# SOLAR POWER FORECASTING

A thesis submitted in fulfilment of the requirements for the
degree of Doctor of Philosophy in the School of Computer Science at
The University of Sydney

Zheng Wang
September 2019

# Abstract

Solar energy is one of the most promising environmentally-friendly energy sources. Its market share is increasing rapidly due to advances in PhotoVoltaic (PV) technologies, which have led to the development of more efficient PV solar panels and the significant reduction of their cost. However, the generated solar energy is influenced by meteorological factors such as solar radiation, cloud cover, rainfall and temperature. This variability affects negatively the large scale integration of solar energy into the electricity grid. Accurate forecasting of the power generated by PV systems is therefore needed for the successful integration of solar power into the electricity grid. The objective of this thesis is to explore the possibility of using machine learning methods to accurately predict the generated solar power so that this sustainable energy source can be better utilized. We consider the task of predicting the PV power for the next day at half-hourly intervals.

At first, we explored the potential of instance-based methods and propose two new methods: the data source weighted nearest neighbor DW$k$NN and the extended Pattern Sequence Forecasting (PSF) algorithms. DW$k$NN is an extension of the standard nearest neighbour algorithm; it uses multiple data sources (historical PV power data, historical weather data and weather forecasts) and considers the importance of these data sources in the final prediction by learning the best weights for them based on previous data. PSF1 and PSF2 are extensions of the standard PSF algorithm which is only applicable to a single data source (historical PV data) to deal with data from multiple related time series. Our evaluation using Australian data showed that the proposed extensions were more accurate than the methods they extend.

Then, we proposed two clustering-based methods for PV power prediction: direct and pair patterns. Many recent algorithms create a single prediction model for all weather types. In contrast, we used clustering to partition the days into groups with similar weather characteristics and then created a separate PV power prediction model

for each group. The direct clustering groups the days based on their weather profiles, while the pair patterns considers the weather type transition between two consecutive days. The proposed methods were evaluated and compared with methods without clustering using Australian data. The results showed that clustering-based models outperformed the other models used for comparison.

We also investigated ensemble methods and proposed static and dynamic ensembles of neural networks. We proposed three strategies for creating static ensembles based on random example and feature sampling, as well as four strategies for creating dynamic ensembles by adaptively updating the weights of the ensemble members based on past performance. Our results showed that all static ensembles were more accurate than the single prediction models and classical ensembles used for comparison, and that the dynamic ensemble further improved the accuracy. We then explored the use of meta-learning to improve the performance of the dynamic ensembles. Instead of calculating the weights of the ensemble members based on their past performance, meta-learners were trained to predict the performance of the ensemble members for the new day and calculate the weights accordingly. The results showed that the use of meta-learning further improved the accuracy of dynamic ensemble.

The methods proposed in this thesis can be used by PV plant and electricity market operators for decision making, improving the utilisation of the generated PV power, avoiding waste, planning maintenance and reducing costs, and also facilitating the large-scale integration of PV power in the electricity grid.

*Dedicated to my loving parents for their endless support.*

# Acknowledgements

I would like to thank all people who have encouraged and supported me to complete this thesis.

I would like to express my sincere gratitude to my supervisor, Associate Professor Irena Koprinska, from the School of Computer Science at the University of Sydney, for her continuous support of my research, which has helped me to successfully grow as a research data scientist. She guided me to explore and clarify my research direction in the beginning of my research work and provided consistent guidance with infinite patience throughout my PhD study. Moreover, I appreciate her kind help and comfort when I suffered from deep sorrows and got lost. Thanks to her consistent help, I finally completed this thesis. I also want to thank Dr Mashud Rana and Dr Ling Luo for their help and technical support. It is my honor to have collaborated with them during my research studies.

I would also like to thank my parents for their endless support and their unreserved love all the time. When I struggled with my research work and almost lost all patience and motivation, it is their love and unlimited encouragement that helped me overcome all the difficulties. I appreciate all the sacrifices they made for me.

I would like to thank the Australian Government for granting me the Postgraduate Award, which reduced my financial burden and enabled me to concentrate on my research work. Special thanks to Associate Professor Xiuying Wang and her research team for their support during my PhD studies.

Finally, I would like to thank my friends Jingcheng Wang, Sheng Hua, Hui Cui, Chaojie Zheng and Yu Zhao for their company, encouragement and kind help with my research work. I can hardly bring this thesis to this successful ending without their support.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Solar energy is an important source of renewable energy. It is clean, abundant and easily accessible. Solar energy can be easily collected by using Photo Voltaic (PV) panels, either small-scale roof-top installations or large-scale solar farms. The solar energy can be transformed into electricity and used to supply the electricity in the building or integrated into the electricity grid. In recent years, the PV technology has developed rapidly and is now one of the most promising technologies for producing solar power. The increased efficiency and affordability of PV solar panels has led to the rapid growth of installed PV solar panels around the world, both stand-alone and grid-connected.

Since solar power is environmentally-friendly, many governments are encouraging its use by providing incentives. Due to all these reasons, solar power is expected to contribute significantly to the future global energy supply. For example, it is predicted that the next four years would witness a triple increase in the capacity of the installed PV power systems worldwide, reaching 540GW [1], and that by 2050, about 30% of Australian energy supply will come from PV systems [2].

Even though solar energy has many advantages compared to other traditional energy sources such as coal and natural gas, the produced PV power output is highly variable as it depends on the solar irradiance and other meteorological factors such as solar angle, solar hours, cloud cover, rainfall and temperature. Solar energy is also an intermittent energy source as it is only available during the day time. This variability and intermittence of solar power makes its large-scale integration into the power grid challenging. Unexpected changes in the solar power often happen, negatively affecting

the grid balance and increasing the operational costs. To minimize the possible negative consequences and ensure a larger penetration of PV power in to the energy mix, there is a need for accurate forecasting of the electricity generated by PV systems.

The solar power forecasting methods can be divided into two groups: indirect and direct. The indirect methods firstly predict the solar irradiance and then convert this prediction into solar power output based on the characteristics of the PV plant and other domain knowledge. Technologies such as NWP and satellite image processing are used together to analyze complex meteorological data such as cloud cover movement and solar angle changes in order to predict the solar irradiance and make the final prediction [3, 4, 5, 6]. The accuracy of indirect models to a large extent depends on the accuracy of individual components and the availability of weather information. However, the meteorological information required for making accurate indirect forecasts is not always available for the the location of the PV plants. The application of indirect methods also heavily relies on domain knowledge of power engineering. These factors limit the applicability of the indirect methods.

