Interpolating high granularity solar generation and load consumption data using super resolution generative adversarial network

Rui Tang[a,b,*], Jonathon Dore[b], Jin Ma[a], Philip H.W. Leong[a]

[a] *School of Electrical and Information Engineering, University of Sydney, Sydney NSW 2006, Australia*
[b] *Solar Analytics Pty Ltd, Sydney NSW 2016, Australia*

Abstract

The vast majority of commonly accessible photovoltaics (PV) generation and load consumption datasets have low temporal resolutions, leading to inaccuracies in the modelling and optimisation of PV-integrated battery systems. This study addresses this problem by proposing an interpolation model based on a super resolution generative adversarial network (SRGAN) that generates 5-minute PV and load power data from 30-minute/hourly temporal resolutions. The proposed approach is validated by two different datasets including large amounts of residential data and compared to an alternative predictive model. The results indicate that the model can adequately capture the targeted data distributions and temporal characteristics with negligible statistical differences from the measured high resolution data. Moreover, it performs consistently across different types of PV/load profiles and on average it results in 0.32% and 0.28% normalised root mean squared errors (NRMSEs) in daily totals of 5-minute PV and load power values when using hourly data as inputs. Under a time-of-use (ToU) tariff, the interpolated 5-minute data leads to 44.7% and 41.7% error reductions compared to using hourly data for estimating electricity costs and battery saving potentials of a PV battery system. Hence, the proposed model can be potentially applied in a battery sizing tool to obtain more accurate sizing results when only low resolution data is available.

Keywords: data interpolation, smart meter, load energy, solar energy

*Corresponding author
Email address: rui.tang@sydney.edu.au (Rui Tang)

Nomenclature

β_1	The momentum value of an Adam optimiser
η	The median of a group of samples
λ	The weighting factor applied for the adversarial loss
$\mathbb{E}$	Expected value operator
θ_D	The discriminator's parameters
θ_G	The generator's parameters
θ_G^*	The optimal parameters of G which minimises its loss function
$\widehat{y_n}$	A secondary value quantitatively computed using one/multiple interpolated high resolution daily power profiles in the test set
D	A discriminator
$F(x)$	Residual mapping using the stacked layers of a residual neural network
G	A generating function that interpolates input low resolution data, which is also referred as a generator
$G(\mathbf{z})$	The generator's output given the input latent noise $\mathbf{z}$
$G(X^{LR})_t$	The interpolated power at a given timestamp t
G_{θ_G}	A generating function parameterised by θ_G, which interpolates input low resolution data
$H(x)$	Output mapping in a residual neural network
I	The number of training iterations
J_A	The adversarial loss component
J_D	The discriminator's cost function
J_G	The generator's cost function
J_{MSE}	The mean squared error (MSE) loss
m	The number of average power values in a low resolution profile
N_{batch}	The size of a mini-batch
N_{test}	The number of test set daily profiles used to compute NRMSE.
N_{train}	The total number of power profiles in the training set
P	The probability distribution of the measured data
$p_{\mathbf{z}}$	A prior data distribution
p_{HR}	The data distribution of high resolution power profiles

p_{LR}	The data distribution of high resolution power profiles	X^{LR}	A low resolution power profile
Q	The probability distribution of the interpolated data	X_n^{HR}	The nth high resolution power profile in the training set
S	A probability space	X_n^{LR}	The nth low resolution power profile in the training set
s	A possible outcome from S	y_{max}	The maximum value of y_n
t	A timestamp	y_{min}	The minimum value of y_n
u	The upsampling factor for interpolation	y_n	A secondary value quantitatively computed using one/multiple measured high resolution daily power profiles in the test set
$V(D,G)$	The value function of GANs		
x	The input to a residual block	M	The number of upsampling blocks in the generator
X^{HR}	A high resolution power profile		
X_t^{HR}	The measured power at a given timestamp t	N	The number of residual blocks in the generator

1. Introduction

There has been an ongoing global trend of smart meter rollouts, driven by the power industry's transition towards smart grids. As a component of the Advanced Metering Infrastructure (AMI), smart meters can monitor and transfer data more frequently and efficiently compared to traditional interval meters [1]. Moreover, with two-way communication between the consumers and utilities, smart metering technology enables and enhances capabilities such as demand response programs, generation and consumption forecasting, optimisation of battery integrated distributed generation and time-varying tariffs [1].

PV generation and load consumption energy/power data, especially in the residential sector, is highly stochastic due to system location, local weather, socio-economic

factors and occupant behaviours. The large amount of high-resolution PV and load data collected by smart meters could be used to enhance the accuracy of data-driven load forecasts [2] and PV forecasts [3] by enabling improved data-derived models. However, high resolution data collection imposes additional costs on the storage, transfer and management of the collected datasets. Moreover, it introduces privacy concerns. As a result, the standard temporal resolutions of smart meter data are still 15-minute or greater [4]. Accessing smart meter data with a finer granularity is still difficult for end-consumers, and most open access smart meter datasets are at 30-minute or hourly temporal resolution [2].

While this level of granularity could be sufficient for billing or deriving aggregated generation or consumption patterns, it may not fully capture the weather transients or consumption spikes. This point is illustrated in the power profiles of Figure 1, where the actual dynamics are not observed with the lower sampling rate. Furthermore, this may lead to inaccuracies in the modelling and optimisation of distributed generation systems. The effects of adopting coarse datasets on distributed generation system optimisation have been investigated in several studies. The impacts of temporal resolution on the optimisation results of micro combined hear and power (CHP) systems are analysed in [5]. The authors found that hourly compared to 5-minute load energy data resulted in up to: 100% overestimation in optimal power generation capacity; 40% overestimation in carbon dioxide emission reduction; and 8% underestimation in lifetime costs. On the other hand, the results derived from 5-minute and 10-minute resolutions have minimal differences (e.g. for half of the cases, the optimal power generation capacities are identical when using 5-minute and 10-minute data). This led to the conclusion that finer resolution than 10-minute provides no improvements and much higher computational costs. Authors in [4] assessed the impacts of granularities of load and PV data on the self-consumption rate and sizing optimisation of PV battery systems. They found that 15-minute resolution is sufficient to obtain a reliable self-consumption rate. However, when estimating optimal battery inverter powers, the relative errors of using 5-minute data are between 5% and 15% for various households, and the errors significantly increased up to 50% when using hourly data. An analysis is performed in [6], where the authors investigated the influences of input data granular-

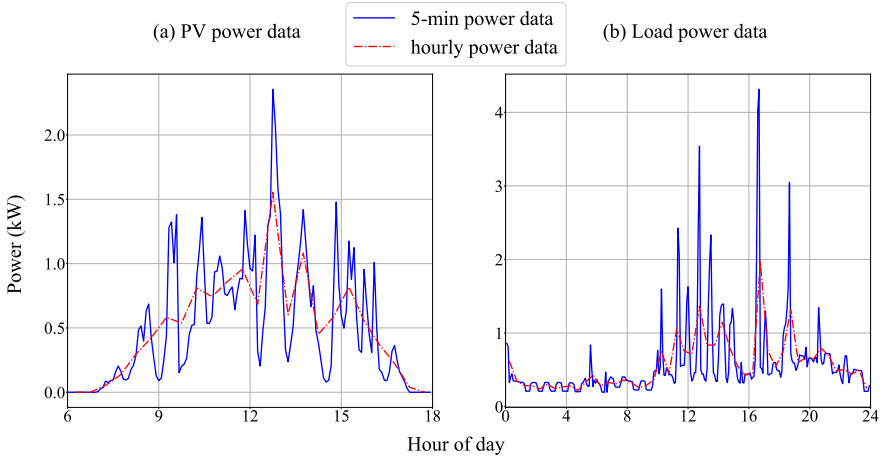

Figure 1: Comparisons between 5-minute and hourly resolutions for (a) a PV power profile on a cloudy day and (b) a residential load power profile. The illustrated power profiles are chosen from the dataset used in this study and described in Section 2.6.

ity on the estimations of saving potential of PV battery systems. The authors found an average of 17% difference between datasets with 1-minute and 30-minute resolution. A similar analysis in [7], evaluating the impacts of data granularity in PV battery optimisation results, found that on average under a flat tariff, the resulting discrepancies between 5-second data and hourly data is 2.9% in estimated electricity costs and 12.6% in battery savings. The authors recommended 5-minute resolution, which achieves a satisfactory balance between accuracy and computational costs.

One potential solution to address the above-mentioned issues is to synthetically interpolate higher resolution smart meter data from lower resolution data. In the existing literature, very few studies have considered interpolating smart meter data. Regarding PV power interpolation, the most relevant studies attempt to interpolate high resolution solar irradiance data from low resolution measurements. In [8], a model was proposed to generate 10-minute irradiance data from hourly measurements where the stochastic component of the 10-minute data is reproduced by randomly generating fluctuations from fitted beta distributions of various classified sky conditions. Some improvements were made in [9], where the authors used the beam clearness index instead of nor-

malised clearness index in [8] to classify sky condition. They also applied a different approach to normalise the deviations of each sky condition and an iterative process to match the daily irradiation of synthetic and measured data. As a result, compared to the original model in [8], the improved model achieved a 40% reduction in relative root mean squared difference between the measured and synthetic irradiance data for daily and hourly irradiation. A similar study [10], made improvements on [8] by using clear sky index instead of normalised clearness index to derive sky conditions with a finer level of categorisation, and applied bootstrapping instead of fitting beta distributions. The study reduced the normalised root mean squared error (NRMSE) from 15% to 1.4-3.0% for hourly sums and from 2-4% to 0.7-1.5% for daily sums of irradiance data. A model is proposed in [11] which can generate 1-minute irradiance data from hourly measurements. The model transforms the high resolution irradiance daily curves to time-normalised daily clearness index series with one year of data used as the training set for each location. Then the model takes in the hourly irradiance data for a day, finds another day with the closest Euclidean distances in the training data and uses its high resolution time-normalised clearness index data and the extraterrestrial horizontal irradiance data of the input day to generate high resolution irradiance data. As a result, the NRMSE in monthly irradiance is 6.2% on average for each location. A study to interpolate 2-minute wind speed data from hourly measured data is proposed in [12]. The model generates random values of the gust, ramp and noise components of wind speed data from the uniform data distributions of these components. The measured input data is used to set upper wind speed limits for the gust and ramp components, respectively. The method led to 13.79%, 62% and 18.3% relative errors in the average, maximum and minimum wind speeds. Regarding load data, the nearest related study sought to improve load disaggregation accuracy by interpolating high frequency load data (100/1000 Hz) from lower frequency load data (10/100 Hz) using a convolutional neural network (CNN) trained with a mean squared error (MSE) loss function [13].

