Day-Ahead Solar Irradiation Forecasting Utilizing Gramian Angular Field and Convolutional Long Short-Term Memory

YING-YI HONG [ID][1], **(Senior Member, IEEE), JOHN JOEL F. MARTINEZ** [ID][2], **AND ARNEL C. FAJARDO** [ID][2]

[1]Department of Electrical Engineering, Chung Yuan Christian University, Taoyuan City 320, Taiwan
[2]Technological Institute of the Philippines, Quezon City 938, Philippines

Corresponding author: Ying-Yi Hong (yyhong@ee.cycu.edu.tw)

ABSTRACT The operations of power systems are becoming more challenging on account of the high penetration of renewable power generation, including photovoltaic systems. One method for improving the power system operation involves making accurate forecasts of day-ahead solar irradiation, enabling operators to minimize uncertainty in managing the balance between generation and load. To overcome the limitations of Long Short-term Memory (LSTM) in a one-dimensional forecasting problem, this work proposes a novel method in forecasting solar irradiation by encoding time-series data into images using the Gramian Angular Field and the Convolutional LSTM (ConvLSTM) network. The pre-processed data become a five-dimensional input tensor that is perfectly suitable for ConvLSTM. The ConvLSTM network uses convolution operations in its input-to-state transition and state-to-state transition. The network thus enables time-series forecasting by a feature-rich approach, which ultimately provides competitive forecasting performance despite the use of a small dataset. The proposed method was evaluated in day-ahead solar irradiation forecasting using a univariate dataset of Global Horizontal Irradiation (GHI) data from Fuhai in Taiwan. The proposed method was resampled using 5×2-fold cross-validation, and assessed using the Wilcoxon signed-rank test to determine the statistical significance of the result. It outperformed benchmark methods such as Autoregressive Integrated Moving Average (ARIMA), Convolutional Neural Network cascaded with Long Short-term Memory (CNN-LSTM), and LSTM cascaded with a fully-connected (FC) network.

INDEX TERMS Day-ahead forecasting, ConvLSTM, gramian-angular field, global horizontal irradiation.

NOMENCLATURE

Variables:

b_c	Bias vector of the cell state
b_f	Bias vector at the forget gate
b_i	Bias vector at the input gate
b_o	Bias vector at the output gate
c_t	Cell state vector at time t
c_{t-1}	The previous cell state vector at time $(t\text{-}1)$
f_t	Forget gate activation vector at time t
h_t	Output vector of LSTM cell at time t
h_{t-1}	Previous output vector of LSTM cell at time $(t\text{-}1)$
i_t	Input gate activation vector at time t

n_{GAF}	Number of GAF images after transformation
o_t	Output gate activation vector at time t
$P_{fore(i)}$	Forecasted time-series value
$P_{true(i)}$	True time-series value
r	A vector containing the radius of each element in the normalized time-series vector
r^2	Coefficient of determination
t	Timestamp vector of the normalized time-series
W_{cf}	Weights matrix between the cell state and the forget gate
W_{ci}	Weights matrix between the cell state and the input gate
W_{co}	Weights matrix between the cell state and the output gate
W_{hi}	Weights matrix between the recurrent layer and the input gate

The associate editor coordinating the review of this manuscript and approving it for publication was Fan-Hsun Tseng [ID].

W_{hc}	Weights matrix between the recurrent layer and the cell state
W_{hf}	Weights matrix between the recurrent layer and the forget gate
W_{ho}	Weights matrix between the recurrent layer and the output gate
W_{xc}	Weights matrix between the input layer and the cell state
W_{xf}	Weights matrix between the input layer and forget gate
W_{xi}	Weights matrix between the input layer and the input gate
W_{xo}	Weights matrix between the input layer and the output gate
x	Time-series vector
$x_{fore(n)}$	Input time-series to the forecasting layer
$x'_{fore(nGAF)}$	Transformed image dataset after GAF
$x'_{fore(nGAF+1)}$	Output forecast image from ConvLSTM
$\tilde{x}_l$	Normalized time-series vector
x_t	Input vector to the LSTM cell
$x_{train(n)}$	Input training dataset prior to GAF
$x'_{train(nGAF)}$	Transformed image dataset after GAF
ϕ	Angular transformation of the normalized time-series vector

Functions:

σ	Sigmoid function
tanh	Hyperbolic Tangent function
max	Maximum function
min	Minimum function
var	Variance function

Constants:

N	Number of elements in time-series vector
n	Number of training and test samples
n_s	Sampling window of GAF
n_{str}	The stride of the sampling window of GAF
R	Timestamp regularizing constant factor
α	Significance level

Abbreviations:

ANN	Artificial Neural Networks
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
BAGGING	Bootstrap Aggregating
CNN	Convolutional Neural Network
Conv3D	Convolutional 3D
ConvLSTM	Convolutional Long Short-term Memory
CUDA	Computed Unified Device
DBN	Deep-Belief Network
DFT	Discrete Fourier Transform
DL	Deep Learning
DNN	Deep Neural Network
ECMWF	European Centre for Medium-Range Weather Forecasts
EM	Ensemble Method
ENN	Elmann Neural Network
FC	Fully Connected
FFNN	Feed-Forward Neural Network
GA	Genetic Algorithm
GAF	Gramian Angular Field
GHI	Global Horizontal Irradiation
GPU	Graphics Processing Unit
HISIMI	Historically Similar Mining Model
LSTM	Long Short-term Memory
MAE	Mean Absolute Error
MBD	Mean Bias Deviation
MLP	Multilayer Perceptron
NARX	Nonlinear Autoregressive Network with Exogenous Input
NWP	Numerical Weather Prediction
OBS	Optimal Brain Surgeon
PI	Prediction Interval
PM	Physical Method
PV	Photovoltaic
RBM	Restricted Boltzmann Machine
RMSE	Root-Mean-Square Error
RNN	Recurrent Neural Network
SOFNN	Self-Organizing Fuzzy Neural Network
SVM	Support-Vector Machine
TSSM	Time-Series Statistical Method

I. BACKGROUND

The capital cost for a renewable energy system has been falling over recent years. In fact, the cost of solar PV systems has declined by 75% since 2010, causing a significant increase in investment in the field [1]. Owing to the high penetration of renewable power generation, power system operations are becoming increasingly challenging.

The power generated by PV plants depends on the intermittent energy that is provided by the sun. The variability caused by the daily sun cycle and other meteorological factors gives rise to uncertainties in the determination of this power generation [2]. Power forecasting is critically important in the operation of power systems, as it helps operators dispatch/schedule the power generation from traditional fossil fuels [3].

Forecasting is generally classified by its horizon as long term (one year to ten years ahead), mid-term (one month to one year ahead), short-term (one hour or several hours ahead to one day or one week ahead), and very short term (one minute to several minutes ahead) [4]–[7]. In the PV power problem, long-term forecasting is essential to the planning stage for the development of a solar power plant, whereas mid-term, short-term, and very short-term forecasting is essential for its operation.

Several studies, such as [4], [7]–[9], summarized different methodologies for solar irradiation forecasting that have been proposed within the last decade. Solar forecasting methods are classified according to their general usage and practices [10]:

- PMs mainly use numerical simulation/computation on meteorological data such as atmospheric conditions and ground-based data to predict solar irradiation.
- TSSMs utilize historical data to generate a forecasting model. Generally, TSSM maps the input data to output data using statistical techniques to generate the forecast.
- EMs combine different methods to take advantage of their strengths and overcome the weaknesses of each constituent method.