In contrast, the direct group of methods directly predict the output of the PV power systems, without the need to firstly predict the solar irradiance. The main data source is the previous PV power data which is readily available, and the additional data sources include historical weather data and weather forecasts for the new days. This weather information is less complex and easily available for the location of the PV plant than the information required by the indirect methods. Also, the additional data sources can be used to improve the accuracy compared to only using the historical PV power data, but they are no longer indispensable. This enables the wider application of direct approaches, compared to the direct ones. The direct approaches can be further divided into two groups, namely statistical and machine learning methods. The former are based on statistical models such as Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ES) [7, 8, 9, 10, 11]. The latter group applies machine learning algorithms such as Neural Networks(NN)[12, 13, 14], Support Vector Regression (SVR)[15], $k$ Nearest Neighbors ($k$-NN) [16, 7].

This thesis is concerned with developing methods for predicting the power output of solar PV systems. In particular, we focus on directly and simultaneously predicting the 24h-ahead solar power output at 30-min intervals. This forecasting horizon is frequently used and allows sufficient time for the PV plant and electricity market

operators to evaluate the situation and make decisions. All case studies in this thesis consider this prediction task. We utilise different data sources, e.g. historical PV power data, historical weather data and weather forecasts. The specific data sources used in different case studies are described in the relevant chapters. The aim of this thesis is to investigate the performance of existing state-of-the-art machine learning methods for solar power forecasting and develop novel methods with improved performance.

## 1.1 Main Contributions

This thesis focuses on using machine learning approaches to directly and simultaneously predict the PV power output for the next day at 30-min intervals. We analyze the limitations of the state-of-the-art methods in this area and and propose new methods to address these limitations and improve the accuracy. The main contributions of this thesis can be summarized as follows:

1. **Instance-based methods.** We propose two new instance-based methods for solar power forecasting, namely DW$k$NN and extended PSF. DW$k$NN is an extension of $k$-NN, which considers the importance of different data sources (historical PV power data, historical weather data and weather forecasts) and learns the best weights for them based on previous data. PSF1 and PSF2 are extensions of the standard PSF algorithm which is only applicable to a single data source (historical PV data) to deal with time series from more than one data source. Our evaluation using Australian data showed that the proposed extensions were more accurate than the original methods they extend.

2. **Clustering-based methods.** We propose two novel clustering-based methods for solar power forecasting that partition the days into groups with similar weather characteristics and then build a separate prediction model for each group. The first method, direct clustering-based method, groups the days into clusters based on their weather profiles and then trains a separate model for each cluster. The second method, weather pair patterns clustering-based method, considers the weather type transition between two consecutive days and then builds a separate prediction model for each type of cluster transition. We evaluated the performance of the two clustering-based methods and compared them with methods

without clustering. The results showed that the clustering was beneficial for solar power forecasting, resulting in higher accuracy.

3. **Ensemble methods.** In addition to building single prediction models, we also investigated ensembles of prediction models to improve the forecasting accuracy. In particular, we investigated ensembles of NNs that use only PV power data since weather data may not always be available for the location of the PV plant. We propose three strategies for creating static ensembles based on random example and feature sampling, and several strategies for creating dynamic ensembles by adaptively weighting the contribution of the ensemble members based on their recent performance. We also proposed another version of the dynamic ensemble called EN-meta which uses meta-learning and predicted performance for the new day instead of actual performance on previous days to calculate the weights of the ensemble members. Our evaluation results showed that proposed ensembles outperformed the single models and classical ensembles used for comparison. The dynamic ensembles were more accurate than the static ensembles, with EN-meta being the most accurate prediction model.

## 1.2 Publications Associated with the Thesis

The following publications are associated with this thesis:

1. Zheng Wang, Irena Koprinska and Mashud Rana (2016). Clustering Based Methods for Solar Power Forecasting, in Proceedings of the *International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, July 2016, IEEE press. **CORE ranking: A**

2. Zheng Wang, Irena Koprinska and Mashud Rana (2017). Solar Power Prediction Using Weather Type Pair Patterns, in Proceedings of the *International Joint Conference on Neural Networks (IJCNN)*, Anchorage, May 2017, IEEE press. **CORE ranking: A**

3. Zheng Wang and Irena Koprinska (2017). Solar Power Prediction with Data Source Weighted Nearest Neighbors, in Proceedings of the *International Joint Conference on Neural Networks (IJCNN)*, Anchorage, USA, May 2017, IEEE press. **CORE ranking: A**

4. Zheng Wang, Irena Koprinska and Mashud Rana (2017). Solar power forecasting using pattern sequences, in Proceedings of the *International Conference on Artificial Neural Networks, ICANN 2017*, Alghero, Italy, September 2017, Springer LNCS. **CORE ranking: B**

5. Zheng Wang, Irena Koprinska, Alicia Troncoso and Francisco Martinez-Alvarez (2018). Static and Dynamic Ensembles of Neural Networks for Solar Power Forecasting, in Proceedings of the *International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, July 2018, IEEE press. **CORE ranking: A**

6. Zheng Wang and Irena Koprinska (2018). Solar Power Forecasting Using Dynamic and Meta-Learning Ensemble of Neural Networks, in Proceedings of the *International Conference on Artificial Neural Networks (ICANN)*, Rhodes, Greece, October 2018, Springer LNCS. **CORE ranking: B**

## 1.3 Thesis Structure

The rest of the thesis is organized as follows:

Chapter 2 provides a comprehensive review of the previous research work on solar power forecasting. Section 2.1 summarizes meteorological models which use NWP and satellite image processing techniques to make indirect predictions for the PV power output. Section 2.2 reviews statistical models using ARMA, ARIMA and ES. Section 2.3 provides an overview of the state-of-the-art machine learning methods used in this area including single models, clustering-based models and ensembles of prediction models. Section 2.4 reviews hybrid systems for solar power forecasting.

