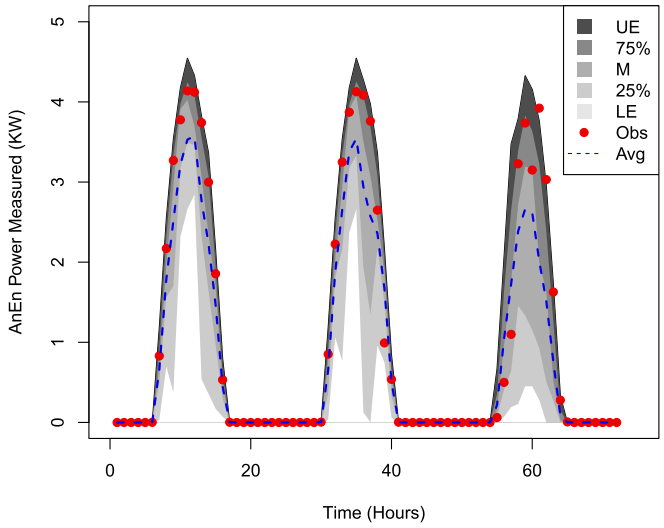

Fig. 5. PM probabilistic forecast (shades), ensemble mean (dashed line), and observations (points) generatred using AnEn for the 41st test day at the SS station. UE = Upper Extremity, 75% = third quartile, M = median or second quartile, 25% = first quartile, LE = Lower Extremity.

equally or they can be optimized. Junk et al. [16] have shown how different weighting schemes for the AnEn metric can lead to improved results. In this study a brute force algorithm is employed ranging the weights between 0 and 1 and with a step increment of 0.1. Only combinations where the sum of the weights is equal to one is considered, leading to 1002 combinations in case of the original five variables, and 3004 when the ANN predicted *PM* is also included. In both cases, the default case with each weight set to one is also added for comparison purposes.

To determine the best weights, a leave-one-out methodology is employed using only the training set because the test set is assumed to be unknown and represents the future. Analogs are computed by iteratively testing on a single day (the leave-one-out), and using all remaining days as training. The error is defined as the root mean squared error between the ensemble means and the observations. The process is repeated for all days in the training set, for all the weights combinations described above (1002 and 3004 combinations), and for each of the three stations SL, SC, and SS. The weighting scheme that achieves best results on the training set is used for performing all tests described in the results over a period that does not overlap with the training.

3.2.2. Parallel implementation of AnEn

The AnEn code is well suited for parallelization because all computations across locations and lead times are independent and can occur in separate processes. It falls within the category of 'embarrassingly parallelizable' code because it naturally allows for most processes to be run as parallel tasks. The AnEn algorithm was implemented in JAVA and Python to facilitate optimal use of multiple cores and multiple nodes. Specifically, JAVA multi-threading, the ability to use multiple CPU cores, was used to parallelize the code within a single node. Python was used to distribute the computation across multiple nodes. Experiments testing the scalability of the algorithm are performed using the NCAR Yellowstone supercomputer.

Algorithm 1. Algorithm describing the AnEn implementation.

```
Data: Predictors[parameters, stations, days, FLT],
      Predictand[stations, days, FLT]
Result: Analogs[stations,days,FLT,ensemble
        members]
Initialization;
for all stations do
    for all test days do
        Compute the standard deviation SD
        for all parameters do
            for all training days do
                for all FLT do
                    Compute error between testing
                    and training data
                end
            end
        end
        Select best metric (QuickSelect or
        PartialSort)
    end
end
```

A typical AnEn execution consists of a series of five nested loops (Algorithm 1). The first two loops are entirely independent because analogs are computed separately for each station and each test day and their parallelization does not present any difficulty. The remaining three loops are necessary for computing the AnEn metric and can also be parallelized, but they require process synchronization because they read and write the same data. Additionally, for each computation of the similarity metric, the standard deviation must be calculated and results sorted. These are sequential tasks.

3.3. AnEn + ANN

The algorithms are fused to determine if a combined methodology can produce a forecast with greater skill. Deterministic $PM-ANN$ output from the ANN is used as an additional predictor in the AnEn. The AnEn uses the six variables ($GHI, CC, T2M, AZ, EL$, and $PM-ANN$) to generate a probabilistic forecasts (ANN + AnEn) of *PM*. When not used in combination with ANN, the AnEn is executed using five predictors: $GHI, CC, T2M, AZ$ and EL.

4. Results

4.1. Deterministic forecasts

Both the ANN and AnEn are employed to generate deterministic forecasts of power produced by the three PV stations as a function of the predictor variables. Deterministic forecasts are created for each of the three stations using the number of training and testing days defined in Table 1.

For ANN, 17 different network sizes (number of nodes in the hidden layer) are tested, from 4 to 20. The learning of each network size is repeated 30 times with different initial random seeds. A total of 1530 networks are learned [3 stations x 17 sizes $\times$ 30 times]. The network achieving the best prediction over the training data is selected to generate deterministic forecasts for the test period. Specifically, for the SL, SC, and SS stations, the best network sizes found are 8, 10, and 9, respectively.

Experiments are performed to identify the best weighting scheme for the AnEn metric defined in Equation (4). Specifically, a brute force algorithm tested all combinations, as described in Section 3.2. The results of the brute force search are shown in Table 2. The top three rows show the best weights when the additional $PM-ANN$ predictor is present. The bottom three rows show the results with only the original five predictors.

Without exception, $PM-ANN$ has an optimal weight of 0.5 (half the weight of the entire metric) every time it is employed. This suggests $PM-ANN$ is an important predictor because it captures the nonlinear interactions among the predictors. The other predictors are weighted more or less equally, with the exception of EL which is always omitted (weight = 0).