PMs rely on various meteorological information, such as satellite-derived data or ground-based sensor information, available to provide the forecast. For instance, HISIMI [11] uses historical weather information to find instances with a high probability of similarity in the current weather conditions; a discrete probability function was utilized to predict the spot forecasts according to the resulting probability. Another example of PM is the use of clear sky index [12] provided by satellite images, which were then processed using the combination of ANN and ECMWF to derive intra-day solar prediction. Similarly, sky-imagery [13] was used to capture high-resolution digital images of sky conditions. Maximum correlation technique was used to determine future irradiance values using the images. The result shows a better forecast than satellite images under cloudy conditions.

TSSMs offer a wide variety of methods that includes ANN, SVM, Regressive methods and Markov chains. The popularity of ANN-based methods is continuously increasing as a result of recent advancements in Deep Learning, and the continued development of computer hardware that supports its implementation. For instance, ANN [14] has been used to predict day-ahead solar power generation using input statistical features that were obtained from Global Horizontal Irradiance and temperature data. The method outperformed a Clear Sky physical method and obtained better results in the domain. This method reduced the number of input dimensions to provide better representations with variations in weather conditions. A pre-processing stage was proposed in [2] to eliminate offset and nighttime, de-trend, and normalize the data prior to the FFNN. Shorter training time and a relatively low error rate were achieved for a one-day test period.

DNN can extract more features compared to its shallow counter-part. In one study [15], a DBN with RBM was used to capture the feature-space of large time-series data. The DBN significantly outperformed the MLP, ARIMA, and the SOFNN. A similar study [16] applied LSTM to forecast solar GHI values. It emphasized the recurrent nature of LSTM and its inherent ability to remember dependence in sequences.

EMs are a combination of different models such as linear, non-linear, or combination of both to greatly increase its performance. For example, NARX [17] was improved by applying network pruning methods, such as GA based optimization and OBS, to forecast hourly solar irradiance. Similarly, K-means clustering was used by [18] to single-out outliers from the dataset to improve the training. The cleaned dataset is then fed to an MLP. Ensemble approach [19] was used to optimize and diversify ANN by using methodologies such as Bagging to identify uncertainty in terms of PI. A novel method [20] of using DFT on spatiotemporal GHI data to extract frequency components for an ENN showed promise in the effectiveness of using frequency components for time-series forecasting. Lastly, a hybrid combination of CNN and LSTM has been proposed [21]. The resulting network was trained on ground-based GHI values from a solar farm in Australia. The results revealed that the cascaded network of CNN and LSTM outperformed the CNN and LSTM, separately.

Ultimately, the ANN-based approach is well-suited for solving these types of forecasting problems. In fact, it appears to attract more attention from researchers in numerous fields of study, including solar irradiation forecasting [22]. The concern, however, has been raised [23] that most studies have failed to provide empirical evidence regarding the generality of its forecast results, this points to the varying nature of datasets that greatly affect its performance. Furthermore, it is required to provide a statistically valid number of datasets, where the methods are tested, to strengthen the conclusion regarding comparative statements. This has been a problem since the lack of data is often the case for most studies. In contrast, DNN (such as CNN and LSTM) is a data-hungry network that provides a better fit and forecasting accuracy in exchange for a very large dataset, which is often not available in real-world applications like solar irradiation forecasting.

This paper presents a novel approach to solar irradiation forecasting from fragmentary information by addressing the limitations of data-hungry DL methods. The proposed method uses recently developed techniques called the Gramian Angular Field and Convolutional LSTM.

The GAF [24] is a data-transformation method that converts one-dimensional time-series data into a sequential set of images that retains the temporal features of the original data while adding temporal correlations. The transformed data are then reshaped into a five-dimensional input tensor for ConvLSTM, which is a recurrent encoder-decoder network that addresses the limitations of LSTM in a two-dimensional image sequence prediction problem. The resulting network has the following advantages: (i) reduced tendency to exhibit vanishing gradients, (ii) ease of training, (iii) high forecasting accuracy, (iv) smaller training dataset, and (v) applicability to general time-series forecasting problems. The contributions of this paper are the following:

1. The one-dimensional time-series solar irradiation data are transformed into 2D images using the GAF to enable deep ConvLSTM to perform very well.
2. The proposed ConvLSTM has three layers (ConvLSTM, batch normalization, and Conv3D) to perform day-ahead solar irradiation forecasting using univariate time-series GHI data.
3. The batch sizes for running the proposed deep ConvLSTM using NVIDIA GPU cards with CUDA cores are explored. Both the training time required and the

accuracy of forecasting are inversely proportional to the batch size.

4. GAF-based images of various resolutions are investigated. Both the training time and accuracy of forecasting is proportional to the resolution.

5. Comparative studies are performed using 5×2-fold cross-validation and the Wilcoxon signed-rank sum test to evaluate the performance of the proposed ConvLSTM, which outperforms benchmark methods (ARIMA, CNN-LSTM, and LSTM with fully-connected (FC) network).

The structure of this paper is as follows: Section II explains the basis of the implementation of GAF and ConvLSTM. Section III describes the proposed method. Section IV presents and discusses the results of the simulations. Section V draws the conclusions.

II. BASIS OF GRAMIAN ANGULAR FIELD AND ConvLSTM
A. GRAMIAN ANGULAR FIELD
The field of computer vision has been rapidly expanding due to the recent developments in DL. Recently, it has also come to dominate fields such as natural language processing, sentiment analysis, and activity recognition. However, time-series-based forecasting problems, as presented in this paper, are yet to benefit from these developments. A transformation technique called Gramian Angular Field (GAF) was proposed [25] to represent time-series data as images, enabling the applications of image-based DL methods.

A time-series vector $x = \{x_1, x_2, x_3, \ldots, x_N\}$ with N number of samples is firstly normalized to [0, 1]:

$$\tilde{x_l} = \frac{(x_i - \max(x)) + (x_i + \max(x))}{\max(x) - \min(x)} \quad (1)$$

Each element of the series is then converted to polar coordinates by encoding the value as angular cosine, and the timestamps of each element, divided by a regularizing constant factor, R, as the radius as shown in (2).

$$\begin{cases} \phi = \arccos(\tilde{x_l}), & 0 < \tilde{x_l} < 1 \in x \\ r = t/R, & t \in \mathbb{R} \end{cases} \quad (2)$$

After the rescaled time-series has been transformed, the Gramian field can be set by defining the angular perspective as the trigonometric sum of each point in the interval. The resulting GAF is given by (3). Figure 1 illustrates two example results of GAF transformation.

$$G = \begin{pmatrix} \cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_N) \\ \vdots & \ddots & \vdots \\ \cos(\phi_N + \phi_1) & \cdots & \cos(\phi_N + \phi_N) \end{pmatrix} \quad (3)$$