There are no existing studies that interpolate high resolution PV/load power or energy data from commonly accessible (e.g. 30-minute/hourly) coarse smart meter data to the best of the authors' knowledge. Although some approaches have been developed to interpolate irradiance, wind speed or high frequency load data, the practicability of

these approaches are questionable for interpolating residential PV and load power data due to four main reasons:

1. Approaches in [8–10] all require an indicator based on irradiance measurements to classify sky conditions. As irradiance measurements are difficult to obtain, especially for residential sites, it is challenging to apply these approaches in real-time.

2. The method used in [11] requires a year of high resolution data of the same location as the training set, which is not easy to obtain for most households.

3. Residential PV and load power profiles are highly stochastic and cannot be modelled with a simple probability distribution such as the beta distribution used in [8] or a uniform distribution applied in [12].

4. Interpolating data using mean squared error (MSE) as a loss function has been proposed in the literature (e.g. in [13]). MSE has been reported to encourage finding the averages of possible solutions [14]. This means the interpolation model using MSE may generate the averages of power values instead of various correct power trajectories, leading to overly smooth interpolated power curves.

Motivated by these facts, a deep learning model is proposed in this study to interpolate high resolution PV generation and load consumption power data from commonly accessible smart meter measurements. The model is inspired by the super resolution generative adversarial network (SRGAN) work proposed in [14], which sets a new state-of-art for image super-resolution. The aims of this study are as follows.

1. Interpolate 5-minute PV/load power data visually and statistically akin to the measured 5-minute power data from 30-minute/hourly resolution. The reasons for setting the targeted temporal resolution to 5-minute are two-fold: 1. 5-minute is sufficient for applications investigated in [4–6] and recommended in [7], achieving a good balance between accuracy and computational costs for optimisation of a PV battery system; 2. Although the model can be easily adjusted to generate data with higher resolution, the amount of higher resolution data required to fit the model is also larger and may not be easily accessible in practice.

2. Evaluate the SRGAN model performance against various PV/load profiles, alternative approaches and different datasets.

3. Validate the interpolated data using an end-use application. The SRGAN interpolated data is adopted in a residential PV battery optimisation model to address the inaccuracies in the optimised results caused by using coarse data.

To the best of the authors' knowledge, the SRGAN model has not been applied to interpolate time series data, and its applications have been focused on interpolating high resolution images (e.g. medical image interpolation in [15] and food image interpolation in [16]). The reasons for developing our approach based on the original SRGAN work are three-fold:

1. A deep learning model uses stacked neural network layers. Hierarchical levels of temporal dependencies can be captured from the input time series data without any hand-crafted preprocessing steps such as separating the dataset into different sky conditions [8–10]. Only the normalisation of the input data is required.

2. The original SRGAN work generates 16 pixels from a single input pixel [14]. This upsampling factor is similar to our applications, especially for interpolating 5-minute data from hourly measured data, making it a good candidate.

3. Generative adversarial networks (GAN), which forms the basis of the SRGAN model, has already been applied for synthesising PV power data [17] and load power data [18] from random noises. They have been reported to have superior performances compared to alternative methods.

Hence, the contributions of this work are:

1. The first SRGAN based model to produce 5-minute average PV generation and load power data from 30-minute/hourly average PV and load power measurements. Several improvements to the original SRGAN model [14] are proposed to make it more suitable for our application, which are described with details in Section 2.3. This work also shows that using only MSE for the loss function

makes the interpolated power profiles overly smooth, and it is necessary to optimise an adversarial loss component to make the interpolated data realistic. It should be noted that the proposed model can also be used to generate 5-minute PV/load energy data, which can be directly converted from 5-minute average power.

2. The source code for the implementation, together with the trained parameters of the proposed SRGAN model are available online at https://github.com/tomtrac/SRGAN_power_data_generation. This allows others to: easily apply our SRGAN model to their own datasets; apply our trained network directly to generate 5-minute data from 30-minute/hourly measurements; and to compare their results with this work.

3. We demonstrate that additional categorical information regarding the daily power profiles, such as the month of a solar profile or the clustering label of a load profile, can further improve the interpolation results.

4. We validate the synthetically interpolated high temporal precision power data in a PV integrated battery optimisation model, which for the first time, addresses the issue of applying coarse PV and load data in modelling residential PV battery systems. The optimisation results are evaluated by a large real-time dataset and an alternative interpolation model.

The remainder of the paper is organised as follows: Section 2 presents the methodology in this work, including a problem formulation and the proposed model; Section 3 presents detailed evaluations of the interpolated data; Section 4 concludes the study and proposes some future work.

2. Methodology

GAN was first proposed in [19] as a generative model and it is capable of generating high quality synthetic data from random input noise. GANs have been responsible for many state-of-art advances in data synthesis and forms the basis for SRGANs. In this section, the problem formulation is first introduced. Then, the concept of GAN, the

loss function, architecture and training process of the proposed SRGAN model are described.

2.1. Problem Formulation

In the context of our work, interpolation aims to estimate a high temporal resolution measured average power generation/consumption profile X^{HR} from its lower resolution version X^{LR}. X^{LR} is essentially a time series with m average power values. u is the upsampling factor (i.e. u power values are interpolated from a single value in X^{LR}). As a result, X^{HR} contains $u \times m$ time-indexed values.

Historical high resolution data collected from multiple sites are used in the training set to train a generating function G_{θ_G} parameterised by θ_G. The training task can be defined as finding θ_G^* in Eq. (1):

$$\theta_G^* = \arg\min_{\theta_G} \frac{1}{N_{train}} \sum_{n=1}^{N_{train}} J_G(G_{\theta_G}(X_n^{LR}), X_n^{HR}), \tag{1}$$

where θ_G^* are the optimal parameters which minimise G_{θ_G}'s loss function J_G described in Section 2.4; X_n^{HR} and X_n^{LR} respectively denote a single high resolution and a low resolution PV/load power profile in the training set; N_{train} is the total number of power profiles in the training set and $n = 1, ..., N_{train}$.

2.2. Generative Adversarial Networks

GANs involve two artificial neural networks, a generator G and a discriminator D. G takes in a random variable $\mathbf{z}$ sampled from a simple prior distribution $p_{\mathbf{z}}$ (e.g. a Gaussian distribution), and generates a synthetic sample (e.g. an image/a time series). D is implemented as a classifier to distinguish whether the input image/time series is synthetic or real data.

For the interpolation task considered in this study, the structure of the original GAN is adjusted to a SRGAN shown in Figure 2. Instead of latent noise, low resolution power profiles are input to the generator to generate high resolution power profiles; then, the discriminator's task is to distinguish synthetically interpolated power profiles from real high resolution power profiles.

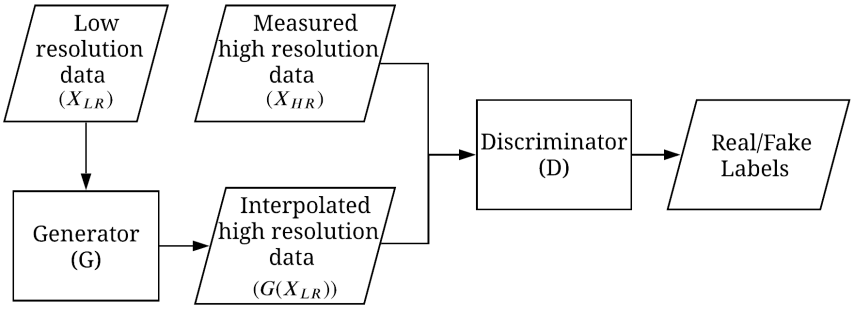

Figure 2: The structure of the SRGAN, which shows the inputs and outputs of the discriminator and the generator.

2.3. Improvements to the original SRGAN model

The original SRGAN model aims to interpolate high resolution images with good perceptual quality. To make it capable of interpolating high resolution power data, the following modifications are made:

1. The original model uses a content loss component for the generator, computed using a pre-trained image classification model. The interpolated and real images are fed into the pre-trained model, and the content loss is the Euclidean distance between the extracted features of real and interpolated images. We do not have a pre-trained model, nor do we aim for perceptual quality. Hence, we use MSE as a replacement for content loss.

2. The original work upsamples the input data using sub-pixel convolutional layers [20]. Unfortunately, this approach sometimes causes unrealistic artifacts in the interpolated data [21]. Instead, we have adopted the resize convolution approach proposed in [22].

3. Fewer convolutional layers are applied in the generator and discriminator, which we found sufficient for interpolating power profiles with minimal computational cost. The final activation function in the generator is changed from a hyperbolic tangent function in [14] to a sigmoid, so the output normalised numerical range

is between 0 and 1.

4. Some adjustments have been made to the hyper-parameters of model training to make the approach more suitable for the intended applications. The details are described in Section 2.6.

2.4. Loss Function

During the model training process, the discriminator aims to maximise the probability of assigning the correct labels to measured high resolution power profiles (X^{HR}) and interpolated profiles ($G(X^{LR})$). This is done by minimising the cross-entropy cost $J_D(\theta_D, \theta_G)$ shown in Eq. (2):

$$J_D(\theta_D, \theta_G) = -\mathbb{E}_{X^{HR}\sim p_{HR}}[\log D(X^{HR})] - \mathbb{E}_{X^{LR}\sim p_{LR}}[\log(1 - D(G(X^{LR})))] \tag{2}$$

Where θ_D, θ_G are the parameters of the discriminator and generator, p_{HR} and p_{LR} represent the data distributions of the high and low resolution power profiles respectively and $\mathbb{E}$ is the expected value operator. $\mathbb{E}_{X^{HR}\sim p_{HR}}[\log D(X^{HR})]$ is the expected value over all measured profiles of the discriminator's estimated log probability that measured data is real and $\mathbb{E}_{X^{LR}\sim p_{LR}}[\log(1 - D(G(X^{LR})))]$ is the expected value of the discriminator's estimated log probability that interpolated data is not real. Minus signs are added to these two components to combine them into a cross-entropy cost.