When $PM-ANN$ is not used, *GHI* is the most important predictor for the two southern stations SC and SS. However *CC* is found to be the most important predictor for SL. This is consistent with the larger presence of cloudy days suggesting this parameter is more important at this station.

Both deterministic and probabilistic forecasts need to be assessed through the use of standard verification measures [15,24]. Table 3 shows results for both the deterministic and probabilistic forecasts. The first column indicates the weighting scheme used, where default means that all predictors are weighted equally, and optimal uses the weights defined in Table 2. The second column indicates the algorithm used, which include the ANN and AnEn if

Table 2
Optimal weights identified for the three stations using the AnEn methodology. The top three rows show the best weights when using the $PM-ANN$ as an additional predictor and the bottom three rows when only the original five predictors are used.

Station	GHI	CC	T2M	AZ	EL	PM-ANN
SL	0.2	0.1	0.1	0.1	0.0	0.5
SC	0.3	0.2	0.0	0.0	0.0	0.5
SS	0.1	0.0	0.3	0.1	0.0	0.5
SL	0.2	0.5	0.2	0.1	0.0	
SC	0.8	0.0	0.0	0.1	0.1	
SS	0.6	0.4	0.0	0.0	0.0	

Table 3
Summary statistics for the deterministic and probabilistic forecast experiments. Best results are shown in bold.

Weighting	Algorithm	Deterministic Measures				Probabilistic Measures			
		Metric	SL	SC	SS	Metric	SL	SC	SS
	ANN	$\frac{\text{RMSE}}{\text{NP}}$	8.10%	**7.07%**	8.88%	MRE			
Default	AnEn		8.39%	7.89%	9.07%		−2.78%	0.59%	−1.75%
Optimal	AnEn		8.21%	7.62%	8.97%		−2.06%	−0.53%	−2.12%
Default	AnEn + ANN		8.27%	7.53%	8.83%		−2.60%	0.70%	−2.09%
Optimal	AnEn + ANN		**8.09%**	7.39%	**8.66%**		**−1.85%**	**0.38%**	**−1.53%**
	ANN	CORR	8.90	9.00	9.00	$\frac{\text{CRPS}}{\text{NP}}$			
Default	AnEn		**9.30**	9.40	9.40		2.78	2.44	2.77
Optimal	AnEn		**9.30**	**9.50**	9.40		2.68	2.38	2.75
Default	AnEn + ANN		**9.30**	**9.50**	**9.50**		2.72	2.32	2.68
Optimal	AnEn + ANN		**9.30**	**9.50**	**9.50**		**2.64**	**2.29**	**2.63**
	ANN	$\frac{\text{BIAS}}{\text{NP}}$	−0.39%	−0.61%	−0.48%	Spread Skill			
Default	AnEn		0.20%	3.53%	0.04%		9.58	9.84	9.77
Optimal	AnEn		−0.25%	−0.11%	−0.11%		**9.84**	9.67	9.71
Default	AnEn + ANN		0.12%	**−0.00%**	**−0.03%**		9.57	9.83	9.79
Optimal	AnEn + ANN		**−0.10%**	−0.34%	−0.15%		9.52	**9.85**	**9.82**

the algorithms are used individually or AnEn + ANN when the PM-ANN predictor is used.

Standard measures such as RMSE, CORR, and BIAS are used to evaluate the deterministic forecasts. BIAS estimates the variation centered about the mean identifying a general tendency of the forecast system. In this study, BIAS reflects the tendency of the ANN, AnEn or AnEn + ANN to over or under predict PM. RMSE identifies the standard deviation of the forecast error and aids in accuracy determination. The correlation verification statistic will determine forecast performance even if any systematic correction or re-scaling has occurred, in this case potentially through employing the ANN.

In terms of RMSE ANN outperforms all AnEn combinations for the SC station, whereas a combination of AnEn + ANN with optimal weights outperforms all other methods for the remaining two stations. In terms of correlation, all AnEn results are very close and outperform the ANN. Similarly for BIAS, all AnEn results outperform ANN. Beyond outperforming the ANN, the bias associated with the AnEn results indicates a slight tendency of the AnEn + ANN (best solution) to under predict *PM* at the SL and SS stations.

Fig. 6 shows scatter plots of observed vs. simulated *PM* values for each of the three stations (SL top, SC center, SS bottom) for ANN (left columns) and AnEn + ANN using the best optimal weights. The correlation, along with the station ID, is indicated at the bottom right corner of each graph. The dashed 45° line indicates a perfect match between predictions and observations. A linear model (LM) is fit to the data and shown with a dotted line. The intercept and slope of the LM is shown in the caption of each figure. The figure shows that all methods tend to underestimate high *PM* values.

This result can be explained with two main arguments. First, that high values are harder to predict because they are most affected by errors in the model forecasts. Second, the models are likely to overfit the training data.

Overall, both ANN and AnEn (all combinations) perform very similarly in terms of the deterministic measures employed, but their combination (AnEn + ANN) results in the best performing method. Execution of the AnEn is much faster than the ANN, but an in depth performance comparison of the two algorithms is not performed because implementation details can greatly affect the speed. Whereas the AnEn was specifically optimized to run on this dataset, the ANN employed is a general purpose algorithm not written by the authors.

4.2. Probabilistic forecasts

Probabilistic forecasts are generated using AnEn and AnEn + ANN, with default and optimal weighting. As discussed in the previous section, the optimal weights are shown in Table 2. Table 3 describes results obtained by the various tests performed. Statistical verification measures of probabilistic processes consist of reliability, resolution and sharpness [15]. Several standard and accepted verification measures can be used to determine the properties of the probabilistic output investigated through this research. Missing rate error (MRE), which is related to rank histograms, continuous ranked probability score (CRPS), and spread skill are three well accepted measures used in this study to verify the ANN, AnEn and AnEn + ANN forecasts.