Chapter 3 investigates instance-based methods for solar power forecasting and proposes two novel method: DW$k$NN and extended PSF. Section 3.1 introduces the DW$k$NN model, which extends the standard $k$-NN method. Section 3.2 proposes two extended PSF models. Both DW$k$NN and the extended PSF models are evaluated using Australian data in sections 3.1.2 and 3.2.2 respectively. The publications related to this chapter are publication 3 and 4 from Section 1.2.

Chapter 4 is concerned with clustering-based methods for solar power forecasting. Section 4.1 proposes a new approach based on direct clustering, while Section 4.2

introduces the weather type pair pattern clustering-based methods. The performance of the proposed methods is evaluated and discussed in sections 4.1.2 and 4.2.2 respectively. The publications associated with this chapter are publication 1 and 2 from Section 1.2.

Chapter 5 focuses on exploring the potential of ensembles of prediction models for PV power forecasting. Section 5.1 introduces strategies for creating static and dynamic ensembles based on previous performance. Section 5.2 introduces a dynamic ensemble based on predicted performance for the new day which uses meta-learners. The performance of the proposed methods is evaluated and discussed in sections 5.1.3 and 5.2.2 respectively. The publications related to this chapter are publication 5 and 6 from Section 1.2.

# Chapter 2

# Literature Review

As described in Chapter 1, our main research task is to employ machine learning methods to make direct forecasts for the PV power output for the next day. The methods used for solar power forecasting can be generally classified into four categories:

1. Meteorological models - These methods ate typically indirect. They use Numerical Weather Prediction(NWP) techniques and satellite image processing to first forecast the solar radiation intensity and then convert it into PV output data.

2. Statistical models - These methods usually use statistical methods such as Auto-Regressive Moving Average (ARMA), Auto-Regressive Intergrated Moving Average (ARIMA) as well as Exponensial Smoothing (ES). These models can be used to make direct forecasts for the PV power outputs, without the need to firstly forecast the solar irradiance.

3. Machine learning models - These methods use machine learning algorithms such as *k*-NN, Neural Networks (NN), Support Vector Regression (SVR) and Pattern Sequence-based Forecasting (PSF), to directly forecast the PV power output. There are generally two ways to utilize machine learning techniques: by building a single prediction model or grouping several prediction models together to form an ensemble of prediction models.

4. Hybrid models - These methods combine models or different components from the previous three categories. Slightly different from the ensembles which typically combine machine learning models, the hybrid models usually combine

meteorological models with machine learning and statistical models or components together.

In the next sections (Section 2.1-2.4), we review the related work in these four categories. The limitations of the current methods and the motivation for our work is discussed in Section 2.5.

## 2.1 Meteorological Models

A traditional way for forecasting the solar irradiance is to construct physical satellite model by measuring local and global meteorological data, and then modeling the relationship between solar irradiance and other factors such as temperature, humidity, rainfall values, etc. This process typically needs to convert digital counts from the satellite-based radiometers into flux density, which requires appropriate calibration [17].

### 2.1.1 Numerical Weather Prediction

The NWP models are usually built on numerical integration equations which require domain knowledge to explain the radiation mechanism and the variations in the atmosphere.

In [18] Cornaro et al. pointed out that the key advantage of NWP is that it is a deterministic physical model. However, the authors also indicated that the NWP model is limited by the non-linearity of the domain equations as well as the insufficient spatial resolution of the integration grid, from 100km to a few km, which is too wide compared to the PV plant size. In [19, 20, 21], the spatial resolution of NWP models is discussed. NWP models can be classified into global and mesoscale models. Due to the coarse resolutions, NWP models do not allow the detailed mapping of small-scale features. Although the NWP resolution is improved in recent years, the range of resolutions still lies in 16-50 km depending on the models, which undermines the accuracy of forecasting.

In terms of the temporal scales, Lorenz and Heinemann [22] indicated that NWP models are widely used to predict atmospheric states up to 15 days ahead and this shows the limitation of using NWP models for longer-term forecasts. In summary, the

accuracy of NWP models depends on the availability of meteorological records and NWP models perform better when applied to short-term prediction tasks.

### 2.1.2 Satellite Image Processing

Another type of meteorological models used to forecast solar power is based on analysing images captured by digital cameras or satellites [23].

Most existing models which use digital cameras were designed to take images of the hemispheric view of the sky and capture the cloud movement. The movement is then vectorised and used to predict the short-term cloud cover, irradiance and solar power. The efficiency of the cloud tracking and detection techniques is significantly influenced by the ways the cameras are set up [24].

Chow et al. [25], proposed a method for intra-hour, sub-kilometer cloud irradiance forcasting using a ground-based sky imager at the University of California, San Diego. They took sky images every 30s and processed the images to determine the sky cover using a clear sky library and sunshine parameters. They generated a two-dimensional cloud map from coordinate-transformed sky cover to estimate cloud shadows at the surface, which is further used to make the forecasts. The accuracy of the forecasts was mainly influenced by two factors: cloud speed and forecast horizon. The results showed that in the 30s forecasts, the forecasting error was reduced to 50%-60% of the error of the persistence models.

Peng et al. [26] developed a short-term solar irradiance estimation for novel 3D cloud detection and tracking system based on multiple sky imagers. They trained a classifier to recognize clouds at a pixel level as well as the output cloud mask. Then, they measured the block-wise base height and the motion of every cloud layer based on the images captured from the multiple sky imagers, ready to be combined together into larger views for solar prediction. Compared with the persistence model, the proposed model achieved a minimum 36% improvement for all irradiance predictions between 1 min and 15 min intervals.

In addition to images captured by digital cameras from the ground, images captured by satelites were also utilized. The use of satellite images is similar to the use of images captured by cameras on the ground. The cloud pattern is captured and deduced from both the visible and infrared images taken by the satellite sensors flying overhead.

In [27], Marquez et al. developed the Global Horizontal Irradiance (GHI) predictions at temporal horizons of 30, 60, 90 and 120-min using a hybrid technique of satellite image analysis and NNs. The cloud fraction parameters were collected and used as the NN inputs. The proposed method outperformed the persistence model by 5-19% for 1 time-step forecasts and by about 10-25% for multi-step forecasts. Similar findings can also be found in [28].