The loss function of the generator, on the other hand, has two main components: one is the MSE between the interpolated and measured high resolution data (shown in Eq. (3)) which shall be minimised to ensure the reconstructed power values are close to the measured high resolution values; the other loss is the adversarial loss (J_A shown in Eq. (4)), where $\mathbb{E}_{X^{LR}\sim p_{LR}}[\log(D(G(X^{LR})))]$ is the expected value of the discriminator's estimated log probability that interpolated data is real. A minus sign is added, which makes this term an adversarial loss component to be minimised during training. The reasons for including the adversarial loss component are two-fold: 1. minimising just the MSE encourages finding the averages of the plausible interpolation solutions, and this creates overly-smooth interpolation results that are not realistic [14]. This issue is found in image super-resolution; 2. adding adversarial loss encourages the generator to

capture high resolution uncertainties to make the interpolated profiles realistic enough to fool the discriminator.

$$J_{MSE}(\theta_G) = \frac{1}{u \times m} \sum_{t=1}^{u \times m} (X_t^{HR} - G(X^{LR})_t)^2 \tag{3}$$

Where J_{MSE} is the MSE loss, t represents a timestamp, X_t^{HR} and $G(X^{LR})_t$ are the corresponding power values in the measured and interpolated high resolution power profile, u is the upsampling factor and m the number of average power values in X^{LR}.

$$J_A(\theta_D, \theta_G) = -\mathbb{E}_{X^{LR} \sim p_{LR}}[\log(D(G(X^{LR})))] \tag{4}$$

The combined loss J_G of the generator is the weighted sum of J_{MSE} and J_A:

$$J_G = J_{MSE} + \lambda \times J_A \tag{5}$$

Where λ is the weighting factor, which is an arbitrary constant applied for the adversarial loss. As J_G and J_D are inversely correlated, they can combine and form a min-max objective V for both functions:

$$\begin{aligned} \min_G \max_D V(D,G) = \frac{1}{u \times m} \sum_{t=1}^{u \times m} (X_t^{HR} - G(X^{LR})_t)^2 + \\ \lambda \times (\mathbb{E}_{X^{HR} \sim p_{HR}}[\log D(X^{HR})] + \mathbb{E}_{X^{LR} \sim p_{LR}}[\log(1 - D(G(X^{LR})))]) \end{aligned} \tag{6}$$

2.5. Model Architecture

The design of the proposed model architecture is inspired by the original SRGAN work in [14], where both the generator and discriminator are implemented as a deep convolutional neural network (CNN). In contrast to a traditional feedforward neural network whose layers are all fully-connected (also referred to as dense) layers, a CNN contains some convolutional layers. A layer in a neural network is a collection of neurons, where each neuron receives some inputs, performs dot products on them using a set of trainable weights and processes the weighted sum of the dot products through the assigned activation function to add non-linearity to a neural network. Then the layer outputs are transferred to other layers or considered as the final outputs of the neural network if it is the output layer.

In a dense layer, each neuron is independently connected to all the neurons in the previous layer. On the other hand, a typical convolutional layer receives a three-dimensional (3D) input, and its neurons are only connected to a small group of neurons of the previous layer. The neurons in a convolutional layer are also arranged in a 3D manner with dimensions defined as height, width and depth. The connected region is referred to as a receptive field, and its corresponding array of weights is called a filter. Figure 3 shows how a filter is applied in a convolutional layer with the depth set to 1 for a more accessible illustration in two dimensions (height, width). Some zeros are added to the boundary of the input matrix to preserve the output sizes when multiple convolutional layers are applied. Then the same filter is applied multiple times along the height and width, which performs dot products and output a feature map. Generally multiple filters are used in a convolutional layer, resulting in multiple feature maps stacked together to produce the output volume of the layer. The stride in Figure 3 indicates the distance between two consecutive reception fields. It is one of the four main hyper-parameters of a convolutional layer, along with the number of filters, filter size and padding amount.

The partial connections of neurons in a convolutional layer have significantly reduced the number of parameters and computational costs in a neural network, so does the risk of over-fitting. By applying multiple filters and stacked convolutional layers, hierarchical levels of temporal dependencies/features can be captured from the input image/time series without requiring extra hand-crafted preprocessing steps other than the normalisation of the input data. As a result, deep CNNs have achieved many breakthroughs in the domains of image recognition [23] and restoration [24], speech recognition [25] and natural language processing [26].

However, as the number of layers for a CNN keeps increasing to a certain extent, often the model accuracy gets saturated and decreases rapidly. This degradation in model performance is addressed by the residual neural network (ResNet) proposed in [27], which includes residual blocks that add skip connections along with the normal data flows in a deep CNN. Hence, in this study, residual blocks are applied for the generator to allow extra useful information to flow from the input data and at the same time avoid the degradation issue of very deep CNNs. The architecture of the adopted residual

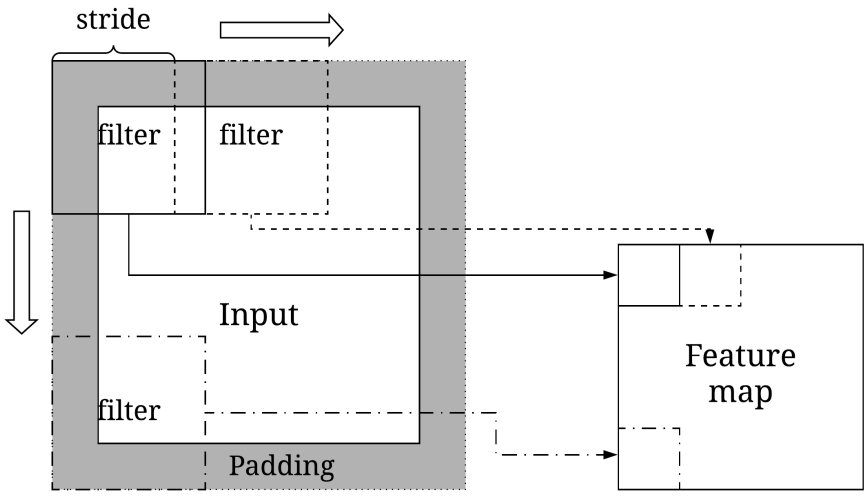

Figure 3: The procedure of applying a filter in a convolutional layer, the depth is set to 1 for a more accessible illustration.

block is shown in Figure 4(a), which follows the work proposed in [14]. Each residual block includes two convolutional layers followed by batch normalisation which standardises the previous layers' outputs to stabilise and accelerate the training process [28]. Parametric rectified linear unit (PReLU) [29] is used as the activation function.

Consider the input of the residual block as x, and the desired output of the residual block is $H(x)$. As shown in Figure 4(a), where a skip/identity connection is added to a stacked CNN, the input x is copied and added to the output of the stacked layers. This means that instead of fitting these in-between layers directly to produce $H(x)$, another mapping $F(x)$ called a residual mapping is used where $F(x) = H(x) - x$. Hence $H(x)$ is recast into $F(x) + x$. The skip connections allow information to flow between layers easily without any transformations and help the later layers utilise the information from the original input layer or previous layers. Moreover, the skip connections enable identity mappings (the output is the same as the input), which is difficult to approximate for traditional non-linear deep CNNs. Hence, if the optimal layer mapping is close to an identity mapping, this skip structure makes it easier to find the optimal layer pa-

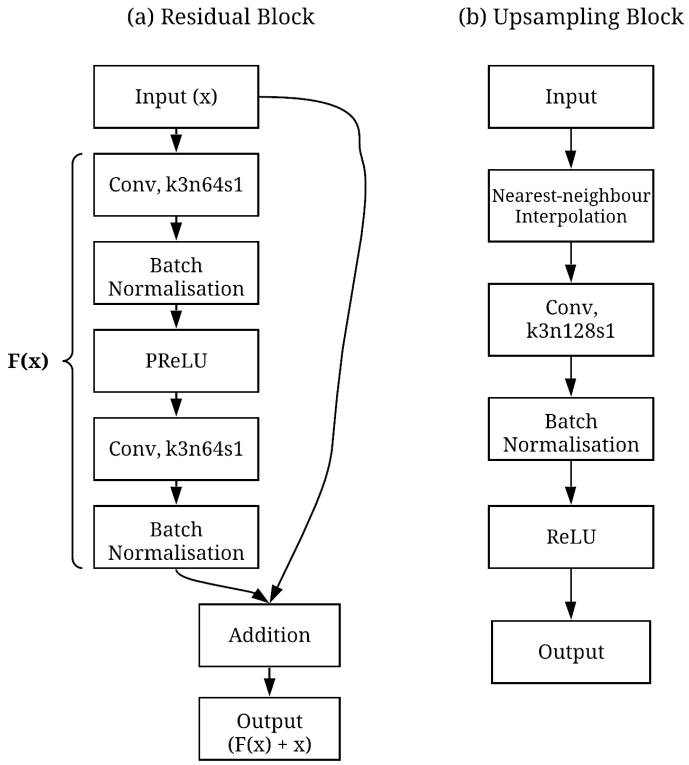

Figure 4: The adopted (a) residual block and (b) upsampling block for the proposed SRGAN model. ”Conv” refers to a convolutional layer, the numbers after ”k”, ”n” and ”s” respectively stand for the filter size, number of filters and stride amount of the convolutional layer(e.g. k3n64s1 indicates that the convolutional layer has a filter size of 3×3, 64 filters and a stride of 1).

rameters. To increase the resolution of the input data, the upsampling block shown in Figure 4(b) is applied, which is inspired by the work in [22] and includes an initial nearest-neighbour interpolation, a convolutional layer with batch normalisation and a rectified linear unit (ReLU) activation function [30].

The model architecture of the generator is shown in Figure 5(a), which includes N residual blocks and M upsampling blocks.

For the discriminator, as fewer convolutional layers are adopted, no residual blocks are required. The design shown in Figure 5(b) simply follows the guidelines proposed in [31] for deep convolutional generative neural networks (DCGAN). Leaky ReLU [32] is applied as the activation function, and batch normalisation is also applied.