The CRPS is the equivalent of the Brier Score integrated over all possible threshold values. It compares a full probabilistic distribution with the observations where both are represented as cumulative distribution functions (CDF). A lower value of the CRPS indicates better performance [2]. Rank histogram diagrams provide a means to determine the statistical consistency of an ensemble. If the observation is indistinguishable from the ensemble member (a requirement for a perfect ensemble) the rank histogram should be flat with all bars indicating the same probability [2]. The MRE is shown in Table 3 and Fig. 7 for each station. An ideal MRE is equal to zero, and it identifies positive values with under dispersion and negative values with over dispersion in the ensemble. The spread skill diagram indicates the ability of an ensemble to quantify the uncertainty of the prediction. The closer to the 1:1 line the better.

MRE values indicate that the optimized AnEn + ANN algorithm produced results with the least amount of dispersion and that predictions tended to be over dispersive at SC and under dispersive at SL and SS. CRPS indicates the AnEn + ANN with optimal weights produces the most skilled forecast of *PM* values. At SL, the AnEn with optimized weights also shows significant promise with the combination of MRE, CRPS and spread skill values. In terms of spread skill, SC and SS both indicating the optimized AnEn + ANN provides the best results.

In Fig. 7, spread skill diagrams are depicted with 95% confidence intervals for the ANN and the AnEn + ANN computed by bootstrapping. The 45° line indicates a perfect spread-skill line, and depicts the spread skill of both the AnEn and the combination AnEn + ANN at each location. The rank histogram shows the statistical consistency of an ensemble, and both the AnEn and the combination AnEn + ANN have comparable results. The middle

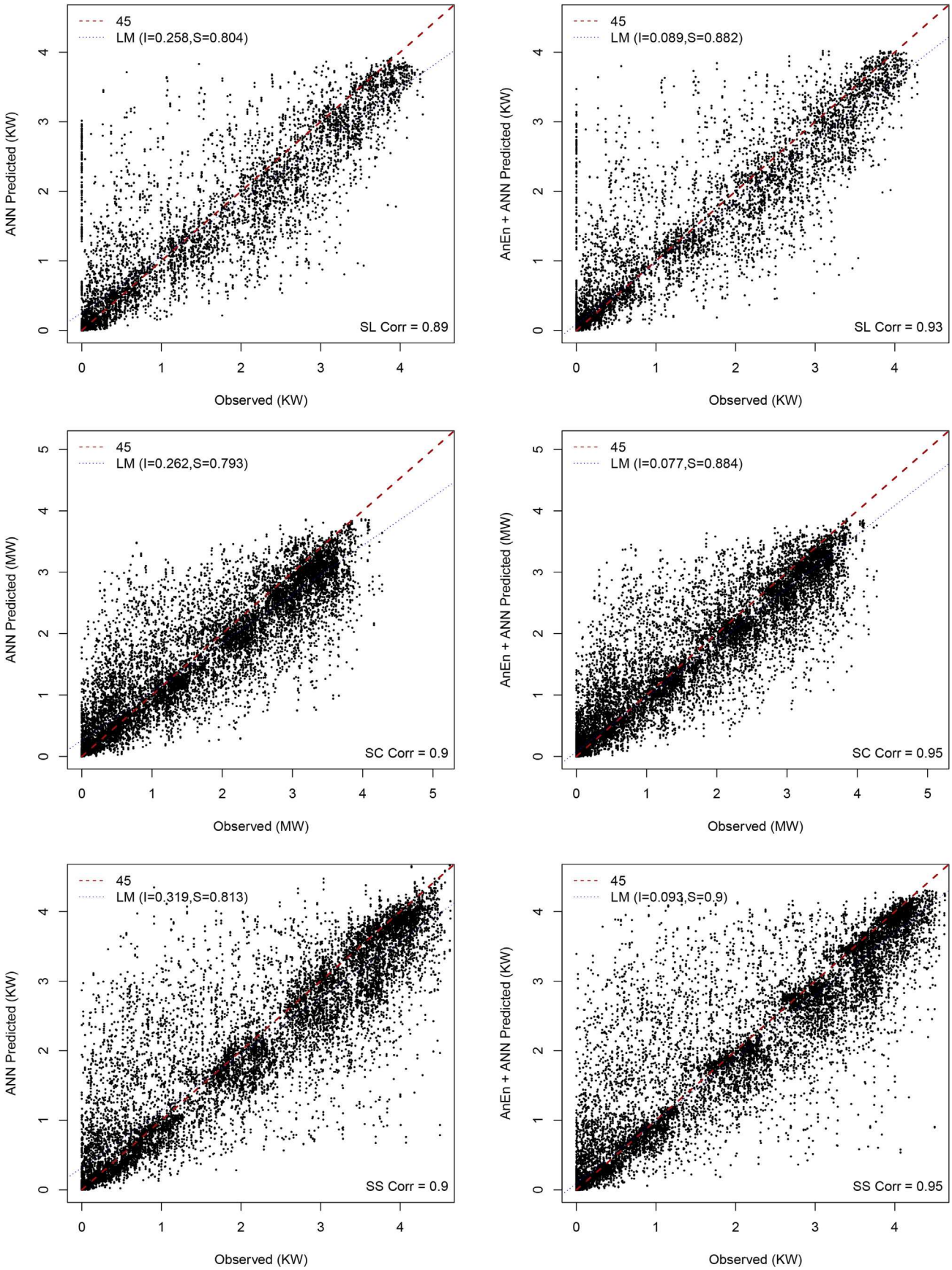

Fig. 6. Summary of the results obtained for the SL, SC and SS stations. Charts in the left column compare observations and *PM* predicted by the ANN while the right column compares observations and *PM* predicted by the AnEn + ANN with optimized predictor weighting.