The transformation retains the temporal dependency between values while providing temporal correlations as a result of the superposition in directions with respect to the time interval. The resulting matrix is bijective. Thus, the inverse function yields an absolute reconstruction of the original data.
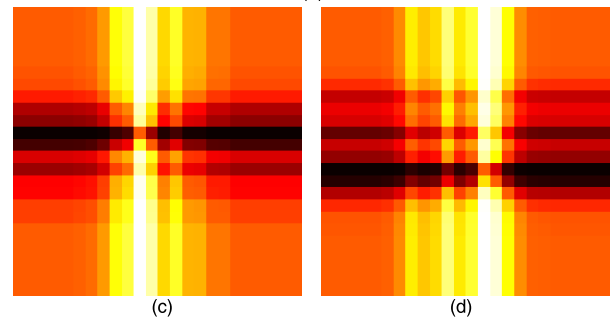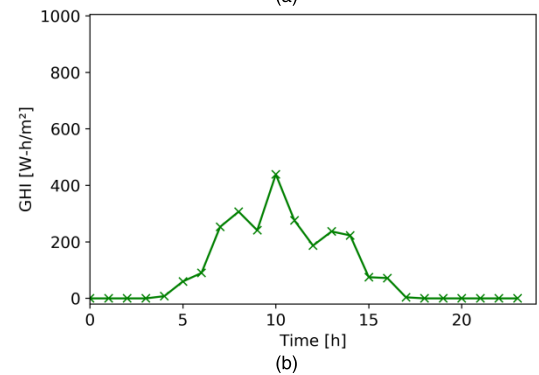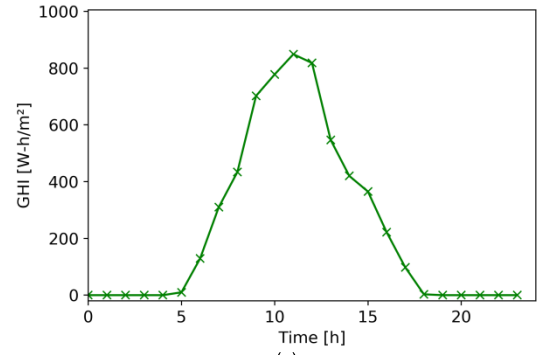

FIGURE 1. (a) High GHI time-series; (b) Low GHI time-series; (c) GAF w.r.t. to high GHI; (d) GAF w.r.t. to low GHI.

B. ConvLSTM
One of the major problems with deeper networks is the tendency of such networks to experience vanishing gradients. The RNN is structured to contain recurring cells in which previous cell states are inputs to the current cell. This process tends to trap the gradient in the network, thereby preventing it from vanishing [26]. The process, however, makes RNN networks difficult to train. LSTM network is a special type of RNN with self-parameterizing gates as a means of determining which information is to be discarded or remembered [27]. The processes that are introduced in LSTM improve the training ability of RNN. The major drawback of the LSTM network for forecasting, however, is the need to convert the input tensor to one-dimensional input, resulting in a loss of spatial information.

A novel approach called ConvLSTM has been proposed [28] as an improvement of LSTM. In ConvLSTM, the Hadamard operators are replaced with convolutional
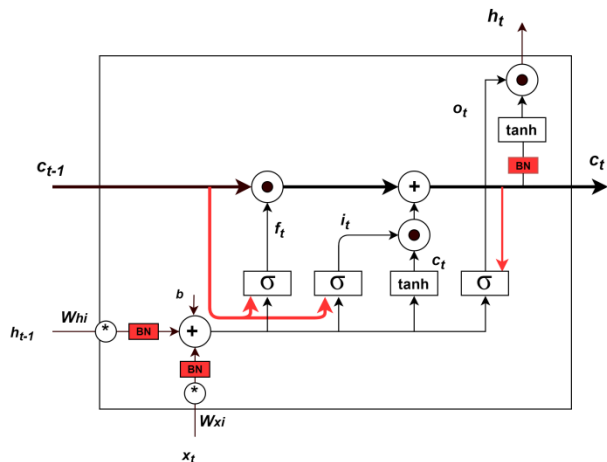
FIGURE 2. A ConvLSTM memory cell.

operators for both the state-to-state transitions and the input-to-state transitions, so the network can accept five-dimensional tensor, as shown in Fig. 2. The functions that are used in LSTM are revised for ConvLSTM, as shown below: where the symbols "$*$" and "o" represents the convolution operation and Hadamard product, respectively.

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \quad (4)$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \quad (5)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (6)$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \circ c_t + b_o) \quad (7)$$

$$h_t = o_t \circ \tanh(c_t) \quad (8)$$

Similar to LSTM, ConvLSTM has a memory cell that acts as an accumulator. Inside it, cells are self-parameterized by gates that control the flow of features. The input gate is represented by (4) decides whether to accept the input or not. The forget gate in (5) turns on if the previous cell state of (6) is to be forgotten. The output gate in (7) decides whether the value in (6) will be relayed to the final state. The final state is obtained by (8).

The ConvLSTM network is normally utilized to solve image sequence forecasting problems. For instance, the precipitation nowcasting problem was undertaken by using ConvLSTM [21], where cloud images acquired from the radar-echo dataset were used to predict its future movements. The resulting coefficient of determination is better than that of using the state-of-the-art ROVER algorithm and LSTM cascaded with a fully-connected network.

III. PROPOSED METHOD

A thorough literature search reveals that ConvLSTM has never been applied to one-dimensional time-series forecasting. Considering the above information, this paper presents a novel approach by applying GAF transformation to time-series data such as GHI for solar irradiation forecasting. The resulting image dataset is then converted to a 5D tensor that will serve as an input to the ConvLSTM network.

The proposed method is comprised of two stages: (i) data preprocessing stage and (ii) encoding-forecasting stage. The data pre-processing stage performs GAF transformation, and then the ConvLSTM encodes and forecasts. The method was implemented using Python with Keras as the backend.

A. DATA PRE-PROCESSING BY GAF

The number of GHI time-series data is N in the pre-processing stage. The process follows the steps that were described in Sec. II.A.

The data pre-processing requires two arguments- a sampling window (n_s) and stride (n_{str}). The sampling window defines the number of samples that are encoded into each image. The stride is the number of skips that is used as the sampling window slides across the entire dataset. The dimension of the resulting image is $n_s \times n_s \times 1$. The number of resulting images is:

$$n_{GAF} = (N/n_{str}) - n_s \quad (9)$$

The data pre-processing stage helps set-up the forecast horizon and normalization of consideration of both spatial and temporal information of the time-series dataset.

B. PROPOSED ConvLSTM

ConvLSTM utilizes three layers, which are: (i) ConvLSTM layer, (ii) batch normalization, and (iii) Conv3D.

The ConvLSTM layer requires a five-dimensional input tensor that contains a set of images with shape (sample, timestep, row, column, channel). The "sample" is the number of elements in the entire dataset; "timestep" denotes the number of frames that constitute a single input to the network; the remaining "row", "column", "channel" are for the uniform dimension of all the images in the dataset. The number of layers and the filter size can also be set for the ConvLSTM layer. An argument *return_sequences*, which is a similar argument in the LSTM layer, takes a Boolean input to define the output shape of the ConvLSTM layer. In this context, *return_sequences* is set to 'true' to produce a five-dimensional output tensor with shape equivalent to (sample, timestep, row, column, filter) containing the features.

As discussed in the previous section, ConvLSTM is similar to an LSTM that contains recurring cells and gate parameters to prevent vanishing gradient and simplify the training process. Convolutional elements are added to maintain the dimensionality of the features throughout the network.

Batch Normalization [29] is applied after every layer of the ConvLSTM. Batch normalization acts as a regularization function for the hidden layers, to improve the training results and, to minimize the covariance shift in the data.

Finally, the Conv3D layer performs sequence-to-sequence mapping. The implementation of this layer is similar to that described for video frame classification [30]. LSTM is used to extract the temporal features from the images. In this context, however, the Conv3D layer accepts input tuples that are compatible with the feature map that is provided by the

ConvLSTM. The Conv3D layer, in turn, operates as the forecasting layer. As in the conventional use of any convolutional layer for sequence-to-sequence mapping [31], the number of filters, kernel size, and activation function are required. The three layers are connected sequentially.