2.6. Dataset and Model Training

The dataset used in the study includes 5-minute average PV generation and load consumption data of 2925 Australian PV households, collected by Solar Analytics [33] using Wattwatcher energy monitors [34] for the period between January 2017 and December 2017. 5-minute data is then resampled into 30-minute and hourly datasets. The PV and load power data are normalised by the household's PV system size and peak load before fitting the SRGAN model. This makes sure the numerical range is between 0 and 1. The power data from 80% of the households is used to train the SRGAN model, 10% as the test set to evaluate the performance of the model and the remaining 10% is used as the validation set to select the optimal model hyper-parameters such as the number of training iterations, numbers of residual blocks and upsampling blocks in the generator.

Both the generator and the discriminator are trained by backpropagation [35] with training steps described in Appendix A to update the parameters of both functions. Model training is performed on a desktop with an Nvidia GeForce RTX 2070 GPU, an Intel Core i7-8700K CPU and 32 GB of RAM, using Keras [36] and Tensorflow [37] as the deep learning packages.

Separate models are trained for interpolating 30-minute and hourly PV and load data, then some tuning of the model hyper-parameters is done using the validation set and the Jensen-Shannon divergence (JSD) [38] defined in Eq. (7) as the evaluation

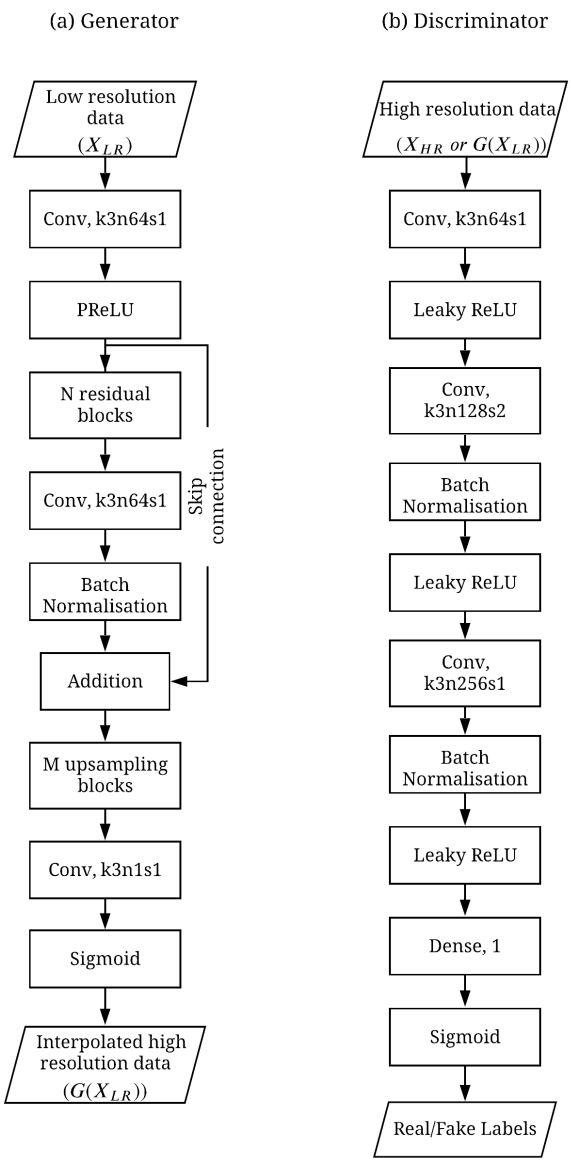

Figure 5: The model architecture of the (a) generator and (b) discriminator. ”Conv” refers to a convolutional layer, the numbers after ”k”, ”n” and ”s” respectively stand for the filter size, number of filters and stride amount of the convolutional (e.g. k3n64s1 indicates that the convolutional layer has a filter size of 3×3, 64 filters and a stride of 1).

Table 1: Hyper-parameters adopted for model training of the SRGAN model.

Hyper-parameter	Selected Value
Learning rate	10^{-4}
Momentum (β_1)	0.5
Mini-batch size	128
Number of residual blocks (N)	5
Number of upsampling blocks (M)	2 for hourly; 1 for 30-minute
Weighting factor (λ)	10^{-3}

metric which measures the distances between the interpolated and the measured data probability distributions:

$$JSD(P\|Q) = \sum_{s\epsilon S}[P(s)log(\frac{P(s)}{Z(s)}) + Q(s)log(\frac{Q(s)}{Z(s)})] \tag{7}$$

Where P and Q are the probability distributions of the measured and interpolated data defined on the same probability space S, $Z = \frac{1}{2}(P+Q)$, s represents a possible outcome from S.

The tuned hyper-parameters are summarised in Table 1. Same as the set-up in [14], we applied an Adam optimiser [39] with a learning rate of 10^{-4} to update the SRGAN's parameters. The weighting factor λ in Eq. (5) is 10^{-3}, achieving a good balance between the MSE loss and adversarial loss. On the other hand, we respectively changed the mini-batch size to 128 and momentum β_1 to 0.5. Another change made is to use five residual blocks instead of 16 used in [14]. Empirically it is found that these adjustments made the training process more efficient and stable for all the evaluated interpolation scenarios using the adopted dataset and GPU. Moreover, interpolating hourly PV/load data requires two upsampling blocks, while one upsampling block is sufficient for interpolating 30-minute data. The range for the optimal numbers of training iterations are between 10^5 to 5×10^5. More iterations are required for interpolating hourly data as it has an additional upsampling block.

3. Results and Discussion

This section can be divided into three main parts according to the objectives of the study: Section 3.1 and 3.2 to visually and statistically assess the performance of the SRGAN model; Section 3.3 and 3.4 to inspect the errors of the SRGAN model for different groups of PV and load power profiles and for a different benchmark dataset; Section 3.5 to introduce the comparative methods and evaluate their performances regarding the end-use application of reducing errors in estimated electricity costs and battery savings for a PV-battery system.

Since the SRGAN model is applied to interpolate 5-minute PV and load power data from 30-minute and hourly power measurements, a total of four interpolation scenarios are evaluated in this section. Hence, most figures in this section include four subplots. Each subplot presents results for a single interpolation scenario.

3.1. Visual inspection

The first step of the model evaluation is to inspect the SRGAN generated profiles and their ground truth visually. This involved two main steps: the first is to check whether both measured and interpolated profiles share similar overall patterns; the second step assesses whether both profiles present the same level of fluctuations, especially for cloudy-day PV profiles and load profiles with consumption spikes. Another aspect of evaluating is whether it is necessary to include the adversarial loss component in the loss function in Eq. (5) for interpolating PV/load power data. Hence, in addition to the SRGAN model, another approach with the same model architecture is trained only using the MSE loss component in Eq. (5). This model is referred to as the super resolution mean squared error (SR-MSE) approach.

Figure 6 shows a few examples of SRGAN and SR-MSE interpolated 5-minute PV and load power profiles and their respective input 30-minute/hourly and 5-minute measured power profiles. For an example of a clear-sky day PV power profile, as shown in Figure 6(a), the synthetic profile generated by SRGAN matches well with the measured profile. A cloudy day PV power profile is illustrated in Figure 6(c). Although some discrepancies can be observed where the SRGAN interpolated profile does not

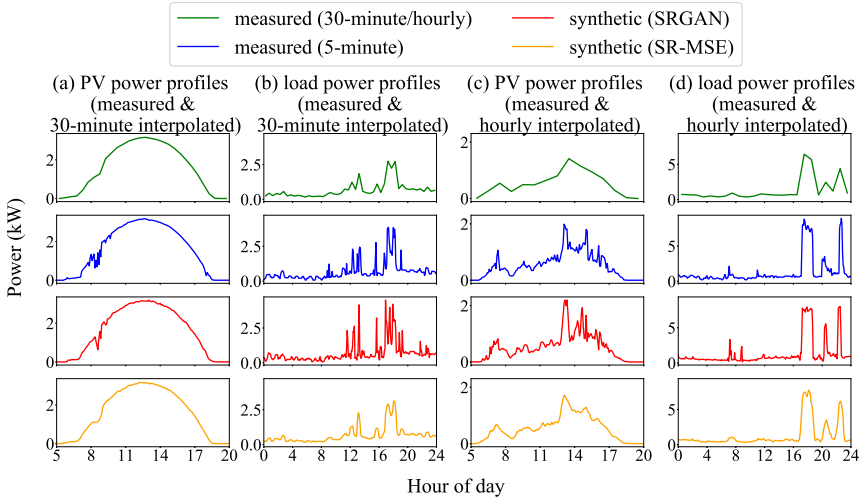

Figure 6: 5-minute daily PV power measured profiles compared to synthetic profiles generated by the SRGAN and SR-MSE models for (a) a clear-sky day interpolated from 30-minute data and (c) a cloudy day interpolated from hourly data; 5-minute daily load measured profiles compared to synthetic load profiles interpolated by the SRGAN and SR-MSE models from (b) a 30-minute power profile and (d) an hourly power profile. The measured 30-minute/hourly and 5-minute profiles are shown in the first and second row, whereas the third and fourth rows show interpolated power profiles using the SRGAN and SR-MSE models.

match the measured profile point by point, it captures the overall pattern and variations in power quite well. Similar results can be observed from Figure 6(b) and 6(d), which compare two 5-minute load profiles respectively interpolated from 30-minutes and hourly resolutions using SRGAN, to their corresponding measured profiles. On the other hand, although the SR-MSE approach can capture the overall patterns of the PV and load profiles, its generated profiles seem to be too smooth and less convincing, especially for load profiles and cloudy-day PV profiles. This shows the necessity of applying the adversarial loss for interpolating PV and load power data, similar to what is found for image super-resolution [14].