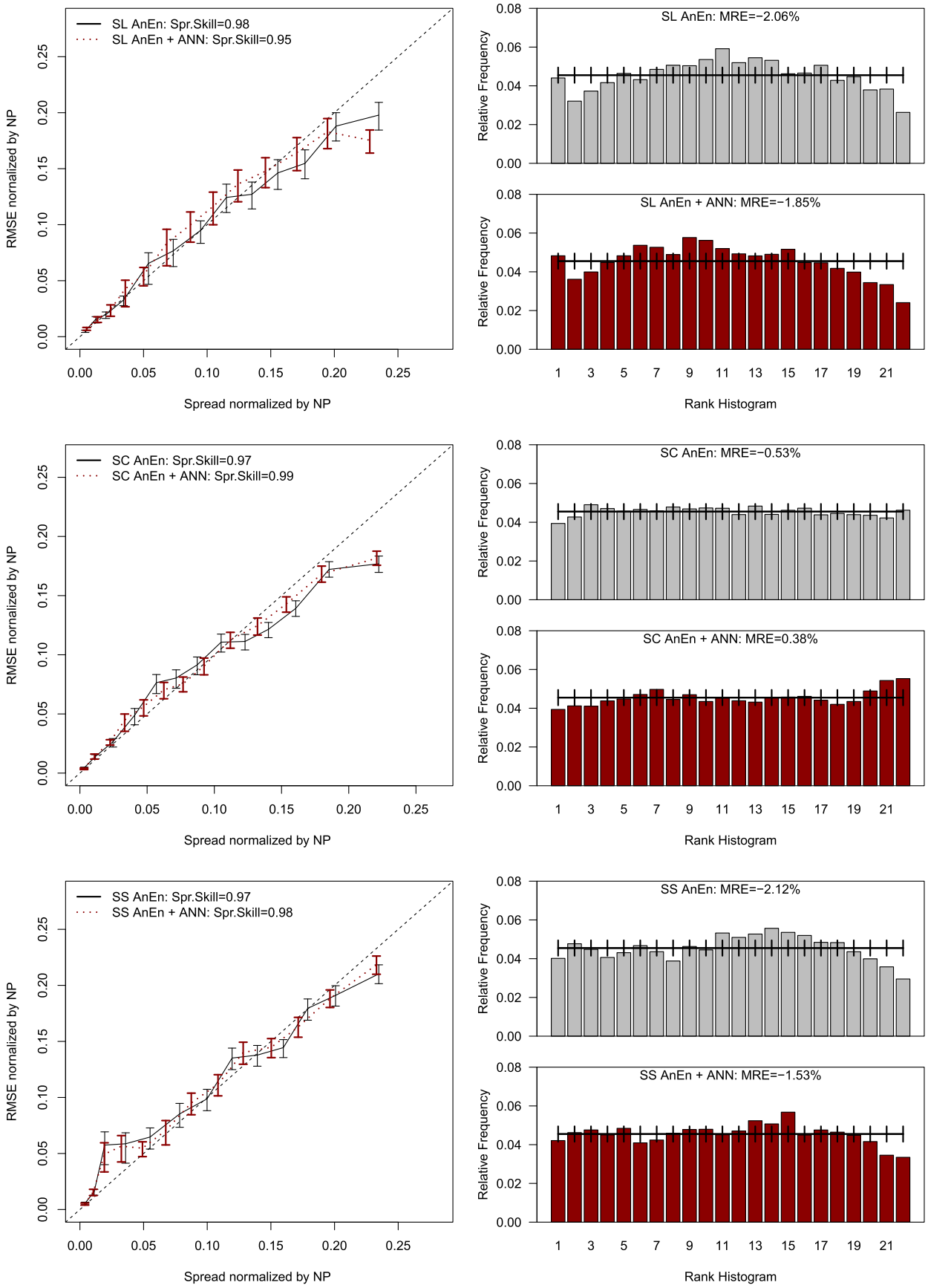

Fig. 7. Summary charts for all three stations depict spread skill score with 95% confidence intervals (left side) and corresponding rank histograms (right) The x-axis on the spread skill denotes the ensemble members. The results are obtained using optimized AnEn metric weights.

figure for SC shows that the combination of the methods exhibits greater ability to quantify the prediction uncertainty than the AnEn by its closeness to the perfect spread-skill line and refined confidence intervals.

The diagrams also indicate that, at both SL and SC, the methods are under dispersive when greater power is measured. SS exhibits this characteristic as well however the spread skill for SS does not indicate this until a higher threshold is reached. Rank histograms are depicted for all three stations on the right hand side of Fig. 7. For rank histograms, an even distribution indicates that there is an equal likelihood for the observation to fall within any of the 21 members, meaning that one member is not biased over another. Overall, statistically consistent forecasts are found at each site with SC and SS exhibiting the most and least reliable behaviors respectively.

The SC station produced the best results and is the largest site where data is quality controlled and PV panels are well maintained. Furthermore, it is in a location that it is not affected by poor air quality, snow or volcanic ash as the other two stations are.

Overall, AnEn performs extremely well both when used in combination with ANN and when used by itself. Without exception, AnEn performs best with optimized weighting for the predictor parameters.

4.3. Computation profiling

The AnEn methodology presents a computational advantage because its efficiency can be enhanced through parallelization, whereas ANN is primarily a sequential code not suited for parallel computation.

The NetBeans profiler is employed to understand the computational behavior of the algorithm [27]. Fig. 8 shows output for the computation of one year of analogs for the SS station. This computation was run on a single node with four cores. Results for the main program and one of the four identical threads are expanded.

About 7% of the execution time is spent in I/O operations (*readDefaultFilesSolar, writeBinaryMatrix4D*) while the remaining 93% is spent on computation of the analogs (*computeAnalogs*). This latter task is fully parallelized and implemented using a ThreadPoolExecutor [29], whose results are shown in pool-1-thread-X where X is the code id and, for this case, ranges between one and four. Because the four pool-threads are identical (reference total execution time for each thread), only pool-1-thread-3 is expanded. It shows that the first five calls have a *Self time* of 0 and are merely a wrapper for the *processCommand*. This method has two main operations:

1. *computeMetric* (84%) for the computation of the similarity metric (Equation (4)).
2. *quickSelect* (16%) for sorting the metric results.