Aguiar et al. [29] proposed a satellite-derived ground data model using solar radiation and total cloud cover forecasted by European Center for Medium-Range Weather Forecasts (ECMWF) to improve the intra-day solar prediction. They used a clear sky index as a solar radiation parameter with statistical models. A NN was trained with ground and exogenous data as inputs such as history GHI, air temperature and ground relative humidity.

The results showed the combination of NN and ECMWF was beneficial, compared to using NN alone. It improved the RMSE with 15.47%-22.17% for the Co-Pozo Izquierdo station and 25.15%-34.09% for the C1-Las Palmas station.

## 2.2 Statistical Models

Another important group of methods used to predict the PV power output or solar irradiance includes statistical models such as Autoregressive Moving Average (ARMA), its extension, Autoregressive Integrated Moving Average (ARIMA), and Exponential Smoothing (ES). These statistical models are usually applied to short-term (within a day) solar power forecasting tasks. Compared to the meteorological forecasting models, statistical models can be directly used to forecast the PV power outputs. This reduces the reliance on domain knowledge about power engineering and PV systems.

### 2.2.1 ARMA and ARIMA

The general procedure of using ARMA for time series forecasting tasks is as follows [30]:

1. The input data is collected. If the data is non-stationary, then a transformation is conducted to make the data stationary.

2. The model order $(p, d, q)$ is identified, estimated and fitted. If the model is not adequate, the model order is modified until it becomes adequate.

3. The model is used to forecast as per the desired horizon.

Pedro and Coimbra [7] evaluated five forecasting models with non-exogenous inputs. They compared ARIMA with a persistent model, standard *k*-NN, standard NN and a NN optimized by Genetic Algorithms (GA-NN) and tested the accuracy of these models using the data for eight months. Even though the results showed that GA-NN outperformed the other methods used for comparison, ARIMA also showed satisfactory accuracy.

Agoua et al. [31] constructed a statistical spatio-temporal method to forecast the power output from several minutes ahead up to 6h ahead. They introduced a new stationarization process to overcome the issue of non-stationarity of the time series. The results show that this pre-processing was beneficial, resulting in better performance compared to using the raw data. They also indicated that including meteorological variables such as wind power contributes to the improvement of the spatio-temporal model. Compared with a persistence model, random forest and AR, the proposed model achieved a 20% higher accuracy.

Yang et al. [32] proposed three forecasting methods to predict the next hour solar irradiance values and the cloud cover effects. The proposed three methods take different types of meteorological data as input. The first method takes in global horizontal irradiance (GHI) values and directly uses it to forecast the GHI values at 1-hour intervals through additive seasonal decomposition, followed by an ARIMA model. The second method forecasts disffuse horizontal irradiance (DHI) and direct normal irradiance (DNI) separately using additive seasonal decomposition, followed by an ARIMA model. The results of the two forecasts are then combined to predict GHI using an atmospheric model. The third method considers cloud cover effects and uses ARIMA to predict cloud transients. The final forecasts is made by non-linear regression techniques which uses GHI at different zenith angles and under different cloud cover conditions. Their results showed that the third method outperformed the other two, leading to MRE = 0.39 and 0.27, RMSE = 29.73 and 32.80 for the Miami and Orland test sets respectively. The results showed that the use of cloud cover techniques improves the performance of the forecasting models.

Yang et al. [33] proposed AR with eXogenous Input based Statistical model (ARX-ST) to improve the accuracy of PV power forecasting models. The model takes local PV data as well as geographically correlated information of solar PV production from other sites as inputs and can be applied to forecasting tasks for multiple horizons. The results showed that the proposed model was the most accurate for 1-h and 2-h ahead forecasting. For 1-h ahead, MAE of the ARX-ST model was 50.79%, 41.8% and 5.15% lower than the persistence, backpropagation NN, and the AR model, respectively. For 2-h ahead, the results were 60.2%, 47.27% and 8.09% ,respectively.

Li et al. [11] pointed out that the standard ARIMA for solar power forecasting considers only the solar power data and fails to take into account the weather information. Hence, they proposed a generalized model, ARIMAX, which allows for exogenous inputs for forecasting power output. The exogeneous inputs of the model are temperature, precipitation amount, insolation duration and humidity, which can be easily accessed. They also indicated that the proposed model is more general and flexible for practical use than the standard ARIMA and improves the performance of the latter based on the experiment results. Their results showed a 36.46% improvement in RMSE, showing that weather information can be used to enhance the performance of ARIMA for solar power forecasting.

### 2.2.2 Exponential Smoothing

Exponential Smoothing (ES) is a very popular and successful statistical method for forecasting energy time series data such as electricity demand and wind power forecasting [34, 35, 36, 37]. It computes the prediction as a weighed combination of the previous values, where the more recent values are weighed higher than the older. The Holt-Winters ES is an extension of the standard ES for data with seasonality. This method has also been applied for solar power forecasting.

Yang et al. [10] proposed three time series decomposition based models to forecast the hourly global horizontal irradiance (GHI) values. The first model implemented an additive seasonal-trend decomposition as a pre-processing technique before ES was used, which reduces the state space and hence improves the computational efficiency. The second model decomposed the GHI time series into a direct component and a diffuse component. Both components were used to make forecasts and their results were combined using the closure equation, forming the final forecasts for GHI. The third

method considered also the cloud cover index and applied ES to the cloud cover time series to obtain the cloud cover forecasts. Then, the GHI was forecast through polynomial regressions. The results showed that all models outperformed the persistence models.

Dong et al. [9] proposed a Exponential Smoothing State Space (ESSS) model to forecast high-resolution solar irradiance time series. They first built a Fourier trend model to stationarize the irradiance data. This was compared with other state-of-the-arts trend methods using residual analysis and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) stationarity test. Then an ESSS model was implemented to forecast the stationary residual series of the testing data. They compared the performance with ARIMA, linear exponential smoothing, simple exponential smoothing and random walk models.Their results showed that ESSS generally outperformed the methods used for comparison.