3.2. Data distribution and autocorrelation

In order to illustrate the distances between the data probability distributions of measured and SRGAN interpolated power profiles for households in the test set, the cumulative distribution functions (CDFs) of measured and interpolated data are shown in

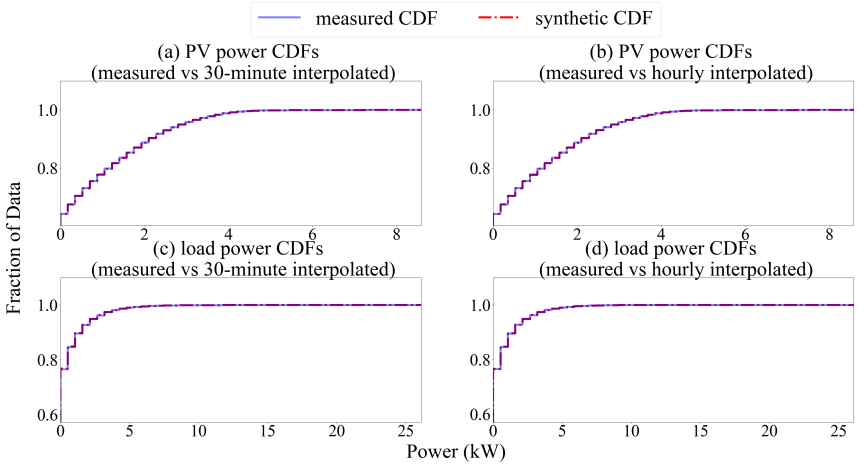

Figure 7: CDFs of measured 5-minute PV power data and synthetic 5-minute data interpolated from (a) 30-minute power data and (b) hourly power data using the SRGAN model; CDFs of measured 5-minute load power data and synthetic 5-minute data interpolated from (c) 30-minute power data and (d) hourly power data using the SRGAN model.

Figure 7. The CDFs are computed using the same method for both the measured and interpolated datasets: the minimum and maximum powers from the measured test set are used to set the range of values in the x-axis; then this range is divided into 50 equal-width intervals and the cumulative probabilities are calculated for each power interval. There is almost no visible difference between the synthetic and measured CDFs for all the evaluated scenarios, which indicates that the SRGAN model is able to generate 5-minute interpolated power profiles from the identical data probability distributions of the measured data.

Figure 8 demonstrates the mean daily autocorrelation profiles of measured and SRGAN interpolated datasets for all the four evaluated scenarios: PV/load data interpolation from 30-minute/hourly resolution. To compute a mean daily profile, autocorrelations are calculated for all the daily power profiles in the measured/synthetic evaluation set. Then they are averaged for each 5-minute timestamp of a day. Similar to the CDF results, the mean daily autocorrelations of the SRGAN interpolated data match quite well with the ground truth, which means the SRGAN model is capable of capturing the

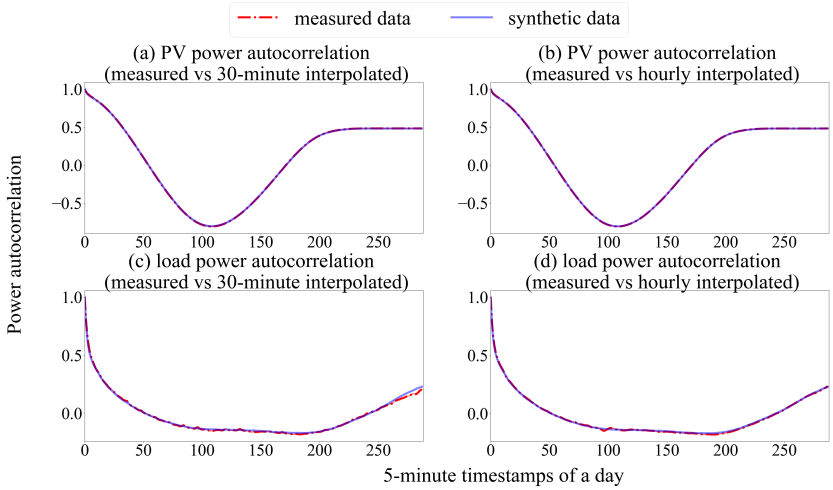

Figure 8: Mean daily autocorrelation profiles of measured 5-minute PV power data compared to synthetic 5-minute data interpolated from (a) 30-minute power data and (b) hourly power data using the SRGAN model; Mean daily autocorrelation profiles of measured 5-minute load power data compared to synthetic 5-minute data interpolated from (c) 30-minute power data and (d) hourly power data using the SRGAN model.

temporal characteristics of 5-minute load and PV power profiles.

3.3. Performance for different power profiles

It is vital to ensure the SRGAN model performs well against different types of PV/load power profiles and to assess which types of PV/load power scenarios result in better performances. Hence, the PV and load power profiles in the test set are segmented into different clusters, and then assessments are carried on these clusters. The daily clearness index [40] is used to separate PV power profiles as it provides a measure of cloudiness. As the daily clearness index ranges between 0 to 0.8 in the test set, eight equally spaced clearness index intervals of 0.1 are used to group the PV power profiles. The K-means algorithm [41] is used to cluster the normalised 30-minute and hourly load power profiles. One of the aims of this analysis is to inform the potential readers on which types of load profiles may have better performance compared to the others. It is more desirable to use the number of clusters that results in the commonly adopted typical load patterns, instead of adopting the number of clusters with the optimal clustering performance (i.e. 12 clusters used in Section 3.5.1). As five typical load

profiles were used in [42], and most of the resulting load cluster centroids using five clusters are similar to the typical load profiles used in [42]. Hence, in this analysis five clusters are adopted for both temporal resolutions.

The evaluation metric is the normalised root mean squared error (NRMSE) in the daily totals of 5-minute power, the reasons for selecting this metric instead of the JSD used in the model tuning process are two-fold: 1. The interpolated data probability distributions match quite well with the ground truth, leading to JSDs that are too small to compare among various clusters; 2. This metric is also adopted in several similar studies [8, 10]. The NRMSE used in this paper is computed using Eq. (8).

$$NRMSE = \frac{\sqrt{\frac{1}{N_{test}} \times \sum_{n=1}^{N_{test}} (y_n - \widehat{y_n})^2}}{y_{max} - y_{min}} \tag{8}$$

where y_n is a secondary value quantitatively computed using one/multiple measured high resolution daily power profiles in the test set (e.g. daily total of PV/load power values), $\widehat{y_n}$ is the same type of value as y_n, but computed using one/multiple interpolated daily profiles in the test set. N_{test} is the number of test set daily profiles used to compute NRMSE. y_{max} and y_{min} are the maximum and minimum values of y_n.

In this case, to compute the NRMSEs in daily totals for different types of PV/load profiles, y_n and $\widehat{y_n}$ are set to be the sums of the measured and interpolated high resolution power values of a daily load/PV power profile in a given daily clearness index interval or a load cluster. N_{test} is the number of daily PV/load profiles for this clearness index interval/load cluster in the test set.

Figure 9 demonstrates the NRMSEs in daily PV totals for different ranges of daily clearness index. As the clearness index increases, the NRMSEs decreases for both the 30-minute and hourly interpolated datasets. This is expected as there are more weather transients during cloudy days, making it difficult for the SRGAN model to capture all the uncertainties within the PV power profiles accurately.

The NRMSEs in daily totals of load power values are computed in a similar manner. Figure 10 shows the K-means cluster centroids of 30-minute and hourly load power data and their corresponding NRMSEs in daily totals. Both data granularities end up with similar load clusters. Load profiles with relatively small daytime focused

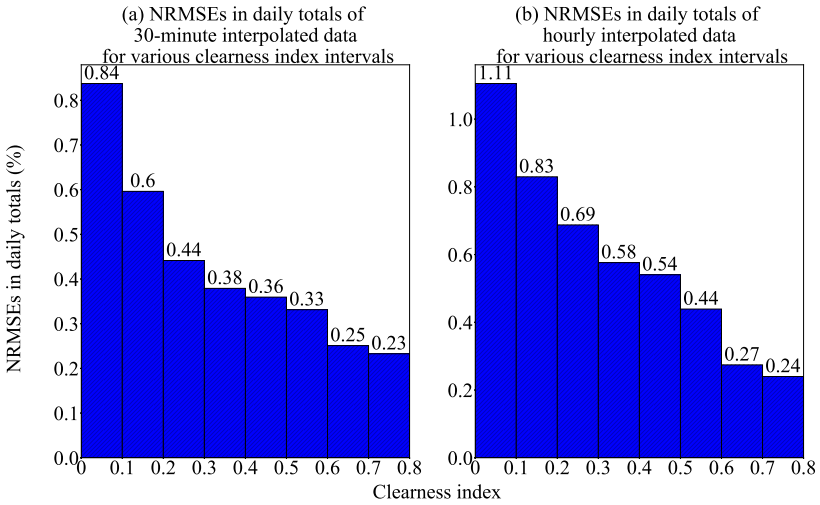

Figure 9: NRMSEs in daily totals of 5-minute synthetic PV power data interpolated from (a) 30-minute and (b) hourly PV power data for different clearness index intervals.

consumption (Cluster 2) result in the smallest NRMSEs in daily totals for both evaluated scenarios, followed by Cluster 5, which contains power profiles with morning and evening peaks. Although Cluster 4 has a similar bell shape as Cluster 2, it has a much higher NRMSE. We then look into the mean daily standard deviations of normalised load profiles in the test set for Cluster 2 and Cluster 4. This is computed by determining the standard deviation of normalised power for each daily profile in a cluster and average them. As a result, Cluster 4 has a much higher mean daily standard deviation and the SRGAN model cannot accurately capture all the consumption fluctuations. This is similar to what is found in Figure 9. Overall the NRMSEs are relatively small and stable across various clusters of load profiles, which means the SRGAN model performs well regardless of the type of load curve.