The computation of the metric, which dominates computation in this part of the code, performs three main operations:

1. *Self time* (50%), consists in the three remaining loops of the AnEn algorithm and computes the difference between past (train) and current (test) forecasts for each parameter, for each training day, and for each forecast lead time.
2. *computeSdDim3* (33%) computes the standard deviation for each day of the training day and for each forecast lead time, and it is used as normalization in the metric. This operation cannot be parallelized efficiently, however, it could be omitted entirely if a different normalization scheme is chosen for the metric (see Equation (4)), or it could be computed offline thus eliminating the real time computation.

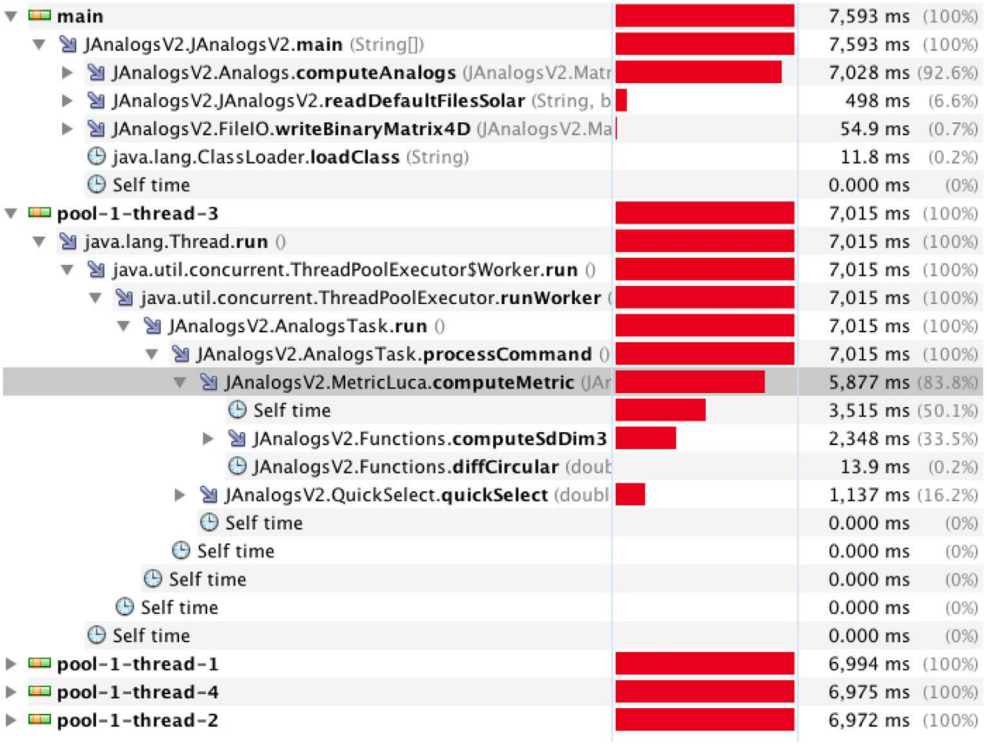

Fig. 8. Output of the NetBeans profiler for the AnEn algorithm for one year of tests for the SS station.

3. *diffCircular* (1%) computes the difference between two variables in the special case they are circular (e.g. solar azimuth and solar elevation).

Sorting of the metric can occur in two different ways: PartialSort and/or QuickSelect [4]. Once the metric between a current forecast and all historical forecasts (training data) is computed, only a small number of k ensemble members corresponding to the best metrics are used. Therefore, it is not necessary to sort all the metrics computed, a task with an average complexity of $O(n \log(n))$ where n is the size of training events (past forecasts). A PartialSort algorithm returns only a list of k smaller metrics and is more computationally efficient with an average complexity of $O(n \log(k))$.

However, it is possible to further increase the computational efficiency by not sorting the list of smaller k values. The QuickSelect algorithm (also called n^{th}_element) can return a list of lowest metrics in linear time, thus achieving a complexity of $O(n)$ [4]. The tradeoff is that the smallest k metrics are not sorted, but only guranteed to be smaller or equal to all remaining metrics. While this does not affect the computation of the analogs, it might cause problems when comparing results with other AnEn implementations that sort the calculated metrics. The AnEn implementation presented in this paper has a toggle option between PartialSort and QuickSelect.

4.4. Scalability of AnEn

The scalability of the AnEn algorithm was tested using an increasing number of cores and nodes on the NCAR Yellowstone supercomputer (see Section 2.4).

4.4.1. Multiple cores on a single node

The first tests are run to determine the scalability of the methods on a single node using the data for the SS station in terms of the time of execution and speedup as function of number of cores. Each Yellowstone main node has 16 cores per node, which can be simulated to 32 cores with hyper-threading enabled. The time of execution $T(n)$ is governed by Amdahl's law:

$$T(n) = T(1)\left(B + \frac{1}{n}(1 - B)\right) \tag{5}$$

where B is the fraction of the algorithm that is strictly serial and n is the number of parallel threads. The speedup for the parallel execution is defined as:

$$S(n) = \frac{1}{T(n)} \tag{6}$$

Amdahl's law states that the maximum theoretical speed up is a function of the portions of code B that cannot be parallelized. Therefore, if 5% of the run time is spent in the sequential portion of the code, as is the case for the AnEn algorithm, the maximum theoretical speedup that can be achieved through parallelization is 20×. This is a theoretical value that does not take into account inefficiency introduced by synchronization and data movements which occur when the code is run in parallel.