## 2.3 Machine Learning Models

The third group of models used for solar power forecasting are based on machine learning and artificial intelligence technique. Most of the state-of-the-arts forecasting models use Neural Networks (NNs), $k$-Nearest Neighbors ($k$-NN) and Support Vector Machines (SVM). These models are data-driven and do not require strong domain knowledge of power engineering as required by the meteorological models. Machine learning forecasting models can be used for directly forecasting the PV power, without the need to first forecast the solar irradiance and then convert it to power output. Another benefit of this group of methods is the flexible forecasting horizon. Most of the meteorological and statistical models introduced in Section 2.1 and Section 2.2 are suitable for short-term or very short-term forecasts, which are usually intra-day forecasts [38, **?**, 10, 33], while the forecasting horizon of the machine models can be more flexible, ranging from intra-hours forecasts[39, 40] to next day forecasts [41, 42].

### 2.3.1 $k$-NN

$k$-NN is a popular instance-based method, that has been successfully used for solar power prediction tasks [43, 44, 45].

Pedro and Coimbra [7] implemented a *k*-NN model and showed that it outperformed the persistence model used for comparison. In [16] they also proposed a new *k*-NN based methodology to forecast intra-hour GHI and DNI, as well as the corresponding uncertainty prediction intervals. The forecasting horizon ranged from 5 min up to 30 min, and the parameters were determined based on an optimization algorithm. The results showed that the proposed model achieved 10% - 25% improvement over the persistence model. The authors also indicated that including sky images in the optimization can lead to a small improvement of about 5%. In [46], they studied the influence of different types of climates into the forecasting perforamnce and proposed *k*-NN and NN based models to forecast the global irradiance. The two models were optimized by using feature extraction methods and the results showed that the proposed models significantly improved the persistence models.

In [47] Chu et al. extended a *k*-NN model using NN-optimized re-forecasting method. This model was evaluated using the data from a 48MW PV plant and their results showed that the reforecasting method could significantly improve the performance of *k*-NN for time horizons of 5, 10 and 15 min.

Chu and Coimbra [48] proposed *k*-NN ensemble models using lagged irradiance and image data to generate probability density function forecasts for intra-hour Direct Normal Irradiance (DNI). The model took diffuse irradiance measurements and cloud cover information as exogenous feature inputs and was evaluated using data from different locations (continental, coastal and island) by metrics such as Prediction Interval Coverage Probability (PICP), Prediction Interval Normalized Averaged Width (PINAW) and other standard error metrics. As baselines they implemented a persistence ensemble probabilistic forecasting model and a Gaussian probabilistic forecasting model. Their results showed that the proposed *k*-NN ensembles outperformed both reference models in terms of all evaluation metrics for all locations when the forecasting horizon was longer than 5 mins.

In [49] Chen et al. proposed a methodology to forecast hourly GSI values. More specifically, they trained a *k*-NN model to preprocess the data prior to training a NN to forecast the 1 hour ahead GSI value for the target PV station. The *k*-NN model uses meteorological data from 8 adjacent PV stations and generates the inputs for the NN model, which is used to make the forecasts. The results showed that the hybrid model achieved Mean Absolute Bias Error (MABE) of 42 W/m$^2$ and RMSE of 242 W/m$^2$.

Martinez et al. [50] proposed the Pattern Sequence similarity Forecasting (PSF)

method to predicting energy-related time series. PSF first clusters the historical data into several groups and labels the days with their cluster label. The days prior to the target day form a pattern sequence of cluster labels. PSF then searches the historical data for nearest neighbours of these pattern sequences, and uses the days immediately after the neighbor sequences to compute the forecast for the new days by taking the average of their values. The results showed that PSF was successful and efficient method for making forecasts.

### 2.3.2 NNs

NNs are the more frequently used methods for solar power forecasting tasks [13, 51, 52, 53]. NNs can be used to solve complex non-linear problems but they require careful parameter selection, including NN structure and training algorithm [54, 55].

Pedro and Coimbra [7] compared *k*-NN, NN, ARIMA and a persistence model and showed that NN can provide more accurate forecasts for solar power data. They also indicated that the NN can be optimized by Genetic Algorithms, forming GA-NN, which further improves the performance of NN. Izgi et al. [12] proposed an NN model to predict the solar power output of a 750W solar PV panel and compared different forecasting horizons. Their results showed that the best forecasts for short-term and middle-term forecasting horizons were for 5 min and 35 min respectively in April, and for 3 min and 40 min respectively in August.

Chen et al. [56] proposed a forecasting model based on fuzzy logic and NN. It takes as an imput the historical hourly solar irradiation, sky conditions and average hourly temperature, and predicts the solar irradiation values for the next month. Fuzzy logic was used to classify temperature and sky conditions before using them in the NN. An evaluation under different sky conditions was conducted, achieving Mean Absolute Percentage Error (MAPE) ranging from 6.03% to 9.65% for the different cases.

Kardakos et al. [57] compared seasonal ARIMA implemented with solar prediction derived from an NWP model and NN with multiple inputs to predict the PV power for both intraday and day-ahead horizons at 1 hour intervals. Their findings showed that the Normalized Root Mean Square Error (NRMSE) of NN model was lower than that of the ARIMA and the persistence model.

Mellit et al. [58] proposed two models based on NNs to forecast solar power generated by 50 Wp Si-Polycrystalline PV modules. The inputs of the models were solar

irradiance and air temperature. The first NN based model was trained to predict solar PV power in cloudy cases with an average daily solar irradiance less or equal to 400 $Wm^{-2}$/day, and the second model was trained to predict sunny days with the average solar irradiance exceeding 400 $Wm^{-2}$. The second NN model showed better performance, achieving MBE = 0.94% - 0.98% and RMSE less than 0.2%. The work showed that the performance of NN models may vary under different weather conditions and explored the possibility of training separate classifiers for different weather conditions.

The same inputs (solar irradiance and temperature) were used in [59], where Almonacid et al. proposed a NN model to predict the 1-h ahead PV power output. The model was evaluated using linear regression analysis, which compares the actual values with the forecast values. The results showed that the proposed model achieved correlation coefficient values close to 1 and RMSE of 3.38%. Similarly, Mellit and Pavan [60] applied NNs to predict the daily solar irradiance at 60-min intervals. In addition to the average daily solar irradiance and temperature, the NN models also take as input information about the day of the month. To predict the PV power output, the predicted solar irradiance is multiplied with coefficients based on the PV panel characteristics such as area, efficiency and balance.