Figure 3 presents a block diagram of the proposed method. The process is two-phase: training and forecasting. In the training phase, a time-series $x_{train(n)}$ data containing n number of samples are fed to the GAF. The resulting image dataset $x'_{train(nGAF)}$ contains images with quantity equal to n_{GAF}. The images are used as input to train the ConvLSTM. The trained network is, then, copied for the forecasting.
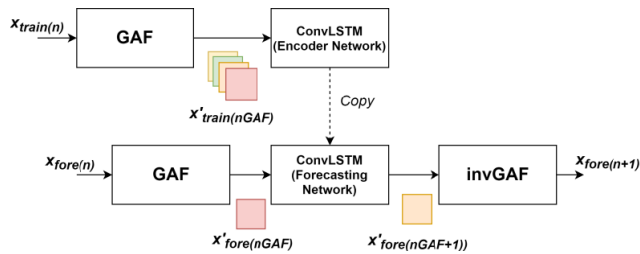

FIGURE 3. Proposed block diagram.

In the forecasting, a test data $x_{fore(n)}$ are needed to forecast the succeeding values, where n should not be less than $2n_s$ The test data are converted to image using GAF. The transformed $x'_{fore(nGAF)}$ is a sequence of images which is directly fed to the forecasting network to generate the forecast image sequence $x'_{fore(nGAF+1)}$. The forecasted image sequence is finally up for final post-processing, which is the inverse-GAF, to reconstruct a time-series version of the result.

C. FORECASTING HORIZON AND BENCHMARK METHODS

The proposed method provides a day-ahead forecast of solar irradiation GHI values. To evaluate the results of ConvLSTM, it will be compared against the following benchmark methods: ARIMA [32], CNN-LSTM [21], and LSTM with a fully-connected (FC) network [33]. These methods were considered since they have been found to perform well on solar irradiance forecasting [21], [26], [33], [34].

The forecast is made using a direct forecasting scheme, wherein 24 values are forecasted at once, and rolled over by one sample at each step across the dataset.

IV. SIMULATION RESULTS

A. DATASET

Day-ahead solar irradiation forecasting is carried out using satellite-derived GHI data from the SolarGIS database [35]. The target location is Fuhai, which is located along the coast of Taiwan, at latitude 24.03 degree and longitude 120.32 degrees, as shown in Fig. 4. With regard to the geographical location, the SolarGIS documentation [36] reports an MBD of ±8%, which is acceptable for the purpose of this study.

FIGURE 4. The geographical location of Fuhai.

The data covers 8760 hours from January 5, 2017, to January 4, 2018. Figure 5 displays the entire dataset. The seasonal classifications are color-coded: green, yellow, red, and blue correspond to spring, summer, autumn, and winter, respectively.

The variation of meteorological values with season increases the complexity of the forecasting problem [26]. Therefore, the dataset must be divided by season [20]. Table 1 presents the divided dataset and the number of samples in each season. Furthermore, the nighttime values are eliminated.

TABLE 1. Summary of number of samples per season.

Seasons	Number of Samples	Number of Samples after removing nighttime
Winter	2160	1080
Spring	2180	1090
Summer	2208	1104
Autumn	2184	1092

Table 2 shows the characteristics of the seasonal time-series data. The spring dataset has the maximum value of GHI while the winter has the lowest. The coefficient of variation implies that the spring dataset has the least variability among the datasets, while the winter has the largest variability rendering it harder to predict. The large differences in the characteristics of the seasons in the Fuhai dataset can also serve as a decent approximation of different weather conditions in other geographical locations.

B. CONFIGURING THE ConvLSTM

To determine the best parameters for the ConvLSTM, a parameter grid search was employed [34]. The key parameters are the following:
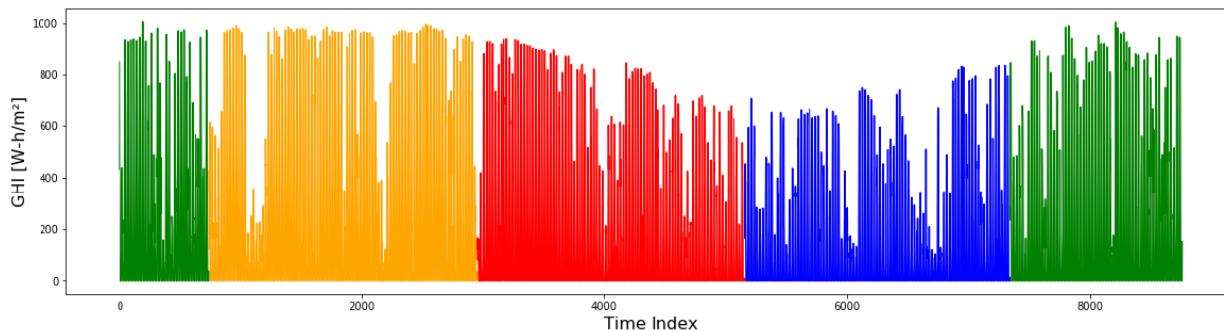
FIGURE 5. Hourly Global Horizontal Solar Irradiations in Fuhai.

TABLE 2. Time-series characteristics of the dataset.

Description	Winter	Spring	Summer	Fall
Mean (Wh/m²)	273.79	434.65	486.41	374.51
Max (Wh/m²)	833	1005	995	940
Standard deviation (Wh/m²)	219.21	290.77	330.81	278.47
Coefficient of Variation	0.8007	0.6690	0.6801	0.7436

- ConvLSTM layers – The number of layers defines the depth of the network. As a rule of thumb, the greater the depth, the better the result. However, the number of data required for such a network increases significantly.
- Batch Size – This controls the stability of the training process. The batch size, when chosen to be large, increases the speed of the training, but yields lower accuracy due to under-fitting. A smaller batch size, on the other hand, generates a better fit of the network at the expense of slow training.
- Filter Size – Filters are a 1D/2D/3D window that is convolved across the images to capture its features. Choosing the filter size is important in detecting features: a large filter size may provide enough receptive fields to fully capture the features. This, however, tends to slow the training process. A smaller filter size, meanwhile, can speed up the training but may overlook the features in the images.
- Number of filters – Each filter generates feature maps of the input image. Increasing the number of filter results in more generated features. The number of filters required for a certain network varies depending on the nature of the images. Choosing an appropriate number of filters greatly improves the convergence of the network.
- Activation Function –This is responsible for the firing of neurons in a neural network, such as CNN. It is important that the activation function is able to learn non-linear relation between inputs and outputs, which are suitable for more complex problems.

- Loss Function – sets up the optimization process to minimize the error between the true value and the predicted value. It has a very important role in the training process - choosing a suitable loss function provides a stable training and prevents the model from diverging.
- Optimizer – The training process focuses on minimizing the result of the loss function by controlling the gradients of the objective. This is accomplished by updating these gradients, depending on the result of the loss function. Choosing the optimizer is critical in the learning of the network, as the unknowns may be updated too slowly or too fast, depending on the nature of the data.

The configuration of the proposed ConvLSTM is determined by simulating the network using different combinations of the parameters mentioned above. The conducted experiments are limited to the functions that are available in the Keras library. Ultimately, the following configurations are attained for the proposed method: One ConvLSTM layer, a Batch Size of 256, 40 ConvLSTM filters, a ConvLSTM kernel size 3×3, five Conv3D filters, *relu* as the activation function [37], *Huber* [38] as the loss function, and *Adam* as optimizer [39].

Specifically, the selection of the proper batch size is explained in this subsection. The hardware accelerator, NVIDIA Geforce RTX 2080, was used in the experiment. The graphics processor contains 4352 CUDA cores, which were utilized for parallel processing. Figure 6 presents the results of the experiments on the batch size.

C. PERFORMANCE METRICS
The results of the simulation are evaluated using the following performance metrics: Mean Absolute Error, Root-Mean-Square Error, and Coefficient of Determination. The performance metrics were chosen as they have been the popular options from the literature [20], [21], [26], [33], [34].