3.4. Performance in a benchmark dataset

It is worthwhile to investigate the performance of the trained SRGAN model on a different dataset. As a widely adopted benchmark dataset, the Smart Grid Smart City (SGSC) dataset includes 30-minute smart meter data (primarily load data) collected between 2010 and 2014 from Australian households in the state of New South Wales

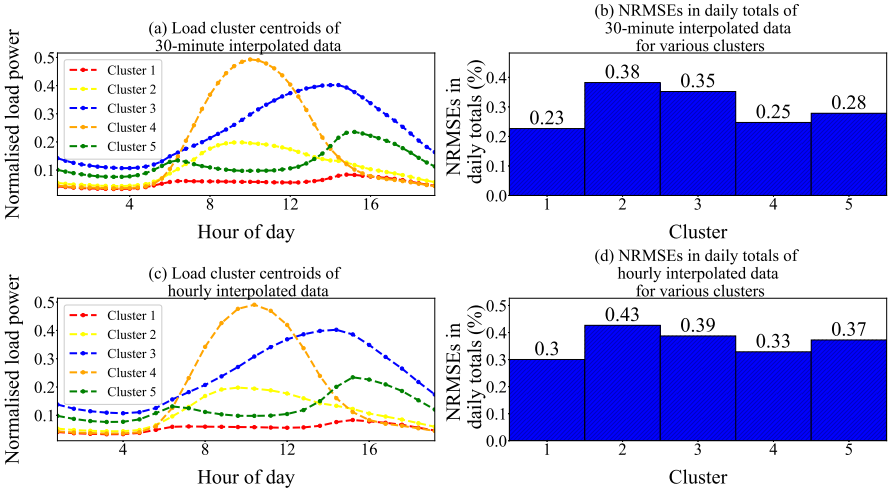

Figure 10: Cluster centroids of (a) 30-minute and (c) hourly normalised daily load power profiles and NRMSEs in daily totals of 5-minute synthetic load power data interpolated from (b) 30-minute and (d) hourly PV power data within these clusters.

(NSW). In this case study, one-year data of 2013-2014 is used for validation, which includes 2839 customers with a full year of load data and 43 households with a whole year of PV data.

Figure 11 demonstrates a few daily power profiles of SGSC data and their interpolated 5-minute power profiles using the SRGAN model trained using the Solar Analytics dataset. Although there is no ground truth for 5-minute SGSC data, visually, the interpolated 5-minute power profiles are realistic. Also they contain weather transients and load spikes that can not be observed from the original measured 30-minute profiles. The absence of 5-minute data also means it is impossible to compare the data probability distributions of the measured and interpolated datasets. Instead, the adopted metric is the NRMSEs in daily totals of load/PV power profiles, which is already used above for validating the model on different types of power profiles in the adopted test set.

Eq. (8) is also used to computed NRMSEs in the daily totals of load/PV power profiles of the SGSC data. In this case y_n and $\widehat{y}_n$ are set to be the measured and interpolated daily totals, which are the sums of the measured 30-minute and interpolated 5-minute power values of a daily load/PV power profile in the SGSC dataset. N_{test} is the total

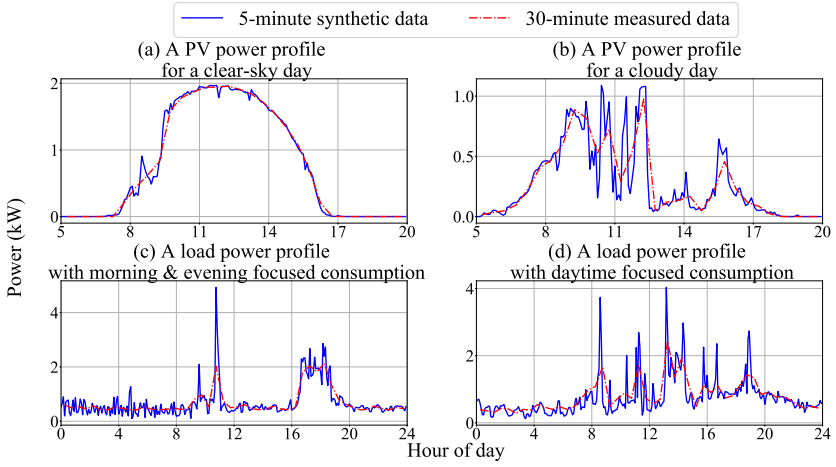

Figure 11: 30-minute measured and 5-minute interpolated PV power profiles from the SGSC dataset for (a) a clear-sky day and (b) a cloudy day; 30-minute measured and 5-minute interpolated load power profiles from the SGSC dataset with (c) morning and evening focused consumption and (d) daytime focused consumption.

number of daily PV/load profiles in the SGSC dataset.

The NRMSEs in daily totals of PV power and load power are respectively 0.39% and 0.14%, which are comparable with the NRMSEs for the test set of the Solar Analytics dataset (0.25% for PV and 0.24% for load). Moreover, the NRMSEs in load daily totals are even lower for the SGSC dataset. Since the SGSC dataset is collected in a different year and quite likely from a different group of households (the Solar Analytics training set only has 693 NSW PV customers, and the remaining 1647 PV households are from other states), this shows that the trained SRGAN model is likely to have the the same level of performance in other datasets with different time windows and geographical scopes.

Although both of the adopted datasets were collected from Australian households, it is suspected that the SRGAN model can still perform well for datasets in other countries since the above results show that the model performs consistently across various types of PV/load profiles and datasets collected in different years.

3.5. Alternative methods and their performances on end-use applications

3.5.1. Conditional SRGAN

The only input to the SRGAN model is the low resolution profile. However, if more information of the power profile can be leveraged to direct the interpolation process of the SRGAN model, it can be extended to a conditional SRGAN (CSRGAN) and possibly improved. This information Y is also referred to as a class label related to the seasonality or classification of the input power profiles, such as the season/month of a year, load clustering labels. Y can be added to both the generator and the discriminator as an extra input vector, as a result, they are both conditioned on Y and Eq. (6) can be easily adjusted to the loss function of the CSRGAN model:

$$\begin{aligned} \min_G \max_D V(D,G) = \frac{1}{u \times m} \sum_{t=1}^{u \times m} (X_t^{HR} - G(X^{LR})_t)^2 + \\ \lambda \times (\mathbb{E}_{X^{HR} \sim p_{HR}}[\log D(X^{HR}|Y)] + \\ \mathbb{E}_{X^{LR} \sim p_{LR}}[\log(1 - D(G(X^{LR}|Y)))]) \end{aligned} \tag{9}$$

In this case, the SRGAN model is converted to a CSRGAN model by respectively adding month number and clustering label as the extra information for interpolating PV and load power profiles. Month number could be helpful to the PV power profile interpolation as it may be related to the seasonal effects on cloud movements, and clustering label could also be useful for generating interpolated load profiles as various load clusters may have their distinct load characteristics such as the amount of the consumption spikes. The K-means algorithm [41] is applied to cluster the low-resolution load power datasets (30-minute and hourly), Davies-Bouldin index (DBI) [43] is used as the metric to select the optimal numbers of clusters. As a result, 12 clusters are adopted for clustering both the 30-minute and hourly load power data.

An alternative naive prediction model is also implemented as a comparison to the SRGAN and CSRGAN approaches. The main idea of the naive prediction is similar to what is done in [11]: for a given 30-minute/hourly daily profile in the evaluation set, another daily profile in the training set that has the closest Euclidean distance is selected. Then for predicting the 5-minute profile, the corresponding 5-minute profile of the closest 30-minute/hour profile is adopted as a naive prediction. To make a compre-

hensive comparison, results are also derived for the cases where measured 30-minute, hourly and 5-minute datasets are available. The results of the 5-minute dataset are used as an ideal case which allows us to compute the errors in daily, monthly and yearly totals of PV/load power values and errors in estimating electricity costs and battery savings, whereas 30-minute and hourly datasets are applied to produce a baseline of the cost and saving results.

3.5.2. Estimation of daily, monthly and yearly totals

Table 2 compares the NRMSEs in the daily, monthly and yearly totals of interpolated 5-minute load/PV power values using the CSRGAN, SRGAN and naive prediction approaches. For the evaluated scenarios, the CSRGAN approach has a better overall performance in estimating the PV power totals compared to the SRGAN model. The only exception is when predicting the daily PV totals using hourly data as input temporal resolution. On the other hand, inputting additional information only improves the estimation of load power totals when interpolating 30-minute load data. The CSRGAN model results in larger NRMSEs when hourly data is provided.

It is also vital to inspect how the NRMSEs of these interpolation models fluctuate for different households in the test set. Moreover, it would be desirable to compare the NRMSEs against other relevant studies. Although there is no existing studies on interpolating 5-minute PV/load power data, studies in [8–10] interpolate 5-minute/10-minute irradiance data from hourly data as reviewed in Section 1. As PV generation is strongly dependent on solar irradiance data, the performances of our model and the reviewed studies can be roughly compared. It should be noted that the comparisons are not entirely fair as the reviewed studies interpolate irradiance data for a few weather stations while this study aims to interpolate PV power data for households. They reported one NRMSE in the daily totals of solar irradiance for each weather station, taking account of all the collected daily irradiance profiles for that weather station. We use the same metric to generate the box plots in Figure 12 for each evaluated scenario and interpolation model. It should be noted that the NRMSEs on this plot are different to the NRMSEs of daily totals in Table 2: The NRMSEs in Table 2 are computed using all daily profiles in the test set while the household-level NRMSEs in Figure 12 are

Table 2: NRMSEs in daily/monthly/yearly totals of interpolated 5-minute PV and load power data. NRMSEs are determined for the whole test set.