Dahmani et al. [61] implemented a NN model to forecast the tilted global solar irradiation derived form the horizontal data gathered from Algeira. The model uses as inputs the horizontal global extra-terrestrial irradiation at 5-min intervals, the declination, zenith angle and azimuth angle. It was was evaluated using data for 2 years showing promising results - the best relative RMSE achieved was 8.82%.

In [62] Teo et al. proposed a NN model based on extreme learning machine algorithms to directly forecast the PV power output. They used three data sets to evaluate the performance of the model. They indicated that modifying input variables and increasing the size of training samples can significantly improve the performance. This conclusion was also made in [63], where Giorgi et al. compared statistical methods based on multi-regression and Elman NN for 1 to 24h ahead PV power prediction. They pointed out that including all weather parameters as input vector provided the best prediction for PV power. However, in [64], a different conclusion was made by Notton et al. who proposed three NN models, evaluated using the data for 5 years at 10 min intervals from a PV plant in France. The first two NN models used declination, time, zenith angle, 10-min extra-terrestrial horizontal irradiation and 10-min extra-terrestrial horizontal global irradiation as inputs. The third NN model included

the inclination angle as the additional input. The results showed that the elimination of one of the input variables improved the RMSE and Relative Mean Absolute Error (RMAE) by 9% and 5.5% respectively.

Amrouche et al. [65] utilized NN and spatial modelling methods to provide daily forecasts for the local GHI values. The models takes the weather forecasts provided by the US National Ocean and Atmospheric Administration for four adjacent sites as inputs. They compared the proposed model with geometric and statistical models. The results showed that the NN-based models outperformed the geometric and statistical models used for comparison, achieving the lowest MSE and RSME.

Yona et al. [66] implemented a Recurrent Neural Network (RNN) with fuzzy logic to predict the PV power output for the next day. They used fuzzy model functions to generate insolation forecast data so that the RNN can be trained smoothly. The proposed model was shown to be more accurate than other models used for comparison (a persistence model, a model using fuzzy logic only and a feedforward NN), achieving a best MAE = 0.1327kW.

In [67] Long et al. compared four different methods: NNs, SVR, $k$-NN and Linear Regression (LR). They studied two groups of inputs: (i) historical PV data only and (ii) a combination of historical PV data and weather information. The evaluation was done on data from a PV plant in Macau, for forecasting horizons up to 3 days ahead. Their results showed that there was no single best performing algorithm for all scenarios, but overall NNs were the most successful.

Azimi et al. [68] proposed a system which combines a clustering algorithm with NNs to predict solar radiation. Firstly, they implemented the transformation based k-means algorithm to classify the time series solar power data into various sets to determine irregular patterns and outliers. Then the clustered data was used to train NNs, which were used to make the final predictions. They compared the proposed hybrid system with several statistical models including ES and ARIMA and a persistence model, showing that the proposed hybrid system was the most accurate.

In [13] Chen et al. proposed an approach which firstly uses Self-Organizing Map (SOM) to group data into three clusters using the daily solar irradiance and cloud cover information collected from NWP predictions and then train a Radial Basis Function Neural Network (RBFNN) for each cluster. RBFNN uses as input the average PV power output for the previous day and the weather forecast for the next day average daily temperature, solar irradiance, wind speed and humidity. They achieved MAPE =

9.45% for sunny days and MAPE= 38.12% for rainy days. A similar idea was followed in [69, 70], where the historical data was partitioned into several groups and then a separate model was trained for each group. The rationale behind this idea is that days with similar weather profiles may have similar PV output characteristics and therefore building separate models for different groups may improve the accuracy.

### 2.3.3 SVR

Apart from NNs, SVM [71] is another state-of-the-art machine learning algorithm that has been widely used for solar power forecasting. When applied to forecasting tasks, the SVM version for regression tasks, Support Vector Regression (SVR), is used. The SVR prediction models are usually compared with NN prediction models [41, 72, 73, 74].

Shi at al. [69] labelled the days as sunny, foggy, cloudy and rainy based on the weather report from a meteorological station and then trained a separate SVR model for each type of day, that predicts the PV power for the next day. As input they used the PV power output of the nearest day in the training data with the same label, and also the average daily temperature forecast for the next day. The highest accuracy was achieved for sunny days (RMSE = 1.57MW) and the lowest for foggy days (RMSE=2.52MW).

In [41] Rana et al. proposed a 2D-interval forecasting model using SVR, which directly forecasts the 2D-interval PV power output from historical solar power and meteorological data. The model was evaluated using Australian PV data for two years. Their results showed that SVR2D provided the most accurate forecasts compared with a number of baselines and other methods used for comparison including NN2D and two persistence models.

Mellit et al. [72] proposed a LS-SVM model to make short-term forecasts for meteorological time series. As input variables they used the wind speed, wind direction, air temperature, relative humidity, atmospheric pressure and solar irradiance. The SVR model was compared with several NN models (MLP, RBF, RNN and PNN), and the results showed that the LS-SVM model provided more accurate forecasts than the NN models.

Ramli et al. [73] compared SVM and NN for solar irradiance forecasts using data from Jeddah and Qassim in Saudi Aribia. They used direct diffuse and global solar irradiation on the horizontal surface as input data, and evaluated the models in terms of

RMSE, MRE, correlation coefficient and computation speed. The results showed that the SVM models provided higher accuracy and more robust computation, achieving MRE = 0.33 and 0.51 for the two cities, and faster forecasting speed of 2.15s.

Chen et al. [75], proposed seven SVM models with various inputs, to forecast the daily solar irradiation values. They compared the proposed models with five empirical sunshine-based models (linear, exponential, linear exponential, quadratic and cubic) which use data gathered from three Chinese stations. The SVM models produced 10% lower RMSE compared to the empirical models, showing the promise of SVM models.

In [76] Wolff et al. developed SVR models to forecast the PV power data for 15-min and 5-h ahead horizons. The model was developed as an alternative to prediction models such as NWP. Their results showed that SVR provided good results for 1-h ahead predictions, while the NWP-based models produced better period forecasts starting at 3-h ahead, with the cloud motion vector model being the most accurate model among them. They authors suggested that combining the results made by different prediction models could further improve the accuracy.