Fig. 9 shows the time of execution and speedup for generating analogs for the SS station as a function of the increasing number of cores on a single Yellowstone node. The solid line shows execution of the AnEn on real cores and the dashed line when hyper-threading is in use. The dotted line shows the theoretical performance of a 100% parallelization, and the alternating dash-dot line shows the theoretical performance of a 95% parallelization.

The figures show that the AnEn algorithm scales very well as the number of cores increases from 1 to 16. A speedup of approximately 14× is achieved which is only slightly less than the theoretical 100% and 95% parallelization lines. The speedup decreases dramatically after 16 cores, when hyper-threading is enabled. This is consistent with findings that hyper-threading may not speed up intensive numerical computations (e.g. Ref. [26]).

However, despite the diminishing rate of speedup increase, it is nevertheless advisable to use hyper-threading on NCAR Yellowstone because of the supercomputer's specific charging scheme. In fact, for the regular queue, usage charges are applied on a node and not on a core basis. Hyper-threading still yields a performance speedup, albeit smaller than for real physical nodes, and it is therefore still advisable because it does not cause a charge increase.

4.4.2. Multiple cores on a multiple nodes

The AnEn is run on a massive scale using 4450 available nodes of the NCAR supercomputer which has 32 cores per node (141,140 cores). The missing 86 Yellowstone nodes were not available due to maintenance issues. Specifically, the computation is distributed such that each node computes analogs for an increasing number of stations. The tests over multiple Yellowstone nodes were performed using the synthetic data described in Section 2.3. Specifically, for each station, one year of data was used for training, and three years for the tests.

In order to perform AnEn computations at such a massive scale, data must be properly prepared to increase overall efficiency. The synthetic data reach nearly two terabytes in size and it would be computationally inefficient if all nodes access the files at the same time for two reasons. First, because each node determines analogs only for specific stations and, consequently, each node only needs to read data for the specific stations the node is generating analogs for. Second, Yellowstone uses a shared Glade file system and I/O operations decrease dramatically when a single file is accessed by multiple nodes. To overcome this problem, data for each station are stored in separate files and each node only reads data for the specific station(s) it is processing. Although this action dramatically increases the number of files (from a single file to several thousands), the overall performance increases dramatically.

Fig. 10 (left) shows the histogram for the execution times obtained using all 4450 available nodes. The median execution time fluctuates around 70 s ± 15%. Execution times are multi-modal and show four bell shape curves roughly centered at 60, 66, 72, 76 s. These results can be explained by the specific network topology of the Yellowstone supercomputer in which all nodes are arranged in four clusters, each served by one infiniband switch. The four peaks are due to different network speeds and latency associated with the clustering of the nodes. The first peak is lower because some of the nodes connected to this switch were unavailable at the time of the experiment.

Fig. 10 (right) shows the total execution time (on a logarithmic scale) as a function of a decreasing number of computation nodes. The interval for each point represents variations in execution time due to different node speeds. The dotted line shows the theoretical performance by taking the average execution time of all 4450 nodes and multiplying it by the number of stations for the number of nodes used. Generally, AnEn is able to scale well on Yellowstone and the execution time ranges between a little over a minute to four days, depending if 4450 nodes or a single node are used.

5. Conclusions

This paper presents a methodology based on ANN and the AnEn technique to generate deterministic and probabilistic forecasts of PV power produced using weather and astronomical predictions. The methodology was tested using observed PM data from three PV

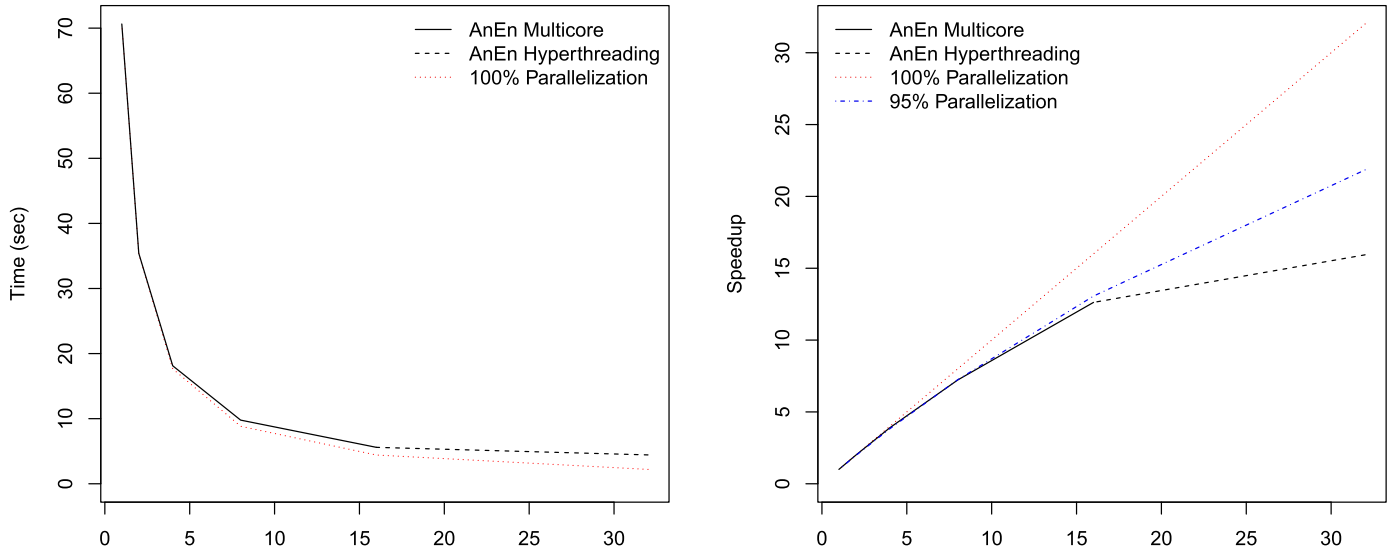

Fig. 9. Time of execution (left) and speedup (right) for generating analogs for the SS station as a function of increasing number of cores on a single Yellowstone node. The solid line is the AnEn on real cores, and dashed line when using hyper threading. The dotted line shows the theoretical performance of a 100% parallelization, and the alternating dash-dot line shows the theoretical performance of a 95% parallelization.