Mean Absolute Error is the average mean between the difference in the true values (P_{true}) and forecasted values (P_{fore}).

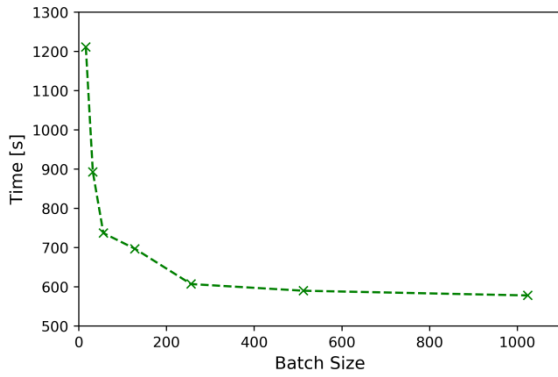$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| P_{fore(i)} - P_{true(i)} \right| \qquad (10)$$

FIGURE 6. Training time decreases as the batch size is increased.

Root-Mean-Square Error measures the global error over the entire prediction set.

$$RMSE = \sqrt{1/N \sum_{i=1}^{N} \left(P_{fore(i)} - P_{true(i)} \right)^2} \qquad (11)$$

Coefficient of Determination captures the correlation between the true values and the forecasted values.

$$r^2 = 1 - \frac{Var(P_{true(i)} - P_{fore(i)})}{Var(P_{fore(i)})} \qquad (12)$$

D. ANALYSIS OF RESOLUTION OF GAF-BASED IMAGE

As described in Sec. III. A, the resolution of GAF-based images depends on two arguments, which are the sampling size n_s and the stride n_{str}. This subsection examines the impacts of n_s and n_{str} on the accuracy of forecasting using the summer dataset before removing the nighttime values.

Increasing the sampling size n_s in GAF will increase the forecast horizon. Figure 7 shows that as n_s is increased (image resolution is reduced), the coefficients of determination r^2 is decreased. Also, the training time increases drastically with image resolution.
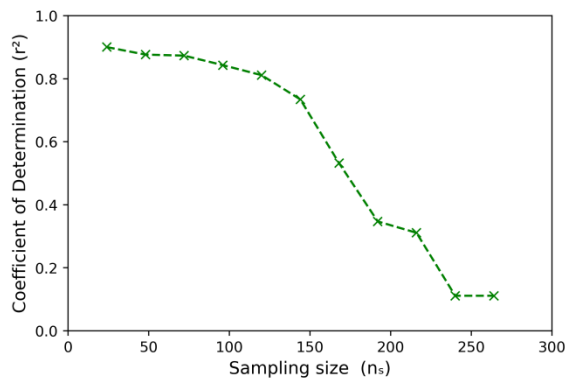

FIGURE 7. Accuracy as the sampling window for GAF increases.

The other parameter that has an impact on the resolution is the stride n_{str}, which is the number of skips as the sampling window slides across the dataset. A smaller stride yields an image with higher resolution. Normally, the stride is set to

unity in order to retain the highest resolution. Figure 8 shows the results that were obtained using strides of 1 to 6. As n_{str} increases, the coefficient of determination r^2 decreases.
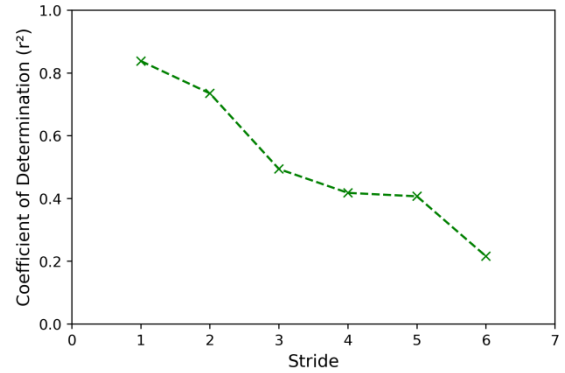

FIGURE 8. Accuracy as the value of stride for GAF increases.

E. COMPARATIVE RESULTS

In this section, the forecasting results are discussed. A resampling method using cross-validation procedure is also explored, followed by the comparative assessment from the result of the statistical analysis.

The resampling method herein is 5×2–fold cross-validation. This method was introduced by Dietterich in [40] and recommended in [41] for machine learning methods. It follows a 2-fold cross-validation approach wherein the dataset is split to 50% training and 50% test sets. Each half of the dataset is set as train and test sets on one run and then switched for the next run. This cycle is repeated five times, as illustrated in Fig. 9. The 5×2-fold cross-validation method is a repeatable procedure as it only allows the samples to appear only once for each trial.
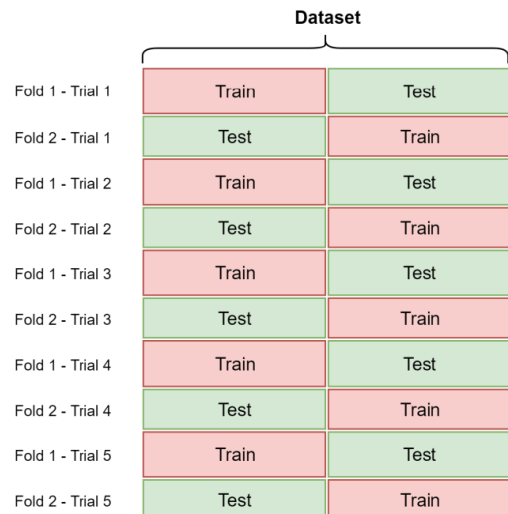

FIGURE 9. 5×2-fold cross-validation.

The resampling methods were run to obtain the performance metrics for every season. The results are summarized
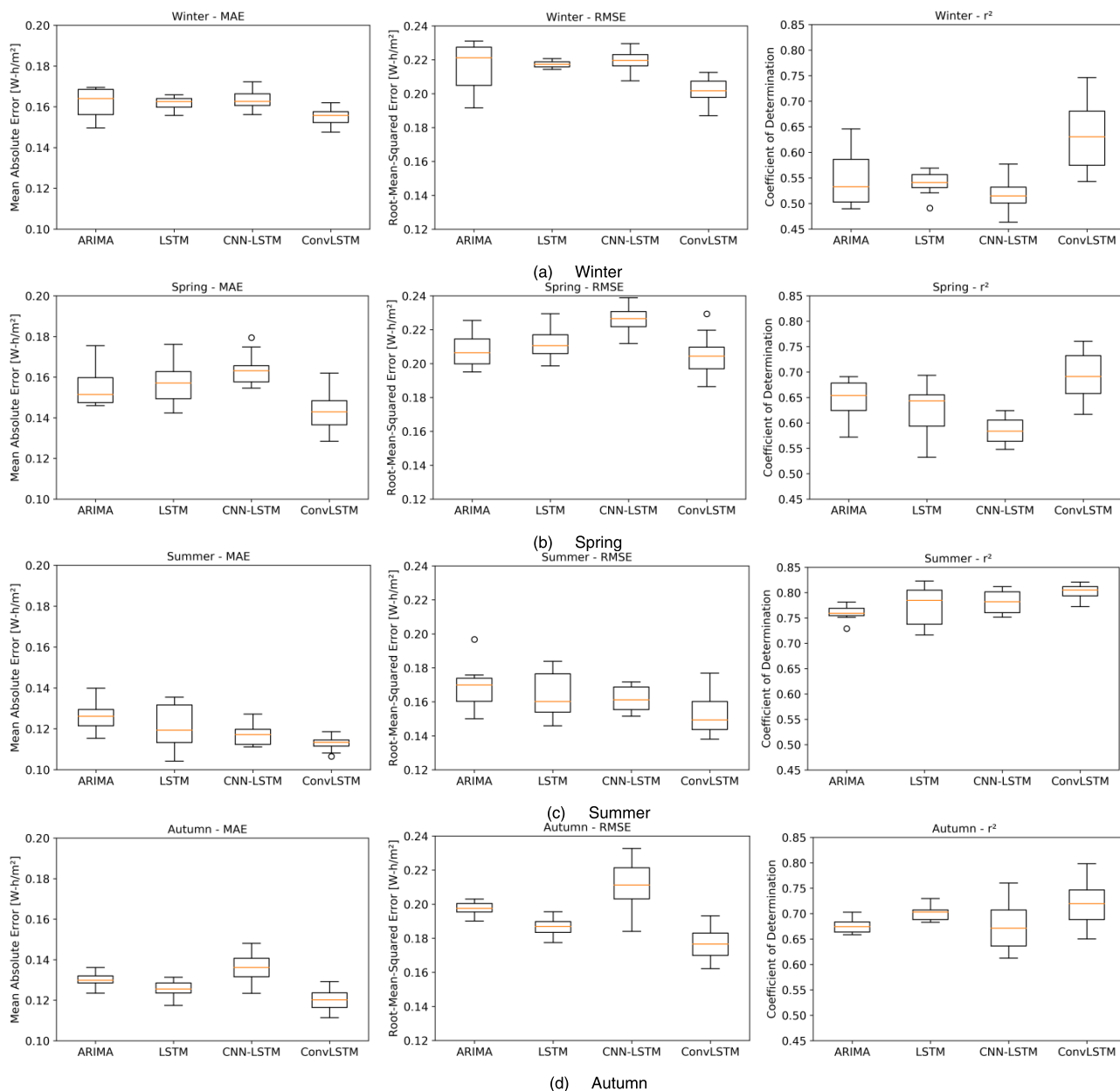
FIGURE 10. Performance metrics results of the methods presented in box and whisker plots.