Ekici [77] proposed a LS-SVM model using RBF kernel to forecast the solar radiation values for the next day. The model used as inputs daily mean and maximum temperature, sunshine duration, and historical solar radiation of the day. The results showed that the proposed model was effective and feasible for the task.

In [74] Mohammadi et al. integrated SVM with a wavelet transform and proposed SVM-WT model to forecast horizontal global radiation for an Iranian coastal city. They combined different input parameters such as daily global radiation on a horizontal surface, relative sunshine duration, minimum ambient temperature, relative humidity, water vapour pressure and extra-terrestrial global solar radiation on a horizontal surface. The performance was compared with ARMA, NN and Genetic Programming (GP) models. The results showed that the proposed model outperformed the other models used for comparison.

Olatomiwa et al. [78] proposed the Support Vector Machine Firefly Algorithm (SVM-FFA) to forecast the mean horizontal global solar radiation values. They used sunshine duration, maximum temperature and minimum temperature as inputs. The proposed model was compared with GP and NN models, and the results showed that the proposed model achieved the best RMSE, MAPE, r and $R^2$.

In [79] Yang et al. integrated weather-based hybrid technique with SOM, SVR, Fuzzy Inference and Learning Vector Quantization (LVQ) approaches. The model

uses SOM and LVQ to categroize historical data based on the PV profile and then trains SVRs using historical solar irradiance, temperature and precipitation probability to make forecasts. The fuzzy inference approach is applied during the forecasting to choose an appropriate trained model based on weather information provided by the Taiwan Central Weather Bureau. The results showed that the proposed method outperformed the NN and SVR approaches.

### 2.3.4 Ensembles

Instead of training one single model, another idea is to utilize ensembles of prediction models which combine the predictions of several models. The idea behind this is to utilize the diversity among the ensemble members - different ensemble members may be more suitable for different situations. The diversity can be generated by varying the input data used for training, the structure of the prediction models and the types of prediction models used in the ensemble.

In [48] Chu and Coimbra proposed $k$-NN ensemble model using lagged solar irradiance and image data to generate probabilistic forecasts for intra-hour Direct Normal Irradiance (DNI). The model took diffuse irradiance and cloud cover information as exogenous feature inputs and was evaluated using data from different locations (continental, coastal and island) using standard error measures and also PICP and PINAW. A persistence ensemble probabilistic model and a Gaussian probabilistic model were used for comparison. The results showed that the proposed $k$-NN ensemble outperformed the reference models in terms of all evaluation metrics for all locations when the forecasting horizon was longer than 5-min.

Rana et al. [80] proposed two non-iterative and one iterative ensembles of NNs for forecasting the PV power output of the next day at 30-min intervals. The NN ensemble members differed in the number of hidden nodes and weight initialisation. The individual predictions were combined by taking the predicted median value for each half-hour. The evaluation using Australian data for one year, showed the the iterative ensemble was the most accurate, outperforming an SVR-based method and two persistence baselines.

Another NN-based ensemble method to directly and simultaneously forecast the PV power output for the next day was proposed in [81].

They first clustered the days based on the weather information and then built a separate NN-based ensemble prediction model for each cluster. The ensemble members were trained to predict the PV power based on the weather data as an input. The evaluation using Australian data for two years showed that the ensemble method achieved MAE=83.90 kW and MRE=6.88%, outperformed the other models used for comparison. The use of ensemble was also compared with using a single NN and shown to be beneficial.

Raza et al. [82] proposed an ensemble of NNs to forecast the one-day ahead PV power output. They created 6 ensembles, each combining 15 NNs. The final prediction was produced using a Bayesian model averaging. The single NNs belonged to three different NN types: feedforward, Elman and cascade-forward backpropagation networks, had different number of hidden neurons and used difefrent versions of the backpropagation training algorithm. The performance was evaluated using two years of Australian data, and the results showed that the use of ensembles was beneficial.

In [83] Li et al. proposed an ensemble method that builds a separate prediction model at a micro level (for each inverter) and then sums the predictions together to produce the final prediction (at a macro level). The individual ensemble members were trained using PV data only; both NN and SVR prediction models were evaluated as ensemble members. The results showed that the ensemble combining the micro forecasts was more accurate than a single macro level prediction model.

## 2.4 Hybrid Models

The previous sections discussed the three main groups of solar power forecasting methods - meteorological, statistical and machine learning. There is also some research work on combining methods from these three groups to build hybrid prediction models. We distinguish between ensembles and hybrid models: ensembles combine the predictions of machine learning models only while hybrid models combine the predictions of any type of models.

In [84] Bouzerdoum et al. proposed a hybrid model combining seasonal ARIMA and SVM to make short-term PV power forecasts. They evaluated the proposed model using data collected from a 20 kW PV plant and compared the proposed model with single seasonal ARIMA and SVM based models. The results showed that the proposed

hybrid model outperformed the single models.

Wu et al. [85] implemented a hybrid system which integrated NN, SVM, ARIMA and ANFIS using GA. They collected historical solar power, solar irradiance and temperature data, and forecasted the solar power data for three PV plants. Their results showed that the proposed hybrid system was more accurate than the single prediction models which comprised the hybrid system, achieving NRMSE = 5.64%, 3.43% and 6.57% for the three sites respectively.

Dong et al. [86] combined satellite image analysis, ES state space and NN methods in a hybrid system. The satellite image technique was primarily used to detect the cloud movements, while ES was utilized to forecast the cloud cover index. Then a NN was trained using the predicted cloud cover index values to forecast the solar irradiance. The results showed that the proposed hybrid model was more accurate than ARIMA, linear ES, simple ES and random walk.

In [47] Chu et al. proposed a re-forecasting method to forecast the intra-hour power output of a PV plant. The first step of the method was to train baseline models, including a physical deterministic model based on cloud tracking, ARMA and a $k$-NN model. Then, a NN was employed to optimize the performance of the baseline models. The results showed that the proposed method was effective, significantly improving the performance of the baseline methods for time horizons of 5, 10, and 15 min.

Dolara et al. [87] proposed a hybrid system, called Physical Hybridized Artificial Neural Network (PHANN), that combines NN with an analytical physical model - the Clear Sky solar Radiation Model (CSRM). CSRM was used to determine the time span between the sunrise and the sunset of each day, while the NN was trained to make 24-72 h ahead forecasts of PV power based on the output of the CSRM model. The results showed that PHANN outperformed the NN model without the CSRM components. A similar PHANN model was also proposed in [88] and shown to achieve most accurate results during sunny days compared to weather day types.