NRMSE in PV totals (%)				
Method	input data resolution	Daily	Monthly	Yearly
CSRGAN	30-minute	**0.25**	**0.12**	**0.07**
	hourly	0.36	**0.15**	**0.11**
SRGAN	30-minute	0.25	0.17	0.16
	hourly	**0.32**	0.16	0.12
Naive prediction	30-minute	1.29	0.72	0.63
	hourly	1.11	0.57	0.48
NRMSE in load totals (%)				
Method	input data resolution	Daily	Monthly	Yearly
CSRGAN	30-minute	**0.23**	**0.15**	**0.18**
	hourly	0.39	0.40	0.53
SRGAN	30-minute	0.24	0.19	0.23
	hourly	**0.28**	**0.17**	**0.17**
Naive prediction	30-minute	1.85	1.27	1.67
	hourly	1.72	1.11	1.43

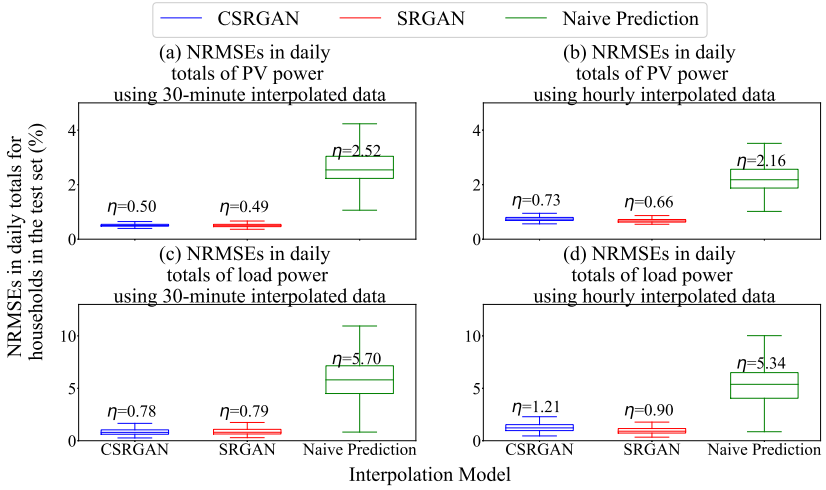

Figure 12: Household-level NRMSEs in daily totals of 5-minute PV power data interpolated from (a) 30-minute and (b) hourly measured data, using the CSRGAN, SRGAN and naive prediction approaches; household-level NRMSEs in daily totals of 5-minute load power data interpolated by the CSRGAN, SRGAN and naive prediction approaches, from (c) 30-minute and (d) hourly measured data across households in the test set. η is the median value of the household-level NRMSEs for an interpolation scenario using one interpolation model.

generated individually for each household in the test set to form a box plot, using on year of daily load/PV power profiles.

As the reviewed studies all used small datasets and it is unclear whether these NRMSEs are normally distributed, it makes more sense to compare the medians of the NRMSEs instead of their means. Hence, the medians (η) for each evaluated interpolation scenario are displayed on top of the box plots in Figure 12 for each interpolation model. The median NRMSEs in daily totals of irradiance among various reported locations are respectively 3% in [8], 0.65% in [9] and 0.9% in [10]. Our model has a better performance compared to the approach in [10]. The work in [8] and [9] interpolate 10-minute instead of 5-minute data from hourly measurements. Despite having a higher upsampling factor, the NRMSE median shown in Figure 12(b) is 0.66 % for the SRGAN model, which is quite close to the 0.65 % median in [9] and much smaller compared to the reported value in [8].

Similar to the results in Table 2, the SRGAN model has better performances over

the other two alternative methods when interpolating 5-minute load/PV data from hourly resolution. However, the medians and interquartile ranges (IQR) of the NRMSEs across test set households for interpolating PV and load power data are relatively close between the SRGAN and CSRGAN models when using 30-minute data as inputs. Hence, paired Wilcoxon signed-rank tests are performed, which is a non-parametric statistical significance test to compare two paired groups of samples [44]. In this case, we use one-sided instead of two-sided tests to assess which approach results in smaller NRMSEs for households in the test set. Another aim is to determine whether there are sufficient households in the test set for us to find the optimal interpolation approach for each evaluated scenario. As a result, Wilcoxon signed-rank tests are conducted to compare the three approaches for each considered interpolation scenario, all of them returned a p-value < 0.05. Moreover, the statistical tests show that the CSGAN model achieves the lowest NRMSEs (p-value = 0.0012) in terms of interpolating load data from 30-minute resolution. On the other hand, despite having a lower NRMSE in Table 2, the CSRGAN model leads to higher household-level NRMSEs (p-value = 0.006) for interpolating 5-minute PV power data from 30-minute resolution compared to the SRGAN model.

Unfortunately, the metric reported in the reviewed load data interpolation study [13] was the root mean square error (RMSE), and there was no unit provided for the RMSEs. Moreover, the reviewed study aimed to interpolate very high frequency data (100/1000 Hz), which is quite different to our scope. Hence, it is not feasible to compare between our model and the approach in [13].

3.5.3. Estimation of electricity costs and battery saving potentials

One potential end-use application of the interpolated PV/load data is to provide more accurate estimations of electricity costs and battery saving potentials for households with PV when only coarse meter data is available. In this work, the battery simulation model in [45] is adopted to evaluate the interpolated data, which requires PV and load data as inputs, simulates the operations of a residential battery and computes the electricity costs with & without a battery and potential battery savings for an Australian solar household. Also, this case study follows the same economical param-

eters, battery specifications, charging & discharging algorithm and tariff structures (flat and time-of-use (ToU)) in [45]. For each household, the battery size range is set to be 1-15 kWh with an increment of 1 kWh (where 1 kWh is equivalent to 3.6 MJ). The potential battery savings are computed for each battery size by taking the difference between the electricity costs with & without a battery.

Table 3 illustrates the normalised root mean squared error (NRMSE) and r-squared values in estimated yearly electricity costs and battery saving potentials using low resolution measured data and interpolated 5-minute data for the households in the test set. For using 30-minute PV & load data as inputs, the CSRGAN model can achieve the smallest errors in estimating electricity costs and battery saving potentials under the tested flat and ToU tariffs. On the other hand, in terms of adopting hourly PV & load data, the SRGAN approach produces the smallest NRMSEs and the highest r-squared values in estimating electricity costs and battery savings for all the evaluated scenarios. Both the CSRGAN and SRGAN have much better performances than the measured low resolution data and the naive prediction approach. Compared to just using hourly measured data, under the flat and the ToU tariffs, the SRGAN model respectively leads to 41.2% and 44.7% error reductions in estimating electricity costs, 42.9% and 41.7% error reductions in estimating battery saving potentials.

Another aspect to assess is the performances of different methods for each individual site. The relative error in terms of battery savings is applied as the metric. It has been used for both sensitivity analyses in [7] and [6] to assess the impacts of using low resolution PV and load data. Figure 13 illustrates the box plots of household-level relative errors in percentages using the four approaches in Table 3. The SRGAN model has the lowest ranges of percentage relative errors using hourly PV and load datasets as inputs. Like Section 3.5.2, one-sided Wilcoxon signed-rank tests are conducted on the percentage relative errors of all four approaches for each considered interpolation scenario with a p-value smaller than 0.05. As a result, the SRGAN model also outperforms the CSRGAN model using PV and load datasets interpolated from 30-minute resolution.

The above results indicate that the SRGAN model can address the inaccuracies in the estimated electricity costs and battery savings caused by using low granularity data

Table 3: NRMSEs and r-squared values for estimating yearly electricity costs and battery saving potentials using low resolution measured data, 5-minute data interpolated by the CSRGAN, SRGAN and naive prediction models.

Errors in yearly electricity costs

Tariff		Flat		ToU	
Method	input data resolution	NRMSE (%)	r squared	NRMSE(%)	r squared
CSRGAN	**30-minute**	**0.24**	**0.99972**	**0.24**	**0.99976**
	hourly	0.46	0.99903	0.48	0.99906
SRGAN	30-minute	0.28	0.99965	0.31	0.99960
	hourly	**0.28**	**0.99963**	**0.29**	**0.99966**
Measured (low resolution)	30-minute	0.29	0.99960	0.31	0.99961
	hourly	0.48	0.99892	0.52	0.99890
Naive prediction	30-minute	2.16	0.97842	2.34	0.97766
	hourly	1.84	0.98430	1.94	0.98459

Errors in yearly battery savings

Tariff		Flat		ToU	
Method	input data resolution	NRMSE(%)	r squared	NRMSE(%)	r squared
CSRGAN	**30-minute**	**2.59**	**0.96576**	**1.82**	**0.98621**
	hourly	3.86	0.92970	2.84	0.96887
SRGAN	30-minute	2.93	0.95917	2.10	0.98220
	hourly	**3.66**	**0.93686**	**2.52**	**0.97526**
Measured (low resolution)	30-minute	4.13	0.93673	2.67	0.97724
	hourly	6.41	0.84935	4.32	0.94229
Naive prediction	30-minute	5.83	0.87075	4.93	0.92733
	hourly	5.75	0.87180	4.61	0.93373

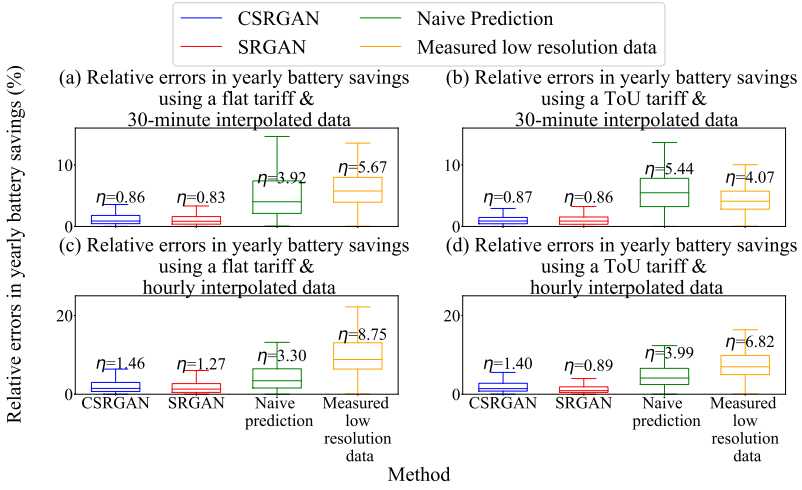

Figure 13: Household-level percentage relative errors in estimating yearly battery saving potentials using 30-minute measured data, data interpolated by CSRGAN, SRGAN and naive prediction models using 30-minute input data under (a) a flat tariff and (b) a ToU tariff; household-level percentage relative errors in estimating yearly battery saving potentials using hourly measured data, data interpolated by CSRGAN, SRGAN and naive prediction models using hourly input data under (c) a flat tariff and (d) a ToU tariff. η is the median value of the household-level percentage relative errors using one interpolation method, an input resolution (30-minute/hourly) and a tariff (flat/ToU).

in the power optimisation of PV battery systems. Hence, the SRGAN model can be potentially integrated into a battery sizing tool that could still achieve accurate battery sizing results when only low resolution input data is available.

4. Conclusion

In this work, a SRGAN based model is proposed to synthetically interpolate 5-minute average PV and load power data from 30-minute and hourly data.

Detailed evaluations of the SRGAN model have been performed, a few key findings are summarised below:

1. Visual inspections show the necessity of adding an adversarial loss component to generate realistic 5-minute power profiles.