in Fig. 10 using box and whisker plots. Each plot shows the median value of the performance metrics, as well as the maximum and minimum values, and the distribution of the results using the first quartile and third quartile range. The mean values of the result for each method are summarized in Table 3. The mean values are the basis of comparison for the succeeding discussions. Figure 11 shows a snippet plot of the forecast to show the closeness of each method to the original values.

In the winter forecasting, the results for MAE and RMSE are relatively high compared to other seasons. The lowest *MAE* and *RMSE* are attained by ConvLSTM with 0.1553 and 0.2018, respectively. It is followed by ARIMA

with 0.1617 and 0.2157. The poorest result was found in CNN-LSTM. Regarding the r^2, ConvLSTM has the highest value of 0.6305, followed by ARIMA with 0.549 and CNN-LSTM with the lowest value of 0.5169.

The spring forecasting results indicate that ConvLSTM attained the best performance in *MAE, RMSE,* and r^2. The results are 0.1430, 0.2044, and 0.6944, respectively. ARIMA is the second with 0.1557, 0.2082, and 0.6455, whereas the poorest performance is CNN-LSTM with 0.1635, 0.2266, and 0.5845.

Similarly, the autumn forecasting results show that the ConvLSTM attained the best values of 0.1202, 0.1766, and 0.7192 for *MAE, RMSE,* and r^2, respectively. It is followed

TABLE 3. Overall results of performance per season.

Seasons	Methods	MAE	RMSE	r^2
Winter	ARIMA	0.1617	0.2157	0.5490
	LSTM	0.1617	0.2174	0.5412
	CNN-LSTM	0.1634	0.2196	0.5169
	ConvLSTM	0.1553	0.2018	0.6305
Spring	ARIMA	0.1557	0.2082	0.6455
	LSTM	0.1571	0.2117	0.6276
	CNN-LSTM	0.1635	0.2266	0.5845
	ConvLSTM	0.1430	0.2044	0.6944
Summer	ARIMA	0.1259	0.1683	0.7602
	LSTM	0.1210	0.1641	0.7744
	CNN-LSTM	0.1172	0.1617	0.7820
	ConvLSTM	0.1130	0.1531	0.8015
Autumn	ARIMA	0.1299	0.1976	0.6748
	LSTM	0.1256	0.1870	0.7018
	CNN-LSTM	0.1355	0.2113	0.6758
	ConvLSTM	0.1202	0.1766	0.7192

by LSTM-FC with 0.1256, 0.1870, and 0.7018. The performance of CNN-LSTM is the least with 0.1355, 0.2113, and 0.6758.

The results for summer forecasting are the best among all the seasons. The proposed ConvLSTM gained the best performance with 0.1130, 0.1531, and 0.8015 for *MAE, RMSE,* and r^2, respectively. The CNN-LSTM attained the second-best results, with 0.1172, 0.1617, and 0.7820. The performance of ARIMA is the least with 0.1259, 0.1683, and 0.7602.

Overall, the ConvLSTM attained the best value for all the seasons. This strongly indicates that the proposed ConvL-STM is superior to the benchmark methods. This analysis of the performance values suggests that ARIMA, CNN-LSTM, and LSTM are greatly affected by the changes in the values of the time-series, resulting from their poor generalization. ConvLSTM, on the other hand, outperforms these methods, regardless of the dataset, indicating its better generalization.

The training time for each method was also measured, as shown in Fig. 12. It would appear that the ARIMA, which belongs to the autoregressive group of TSSM, is the fastest to provide the forecasting results. The CNN-LSTM and LSTM have shorter training time than the ConvLSTM. This is to be expected because the ConvLSTM layer, combined with a Conv3D layer, has more unknowns than the other methods. The number of unknowns, however, does not really qualify as a disadvantage when the dataset becomes more complex.

F. WILCOXON SIGNED-RANK TEST

The methods were assessed using a non-parametric test called the Wilcoxon signed-rank test [42]. The test is used to determine whether a pair of observations belongs to different distributions. The test is important to ensure that the difference
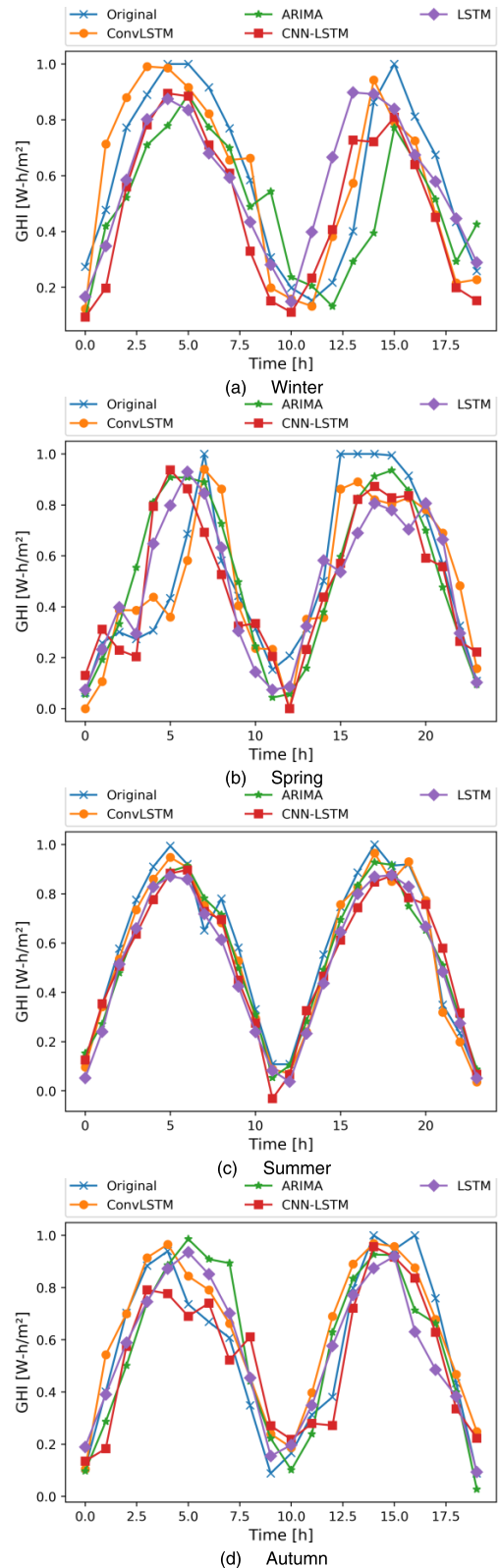