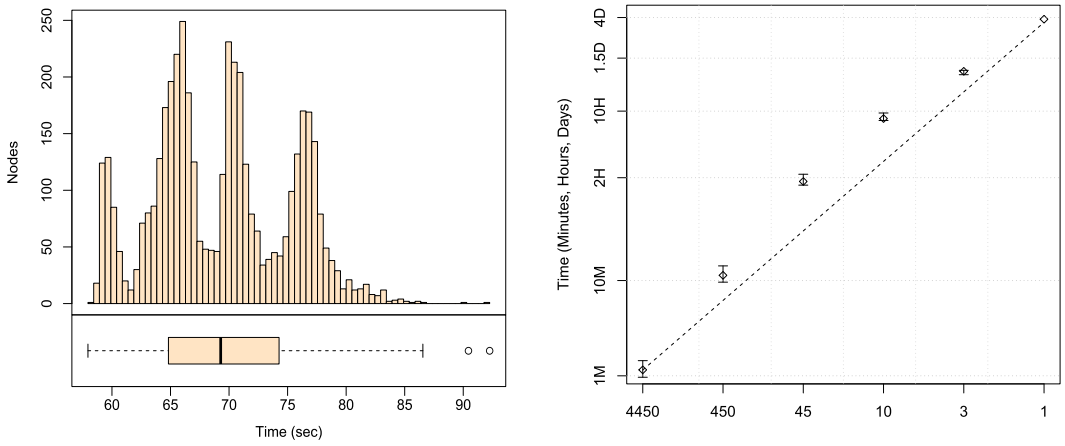

Fig. 10. Histogram of the elapsed time when all 4450 nodes are used (left) and time of execution for generating analogs as a function of decreasing Yellowstone nodes using 32 cores per nodes (right). Results are relative to the SS station. The dotted line shows the theoretical performance of a 100% parallelization.

stations in Italy and weather forecasts from RAMS. Additional experiments are performed using a very large synthetic dataset that simulates thousands of stations to test the computational efficiency and scalability of the proposed solution. Results are analyzed using a series of deterministic and probabilistic statistical measures.

Generally, both the AnEn and ANN achieve comparable small errors when generating deterministic forecasts. The AnEn generates reliable probabilistic forecast with a combination of AnEn + ANN yielding best results. Furthermore, tests show that the proposed solution scales extremely well as the algorithm is particularly suited for parallel computation. Experiments are performed using the NCAR Yellowstone supercomputer with an increasing number of nodes and cores. The elapsed time to compute the tests on all 4450 stations ranges between an average of 70 s when 4450 nodes (141,140 cores) are used, and over four days when one node (32 cores) are used.

The computational efficiency shown is particularly suited for real-time applications of distributed PV power production when forecasts must be quickly run for thousand of stations. The AnEn methodology is shown to scale extremely well for massively parallel applications. Future work will include the comparison of the proposed technique with other methodologies.

Acknowledgments

Work performed under this project has been partially funded by the National Science Foundation (NSF) award, #1639707 and by the Office of Naval Research (ONR) award #N00014-14-1-0208 (PSU #171570).

Appendix A. List of acronyms

AnEn	Analog Ensemble
ANN	Artificial Neural Network
Az	Solar Azimuth
Bias	Deterministic Bias
CC	Cloud Cover
CORR	Correlation
ECMWF	European Center for Medium range Weather Forecasting
EL	Solear Elevation

FLT	Forecast Lead Times
GHI	Global Horizontal Irradiance
IQR	Inter Quantile Range
MTP	Maximum Theoretical Power
MW	MegaWatts
NWP	Numerical Weather Prediction
KW	KiloWatts
PM	Power Measured
PM-ANN	Power Measured estimated by the ANN
PV	PhotoVoltaic
RAMS	Regional Atmospheric Modeling System
RMSE	Root Mean Squared Error
SL	Station in Lombardy
SC	Station in Calabria
SS	Station in Sicily
T2M	Temperature at 2 m
UTC	Coordinated Universal Time

References

[1] S. Alessandrini, L. Delle Monache, S. Sperati, G. Cervone, An analog ensemble for short-term probabilistic solar power forecast, Appl. Energy 157 (1) (2015a) 95–110, http://dx.doi.org/10.1016/j.apenergy.2015.08.011.

[2] S. Alessandrini, L. Delle Monache, S. Sperati, J. Nissen, A novel application of an analog ensemble for short-term wind power forecasting, Renew. Energy 76 (2015b) 768–781.

[3] D. Arent, J. Pless, T. Mai, R. Wiser, M. Hand, S. Baldwin, G. Heath, J. Macknick, M. Bazilian, A. Schlosser, et al., Implications of high renewable electricity penetration in the us for water use, greenhouse gas emissions, land-use, and materials supply, Appl. Energy 123 (2014) 368–377.

[4] M. Austern, Generic Programming and the STL: Using and Extending the C++ Standard Template Library, Addison Wesley, 1998.

[5] R. Banos, F. Manzano-Agugliaro, F. Montoya, C. Gil, A. Alcayde, J. Gómez, Optimization methods applied to renewable and sustainable energy: a review, Renew. Sustain. Energy Rev. 15 (4) (2011) 1753–1766.

[6] S. Becker, B.A. Frew, G.B. Andresen, T. Zeyer, S. Schramm, M. Greiner, M.Z. Jacobson, Features of a fully renewable US electricity system: optimized mixes of wind and solar PV and transmission grid extensions, Energy 72 (2014) 443–458.