2. The SRGAN model can fully capture the data probability distribution and temporal characteristics of the measured 5-minute data.

3. The model performs consistently on different types of load and PV profiles. Even though the SRGAN model is trained using the Solar Analytics dataset, it achieves the same level of performance on the Smart Grid Smart City (SGSC) dataset, which has a different time window and geographical scope.

4. The SRGAN interpolated data can be applied to derive much better estimations of electricity costs and battery saving potentials of PV battery systems than using low resolution data or a naive forecasting approach.

5. By providing additional information during the data interpolation process, the SRGAN model is turned into a CSRGAN approach. However, the CSRGAN does not provide any significant improvements over the SRGAN model. It even leads to more inaccuracies when interpolating hourly data especially for load data.

The above findings indicate that the proposed model can address the issue of limited proprietary high resolution data in modelling and optimisation of a PV-integrated battery system. Moreover, the SRGAN model can be potentially used to generate high resolution PV/load power data for data-driven PV/load forecasting and load disaggregation models, when only low resolution smart meter data is available.

For future work, it would be desirable to improve the CSRGAN model and explore other types of information that could potentially assist the interpolation process. It will also be worthwhile to evaluate the proposed approach for interpolating power profiles of other types of renewable generation (e.g. wind) or finer temporal resolutions (e.g. 1-minute) in order to assess how well the SRGAN model generalises in time series that are different to the datasets adopted in this work.

Acknowledgment

The authors would like to thank Solar Analytics for providing the datasets for this study.

Appendix A. Training Steps for the SRGAN Model

1: **Input** number of iterations I, mini-batch size N_{batch}, LR and HR daily PV/load profile pairs $[X_n^{LR}, X_n^{HR}]$.

2: **for** i in (1, 2, ..., I) **do**

3: A mini-batch that consists of multiple LR and HR daily PV/load profile pairs ($[X_n^{LR}, X_n^{HR}]$, $n = 1, 2, ..., N_{batch}$) is randomly drawn from the training data.

4: The generator parameters are kept constant. The mini-batch is used to update the discriminator parameters through backpropagating the loss defined in Eq. (2).

5: Another mini-batch ($[X_n^{LR}, X_n^{HR}]$, $n = 1, 2, ..., N_{batch}$) is sampled from the training data.

6: The discriminator parameters are kept constant. The second min-batch is used to update the generator parameters through backpropagating the loss defined in Eq. (5).

7: **end for**

8: **Output** the trained SRGAN model.

References

[1] B. Yildiz, J. Bilbao, J. Dore, A. Sproul, Recent advances in the analysis of residential electricity consumption and applications of smart meter data, Applied Energy (2017). doi:10.1016/j.apenergy.2017.10.014.
URL http://linkinghub.elsevier.com/retrieve/pii/S0306261917314265

[2] Y. Wang, Q. Chen, T. Hong, C. Kang, Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges, IEEE Transactions on Smart Grid 10 (3) (2019) 3125–3148. arXiv:1802.04117, doi:10.1109/TSG.2018.2818167.

[3] K. Wang, X. Qi, H. Liu, A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network, Applied Energy 251 (May) (2019) 113315. doi:10.1016/j.apenergy.2019.113315.
URL https://doi.org/10.1016/j.apenergy.2019.113315

[4] T. Beck, H. Kondziella, G. Huard, T. Bruckner, Assessing the influence of the temporal resolution of electrical load and PV generation profiles on self-consumption and sizing of PV-battery systems, Applied Energy 173 (2016) 331–342. doi:10.1016/j.apenergy.2016.04.050.
URL http://dx.doi.org/10.1016/j.apenergy.2016.04.050

[5] A. Hawkes, M. Leach, Impacts of temporal precision in optimisation modelling of micro-combined heat and power, Energy 30 (10) (2005) 1759–1779. doi:10.1016/j.energy.2004.11.012.

[6] K. Abdulla, K. Steer, A. Wirth, J. D. Hoog, S. Halgamuge, The Importance of Temporal Resolution in Evaluating Residential Energy Storage, Ieee Pesgm 2017 (2017) 7.
URL https://www.researchgate.net/publication/313857976{_}The{_}Importance{_}of{_}Temporal{_}Resolution{_}in{_}Evaluating{_}Residential{_}Energy{_}Storage

[7] R. Tang, K. Abdulla, P. H. Leong, A. Vassallo, J. Dore, Impacts of Temporal Resolution and System Efficiency on PV Battery System Optimization, Asia-Pacific Solar Research Conference 2015 (2017).
URL http://apvi.org.au/solar-research-conference/wp-content/uploads/2017/12/029{_}R-Tang{_}DI{_}Paper{_}Peer-reviewed.pdf

[8] J. Polo, L. F. Zarzalejo, R. Marchante, A. A. Navarro, A simple approach to the synthetic generation of solar irradiance time series with high temporal resolution, Solar Energy 85 (5) (2011) 1164–1170. doi:10.1016/j.solener.2011.03.011.
URL http://dx.doi.org/10.1016/j.solener.2011.03.011

[9] M. Larrañeta, S. Moreno-Tejera, M. A. Silva-Pérez, I. Lillo-Bravo, An improved model for the synthetic generation of high temporal resolution direct normal irradiation time series, Solar Energy 122 (2015) 517–528. doi:10.1016/j.solener.2015.09.030.

[10] A. P. Grantham, P. J. Pudney, L. A. Ward, M. Belusko, J. W. Boland, Generating

synthetic five-minute solar irradiance values from hourly observations, Solar Energy 147 (2017) 209–221. doi:10.1016/j.solener.2017.03.026.
URL http://dx.doi.org/10.1016/j.solener.2017.03.026

[11] C. M. Fernández-Peruchena, M. Gastón, A simple and efficient procedure for increasing the temporal resolution of global horizontal solar irradiance series, Renewable Energy 86 (2016) 375–383. doi:10.1016/j.renene.2015.08.004.

[12] A. Attya, T. Hartkopf, Generation of high resolution wind speeds and wind speed arrays inside a wind farm based on real site data, in: 11th International Conference on Electrical Power Quality and Utilisation, IEEE, 2011, pp. 1–6.

[13] G. Liu, J. Gu, J. Zhao, F. Wen, G. Liang, Super Resolution Perception for Smart Meter Data, Information Sciences 526 (2020) 263–273. doi:10.1016/j.ins.2020.03.088.

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, et al., Photo-realistic single image super-resolution using a generative adversarial network, Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 2017-Janua (2017) 105–114. arXiv:1609.04802, doi:10.1109/CVPR.2017.19.

[15] J. Liu, F. Chen, X. Wang, H. Liao, An edge enhanced srgan for mri super resolution in slice-selection direction, in: Multimodal Brain Image Analysis and Mathematical Foundations of Computational Anatomy, Springer, 2019, pp. 12–20.

[16] Y. Nagano, Y. Kikuta, Srgan for super-resolving low-resolution food images, in: Proceedings of the Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management, 2018, pp. 33–37.

[17] Y. Chen, Y. Wang, D. Kirschen, B. Zhang, Model-Free Renewable Scenario Generation Using Generative Adversarial Networks, IEEE Transactions on Power Systems 33 (3) (2017) 3265–3275. arXiv:1707.09676, doi:10.1109/TPWRS.

2018.2794541.
URL http://arxiv.org/abs/1707.09676

[18] R. Tang, P. H. W. Leong, J. Dore, A. Vassallo, Generating Residential PV Production and Electricity Consumption Scenarios via Generative Adversarial Networks, 2018 Asia-Pacific Solar Research Conference (2018).
URL http://apvi.org.au/solar-research-conference/wp-content/uploads/2019/01/068_DI_Tang_R_2018.pdf

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[20] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, et al., Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1874–1883.

[21] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, W. Shi, Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize, arXiv preprint arXiv:1707.02937 (2017).

[22] A. Odena, V. Dumoulin, C. Olah, Deconvolution and checkerboard artifacts, Distill (2016). doi:10.23915/distill.00003.
URL http://distill.pub/2016/deconv-checkerboard

[23] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90.

[24] X. Mao, C. Shen, Y.-B. Yang, Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections, in: Advances in neural information processing systems, 2016, pp. 2802–2810.

[25] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition, IEEE/ACM Transactions on audio, speech, and language processing 22 (10) (2014) 1533–1545.

[26] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3128–3137.

[27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[28] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).

[29] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[30] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: ICML, 2010, p. 807–814.

[31] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks, arXiv preprint arXiv:1511.06434 (2016) 1–16arXiv:1511.06434, doi:10.1051/0004-6361/201527329.
URL https://arxiv.org/pdf/1511.06434.pdf

[32] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. icml, Vol. 30, 2013, p. 3.

[33] Solar Analytics, Connect with your solar, https://www.solaranalytics.com/au/, [Online; accessed 10-October-2019] (2019).

[34] Wattwatchers, Wattwatchers: Super-smart devices for energy monitoring, https://wattwatchers.com.au/, [Online; accessed 08-November-2019] (2019).

[35] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, nature 323 (6088) (1986) 533–536.

[36] F. Chollet, et al., Keras, https://keras.io (2015).

[37] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.

[38] J. Lin, Divergence measures based on the shannon entropy, IEEE Transactions on Information theory 37 (1) (1991) 145–151.

[39] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[40] J. Black, C. Bonython, J. Prescott, Solar radiation and the duration of sunshine, Quarterly Journal of the Royal Meteorological Society 80 (344) (1954) 231–235.

[41] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, University of California Press, Berkeley, Calif., 1967, pp. 281–297.
URL https://projecteuclid.org/euclid.bsmsp/1200512992

[42] Solar Choice, Solar & battery calculator – advanced version, https://www.solarchoice.net.au/blog/solar-pv-battery-storage-sizing-payback-calculator, [Online; accessed 01-April-2021] (2021).

[43] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE transactions on pattern analysis and machine intelligence PAMI-1 (2) (1979) 224–227.

[44] R. Woolson, Wilcoxon signed-rank test, Wiley encyclopedia of clinical trials (2007) 1–3.

[45] R. Tang, B. Yildiz, P. Leong, A. Vassallo, J. Dore, Residential battery sizing model using net meter energy data clustering, Applied Energy 251 (2019). doi:10.1016/j.apenergy.2019.113324.