FIGURE 11. Day-ahead forecasting results for (a) Winter, (b) Spring, (c) Summer, and (d) Autumn.

in the results of the methods is statistically significant and does not rely only on statistical chance. The advantage of the Wilcoxon signed-rank test is that it ignores the requirement
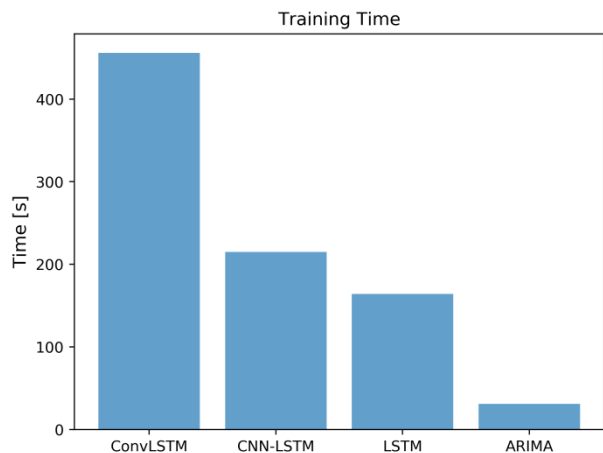
FIGURE 12. The training time for each of the method.

of parametric tests regarding the restriction in the normally of the observations.

Table 4 shows the mean difference between the methods by subtracting the mean of the results (MAE, RMSE, and r^2) of each benchmark method from the mean of the results of ConvLSTM. The negative values of the mean differences in both MAE and RMSE indicate that the ConvLSTM is better. A similar indication is given if the values in the r^2 are positive.

TABLE 4. Mean differences of the benchmark methods from ConvLSTM.

Season	Method	Mean Difference (MAE)	Mean Difference (RMSE)	Mean Difference (r^2)
Winter	ARIMA	-0.0064	-0.0139	0.0815
	LSTM	-0.0064	-0.0155	0.0894
	CNN-LSTM	-0.0080	-0.0178	0.1136
Spring	ARIMA	-0.0127	-0.0038	0.0489
	LSTM	-0.0141	-0.0073	0.0668
	CNN-LSTM	-0.0205	-0.0222	0.1099
Summer	ARIMA	-0.0130	-0.0152	0.0413
	LSTM	-0.0081	-0.0109	0.0272
	CNN-LSTM	-0.0042	-0.0085	0.0195
Autumn	ARIMA	-0.0097	-0.0210	0.0444
	LSTM	-0.0053	-0.0104	0.0175
	CNN-LSTM	-0.0160	-0.0347	0.0435

In the Wilcoxon signed-rank test, the null hypothesis H_o indicates that the paired observations belong to the same distribution and that there is no significant difference between the median of their results. The rejection of the null hypothesis indicates otherwise. Considering the results in Table 4, the rejection of the null hypothesis would suggest that the ConvLSTM is better than the benchmark methods. The null

hypothesis is rejected if the *p-value* is lesser than the significance level α. A widely accepted value of 0.05 for α is used.

ConvLSTM is paired with each of the benchmark methods, as shown in Table 5. The results of the Wilcoxon signed-rank test show that the paired tests for each season yield a *p-value* lesser than the significance level which results in the rejection of the null hypothesis for all.

TABLE 5. Wilcoxon signed-rank test results.

Season	Method	*p-value* (MAE)	*p-value* (RMSE)	*p-value* (r^2)
Winter	ARIMA	0.0469	0.0367	0.0166
	LSTM	0.0069	0.0051	0.0069
	CNN-LSTM	0.0051	0.0069	0.0069
Spring	ARIMA	0.0369	0.0469	0.0367
	LSTM	0.0069	0.0469	0.0069
	CNN-LSTM	0.0051	0.0051	0.0051
Summer	ARIMA	0.0051	0.0051	0.0367
	LSTM	0.0218	0.0069	0.0093
	CNN-LSTM	0.0069	0.0051	0.0093
Autumn	ARIMA	0.005	0.0093	0.0069
	LSTM	0.0367	0.0369	0.0453
	CNN-LSTM	0.0284	0.0472	0.0284

Conclusively, the Wilcoxon signed-rank test shows that the performance of the proposed ConvLSTM is better than the rest of the benchmark methods.

V. CONCLUSION

This work proposed a novel approach for forecasting day-ahead solar irradiation by employing Gramian Angular Field to transform time-series data into feature-rich images. The pre-processed data were then used to train a deep ConvLSTM method. The proposed method was evaluated using a solar irradiation dataset from Taiwan. The obtained results showed that the proposed method achieved a smaller mean absolute error, root-mean-squared error, and better coefficient of determination than ARIMA, LSTM-FC, and CNN-LSTM. It also performed very well despite the small number of samples in the dataset given after the cross-validation.

REFERENCES

[1] *World Energy Investment 2019 Message From the Executive Director*, Int. Energy Agency, Paris, France, 2019.

[2] M. Alanazi and A. Khodaei, "Day-ahead solar forecasting using time series stationarization and feed-forward neural network," in *Proc. 48th North Am. Power Symp. (NAPS)*, 2016, pp. 1–6.

[3] W. Li, C. Yuen, N. Ul Hassan, W. Tushar, C.-K. Wen, K. L. Wood, K. Hu, and X. Liu, "Demand response management for residential smart grid: From theory to practice," *IEEE Access*, vol. 3, pp. 2431–2440, 2015.

[4] S. Sobri, S. Koohi-Kamali, and N. A. Rahim, "Solar photovoltaic generation forecasting methods: A review," *Energy Convers. Manage.*, vol. 156, pp. 459–497, Jan. 2018.

[5] W. Kamrat, "Modeling the structure of local energy markets," *IEEE Comput. Appl. Power*, vol. 14, no. 2, pp. 30–35, Apr. 2001.

[6] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, Oct. 2015.

[7] M. Q. Raza, M. Nadarajah, and C. Ekanayake, "On recent advances in PV output power forecast," *Solar Energy*, vol. 136, pp. 125–144, Oct. 2016.

[8] A. Agüera-Pérez, J. C. Palomares-Salas, J. J. G. De La Rosa, and O. Florencias-Oliveros, "Weather forecasts for microgrid energy management: Review, discussion and recommendations," *Appl. Energy*, vol. 228, pp. 265–278, Oct. 2018.

[9] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. Martinez-de-Pison, and F. Antonanzas-Torres, "Review of photovoltaic power forecasting," *Sol. Energy*, vol. 136, pp. 78–111, Oct. 2016.

[10] A. Nespoli, E. Ogliari, S. Leva, A. Massi Pavan, A. Mellit, V. Lughi, and A. Dolara, "Day-ahead photovoltaic forecasting: A comparison of the most effective techniques," *Energies*, vol. 12, no. 9, p. 1621, Apr. 2019.