[7] M. Chamana, B.H. Chowdhury, Impact of smart inverter control with pv systems on voltage regulators in active distribution networks, in: High-capacity Optical Networks and Emerging/Enabling Technologies (HONET), 2014 11th Annual. IEEE, 2014, pp. 115–119.

[8] F. Davò, S. Alessandrini, S. Sperati, L. Delle Monache, D. Airoldi, M.T. Vespucci, Post-processing techniques and principal component analysis for regional wind power and solar irradiance forecasting, Sol. Energy 134 (2016) 327–338.

[9] L. Delle Monache, T. Nipen, Y. Liu, G. Roux, R. Stull, Kalman filter and analog schemes to postprocess numerical weather predictions, Mon. Weather Rev. 139 (11) (2011) 3554–3570.

[10] L. Delle Monache, F.A. Eckel, D.L. Rife, B. Nagarajan, K. Searight, Probabilistic weather prediction with an analog ensemble, Mon. Weather Rev. 141 (10) (2013) 3498–3516.

[11] P. Eser, A. Singh, N. Chokani, R.S. Abhari, Effect of increased renewables generation on operation of thermal power plants, Appl. Energy 164 (2016) 723–732.

[12] M. Hand, S. Baldwin, E. DeMeo, J. Reilly, T. Mai, D. Arent, G. Porro, M. Meshek, D. Sandor, in: Renewable Electricity Futures Study, vol. 4, National Renewable Energy Laboratory, Golden, CO, 2012. Tech. rep., NREL/TP-6A20–52409.

[13] J.Y. Harrington, The Effects of Radiative and Microphysical Processes on Simulated Warm and Transition Season Arctic Stratus, Ph.D. thesis, Colorado State University, 1997.

[14] S.S. Haykin, S.S. Haykin, S.S. Haykin, S.S. Haykin, Neural Networks and Learning Machines, vol. 3, Pearson Upper Saddle River, NJ, USA, 2009.

[15] I.T. Jolliffe, D.B. Stephenson, Forecast Verification: a Practitioner's Guide in Atmospheric Science, John Wiley & Sons, 2012.

[16] C. Junk, L. Delle Monache, S. Alessandrini, G. Cervone, L. von Bremen, Predictor-weighting strategies for probabilistic wind power forecasting with an analog ensemble, Energy Meteorol. (2015).

[17] M. Kubik, P. Coker, J. Barlow, Increasing thermal plant flexibility in a high renewables power system, Appl. Energy 154 (2015) 102–111.

[18] N.S. Lewis, D.G. Nocera, Powering the planet: chemical challenges in solar energy utilization, Proc. Natl. Acad. Sci. 103 (43) (2006) 15729–15735.

[19] F.J. Lima, F.R. Martins, E.B. Pereira, E. Lorenz, D. Heinemann, Forecast for surface solar irradiance at the brazilian northeastern region using nwp model and artificial neural networks, Renew. Energy 87 (2016) 807–818.

[20] J.P. Lopes, N. Hatziargyriou, J. Mutale, P. Djapic, N. Jenkins, Integrating distributed generation into electric power systems: a review of drivers, challenges and opportunities, Electr. power Syst. Res. 77 (9) (2007) 1189–1203.

[21] R. Margolis, C. Coggeshall, J. Zuboy, Sunshot Vision Study, US Dept. of Energy, 2012.

[22] T. McCandless, S. Haupt, G. Young, A regime-dependent artificial neural network technique for short-range solar irradiance forecasting, Renew. Energy 89 (2016) 351–359.

[23] J. Muller, M. Hildmann, A. Ulbig, G. Andersson, Grid integration costs of fluctuating renewable energy sources, in: Technologies for Sustainability (SusTech), 2014 IEEE Conference on. IEEE, 2014, pp. 15–21.

[24] A.H. Murphy, R.L. Winkler, A general framework for forecast verification, Mon. Weather Rev. 115 (7) (1987) 1330–1338.

[25] S. Russell, P. Norvig, Artificial Intelligence, a Modern Approach, 1995 (Prentice Hall, Upper Saddle River, New Jersey).

[26] S. Saini, J. Chang, H. Jin, Performance evaluation of the intel sandy bridge based nasa pleiades using scientific and engineering applications, in: High Performance Computing Systems. Performance Modeling, Benchmarking and Simulation, Springer, 2014, pp. 25–51.

[27] D. Salter, R. Dantas, Netbeans IDE 8 Cookbook, Packt Publishing Ltd, 2014.

[28] J. Smith, W. Sunderman, R. Dugan, B. Seal, Smart inverter volt/var control functions for high penetration of pv on distribution systems, in: Power Systems Conference and Exposition (PSCE), 2011, pp. 1–6, 2011 IEEE/PES. IEEE.

[29] B. Spell, Lambdas and other java 8 features, in: Pro Java 8 Programming, Springer, 2015, pp. 81–104.

[30] S. Sperati, S. Alessandrini, P. Pinson, G. Kariniotakis, The weather intelligence for renewable energies benchmarking exercise on short-term forecasting of wind and solar power generation, Energies 8 (9) (2015) 9594–9619.

[31] A. Troccoli, J.-J. Morcrette, Skill of direct solar radiation predicted by the ecmwf global atmospheric model over Australia, J. Appl. Meteorol. Climatol. 53 (11) (2014) 2571–2588.

[32] J. Wang, A.J. Conejo, C. Wang, J. Yan, Smart grids, renewable energy integration, and climate change mitigation–future electric energy systems, Appl. Energy 96 (2012) 1–3.

[33] D. Yang, C. Gu, Z. Dong, P. Jirutitijaroen, N. Chen, W.M. Walsh, Solar irradiance forecasting using spatial-temporal covariance structures and time-forward kriging, Renew. Energy 60 (2013) 235–245.