[11] C. Monteiro, T. Santos, L. Fernandez-Jimenez, I. Ramirez-Rosado, and M. Terreros-Olarte, "Short-term power forecasting model for photovoltaic plants based on historical similarity," *Energies*, vol. 6, no. 5, pp. 2624–2643, May 2013.

[12] L. M. Aguiar, B. Pereira, P. Lauret, F. Díaz, and M. David, "Combining solar irradiance measurements, satellite-derived data and a numerical weather prediction model to improve intra-day solar forecasting," *Renew. Energy*, vol. 97, pp. 599–610, Nov. 2016.

[13] J. Alonso-Montesinos, F. Batlles, and C. Portillo, "Solar irradiance forecasting at one-minute intervals for different sky conditions using sky camera images," *Energy Convers. Manage.*, vol. 105, pp. 1166–1177, Nov. 2015.

[14] C. S. Ioakimidis, S. Lopez, K. N. Genikomsakis, P. Rycerski, and D. Simic, "Solar production forecasting based on irradiance forecasting using artificial neural networks," in *Proc. IECON Ind. Electron. Conf.*, Nov. 2013, pp. 8121–8126.

[15] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," *Neurocomputing*, vol. 137, pp. 47–56, Aug. 2014.

[16] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, vol. 148, pp. 461–468, Apr. 2018.

[17] A. Di Piazza, M. C. Di Piazza, and G. Vitale, "Solar and wind forecasting by NARX neural networks," *Renew. Energy Environ. Sustain.*, vol. 1, no. 5, p. 39, 2016.

[18] R. Azimi, M. Ghayekhloo, and M. Ghofrani, "A hybrid method based on a new clustering technique and multilayer perceptron neural networks for hourly solar radiation forecasting," *Energy Convers. Manage.*, vol. 118, pp. 331–344, Jun. 2016.

[19] S. Al-Dahidi, O. Ayadi, M. Alrbai, and J. Adeeb, "Ensemble approach of optimized artificial neural networks for solar photovoltaic power prediction," *IEEE Access*, vol. 7, pp. 81741–81758, 2019.

[20] H. Lan, C. Zhang, Y.-Y. Hong, Y. He, and S. Wen, "Day-ahead spatiotemporal solar irradiation forecasting using frequency-based hybrid principal component analysis and neural network," *Appl. Energy*, vol. 247, pp. 389–402, Aug. 2019.

[21] K. Wang, X. Qi, and H. Liu, "A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network," *Appl. Energy*, vol. 251, Oct. 2019, Art. no. 113315.

[22] J. Liu, X. Kong, F. Xia, L. Wang, Q. Qing, I. Lee, and X. Bai, "Artificial Intelligence in the 21st Century," *IEEE Access*, vol. 6, pp. 34403–34421, 2018.

[23] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning forecasting methods: Concerns and ways forward," *PLoS ONE*, vol. 13, no. 3, Mar. 2018, Art. no. e0194889.

[24] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jan. 2015, pp. 3939–3945.

[25] Z. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Proc. AAAI Workshops*, Jan. 2015, pp. 40–46.

[26] D. Lee and K. Kim, "Recurrent neural network-based hourly prediction of photovoltaic power output using meteorological information," *Energies*, vol. 12, no. 2, p. 215, Jan. 2019.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2015, pp. 802–810.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn. (ICML)*, vol. 37, 2015, pp. 448–456.

[30] X. Ouyang, S. Xu, C. Zhang, P. Zhou, Y. Yang, G. Liu, and X. Li, "A 3D-CNN and LSTM based multi-task learning architecture for action recognition," *IEEE Access*, vol. 7, pp. 40757–40770, 2019.

[31] S. Li, J. Zhao, G. Shi, Y. Tan, H. Xu, G. Chen, H. Lan, and Z. Lin, "Chinese grammatical error correction based on convolutional sequence to sequence model," *IEEE Access*, vol. 7, pp. 72905–72913, 2019.

[32] A. K. Chattopadhyay and T. Chattopadhyay, "Time series analysis," in *Statistical Methods for Astronomical Data Analysis* (Springer Series in Astrostatistics). New York, NY, USA: Springer, 2014.

[33] S. Srivastava and S. Lessmann, "A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data," *Sol. Energy*, vol. 162, pp. 232–247, Mar. 2018.

[34] S. Ghimire, R. C. Deo, N. Raj, and J. Mi, "Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms," *Appl. Energy*, vol. 253, Nov. 2019, Art. no. 113541.

[35] (2017). *Solar Radiation Modeling | Solargis*. Accessed: Oct. 21, 2019. [Online]. Available: http://solargis.com/support/methodology/solar-radiation-modeling/

[36] *Validation and Uncertainty of Solar Resource Data | Solargis*. Accessed: Dec. 4, 2019. [Online]. Available: https://solargis.com/docs/accuracy-and-comparisons/overview

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[38] J. Cavazza and V. Murino, "Active regression with adaptive Huber loss," *CoRR*, vol. 1606.01568, pp. 1–14, Jun. 2016.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. 1412.6980, pp. 1–15, Dec. 2014.

[40] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.

[41] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Mach. Learn.*, vol. 52, no. 3, pp. 239–281, 2003.

[42] I. C. A. Oyeka and G. U. Ebuh, "Modified Wilcoxon signed-rank test," *Open J. Stat.*, vol. 2, no. 2, pp. 172–176, 2012.

YING-YI HONG (Senior Member, IEEE) received the B.S.E.E. and M.S.E.E. degrees from the Chung Yuan Christian University (CYCU) and National Cheng Kung University (NCKU), Taiwan, in 1984 and 1986, respectively, and the Ph.D. degree from the Department of E.E., National Tsing-Hua University (NTHU), Taiwan, in December 1990. Sponsored by the Ministry of Education of R.O.C., he conducted research at the Department of E.E., University of Washington, Seattle, WA, USA, from August 1989 to August 1990. He has been with CYCU, since 1991. He was the Dean of the College of Electrical Engineering and Computer Science, CYCU, from 2006 to 2012. He was promoted to be a Distinguished Professor, in 2012, due to his exceptional performance in research, leadership, teamwork, and international collaboration. From 2012 to 2018, he has served as the Secretary-General at CYCU. He is currently the Dean of the Research and Development at CYCU. His areas of interests are power system analysis and AI applications. He received the Outstanding Professor of Electrical Engineering Award from the Chinese Institute of Electrical Engineering (CIEE), Taiwan, in 2006. He was the Chair of the IEEE PES Taipei Chapter, in 2001.
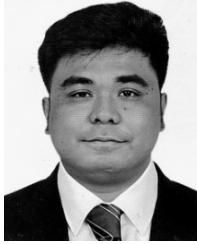
JOHN JOEL F. MARTINEZ received the bachelor's and master's degrees in electronics engineering from the Technological Institute of the Philippines, Quezon City. He currently works as an Assistant Professor with the Electronics Engineering Department. He is conducting collaborative research for his dissertation at Chung Yuan Christian University. He is also involved in collaborative researches and various technopreneurship activities. His research interests are in the field of machine learning, robotics, control systems, remote sensors, the IoT, and renewable energy.

ARNEL C. FAJARDO received the bachelor's degree in electrical engineering from the Mapua Institute of Technology Philippines, in 1991, the master's degree in computer science from De La Salle University, in 1999, and the Ph.D. degree in computer engineering from the Hanbat National University, Daejon, South Korea, in 2014. He is currently the Senior Assistant Vice President and the Head of the Research and Development Department, Manuel L. Quezon University, Philippines. He is also a Professorial Lecturer at the Technological Institute of the Philippines, Quezon City. His research interests include speech recognition, artificial intelligence, and engineering education.

● ● ●