In [89] Filipe et al. combined statistical and meteorological methods and proposed a hybrid system to forecast the 48 h ahead solar power output of a PV plant. They first combined an electrical model of the PV system and a gradient boosting statistical model. The statistical model was used to convert NWP into solar power for short-term time horizons. Then, they employed different NWP models and combined these models with information from past PV observations. The results showed that the hybrid system outperformed two naive models (persistence model and diurnal), showing a

57.3% and 34.06% improvement, respectively.

Voyant et al. [90] proposed a hybrid model combining ARMA, NN and data from NWP to predict the hourly mean GHI. The NN uses the NWP data as an input and predicts the clear sky index for the next day. A regression-based variable selection is applied to select the inputs of the NN. The trained NN model is combined with an ARMA model to form the final prediction. The results showed that the hybrid system outperformed a single NN and a persistence model for all five tested locations in the Mediterranean area, e.g. the NRMSE of the hybrid model was 14.9% compared to 18.4% for the NN and 26.2% for the persistent model.

Marquez and Coimbra [91] developed and validated a medium-term solar irradiance forecasting method for both GHI and DNI based on stochastic learning, ground experiments and the US National Weather Services (NWS) database. They used GA to select the most relevant input variables for the NN. The results showed that the developed forecasting models improved the RMSE for GHI by 10-15% compared with the reference model.

## 2.5 Discussion

The previous sections introduced the state-of-the-art methods used for solar power forecasting. These methods have been classified into four groups: meteorological, statistical, machine learning and hybrid. Each group has its own strengths and weaknesses that have to be taken into consideration when applying these models in practice.

The meteorological methods for solar power forecasting are indirect methods which heavily rely on forecasts of meteorological variables such as temperature, solar irradiance, humidity, solar angle, wind speed and cloud cover index The ability to predict weather variables and weather changes is useful for solar power forecasting, e.g. the movement of clouds directly affects the output of PV panels and the air temperature influences the conversion rate of PV panels. Due to this reason, meteorological models based on NWP data and satellite images have been widely used. The former integrate global or local meteorological information which can influence the fluctuation of PV power output, while the latter can be more effective for predicting the movement of clouds.

However, the limitations of the meteorological models for solar power forecasting should not be underestimated. As discussed before, their success depends on the availability of accurate weather forecasts which may not be available for the location of the PV plants. The absence of recording of variables such as wind speed and cloud cover index or the insufficient accuracy of the forecasts for those variables may lead to significant decrease in the forecasting accuracy. This limits the practical application of meteorological models for solar power forecasting.

Another limitation of these methods is that the application and modification of meteorological models requires strong domain knowledge of meteorology and power engineering. For instance, modifications in the NWP models cannot be successfully implemented by forecasting engineers without a clear understanding of the complex meteorological data models. This limits the ability of forecasting engineers to fully utilize these models. Instead, they can only assume that the forecasts made by NWP are sufficiently accurate and use the results to make the next-step indirect forecasts for the PV power output. Finally, many meteorological models can only perform well for short or very short time horizons [25, 27, 29]. This is due to the fact that the fundamental techniques for such models, e.g. NWP or satellite image processing, are more accurate for intra-hours and intra-day tasks. Forecasts for several hours ahead forecasts may not be accurate when clouds are quickly forming and dissipating. This short forecasting horizon limits the applicability of meteorological models and makes them less suitable for tasks with longer forecasting horizons such as 24, 48 and 72-hours ahead. However, 1-day ahead forecasts are common in industry as they allow the PV plants sufficient time to make operational and maintenance decisions.

The statistical group of methods employs algorithms such as AR, MA, ARMA, ARIMA, seasonal ARIMA and ES, and can be directly used to forecast the PV power output. Compared with the meteorological models, they do not heavily rely on the availability of accurate weather forecasts, meteorological or power engineering knowledge. As a result, data scientists often employ these models and tune them to improve the prediction. Another benefit of statistical models over meteorological models is that they can be used to make direct predictions for the PV power instead of being used to first predict the solar irradiance and then convert it into power output. This simplifies the forecasting process and typically improves the accuracy since the conversion rate of PV panels is not invariable and can be influenced by other factors such as temperature. However, similar to the meteorological methods, most statistical methods are

more suitable for short-term forecasts [31, 32, 33] such as intra-day, and when the horizons are extended to a day ahead level, their accuracy decreases. This also restricts the practical application of statistical models.

Another point to notice is that statistical models usually focus on the target time series, namely the PV power output data, and are not well suited to deal with unexpected changes in the weather conditions especially over a very short period, which is the advantage of meteorological models. Finally, statistical models as well as meteorological models have been widely used for one-step ahead forecasts. However, these models are not suitable for making simultaneous forecasts for more than one time stamp, e.g. for all half-hours of the next day.

Machine learning methods are also widely used for making forecasts for the PV power output. Their use also do not require strong domain knowledge of meteorology and power engineering, and this means that more data scientists can use them. Another benefit of the machining learning models over the meteorological and statistical ones is the flexible forecasting horizon. Machine learning models are able to deal with both intra-hour, day-ahead and even month-ahead forecasts [7, 57, 56], which offers more flexibility for the PV plants to arrange operations. Also, machine learning models can be trained to forecast the PV power output at a certain time stamp in the future or provide forecasts for the values of certain intervals of the next day simultaneously, which is also very useful for supporting operational decisions at PV plants and electricity markets.

Machine learning methods are also flexible and can be used for both direct and indirect prediction of the PV power output. Similar to statistical models, they can make predictions using only the target PV power time series when weather information is not available. This extends their applicability as reliable weather data is not always available for the location of the PV plant. Machine learning models can also easily integrate more input features and this may help to improve the the performance of the models [61, 64].

However, there are several areas of the previous applications of machine learning methods to solar power forecasting that can be further investigated and improved. Firstly, most of the previous work using instance-based methods such as $k$-NN and PSF either failed to utilize all data sources or just treated them as equally important. There is a need to extend these instance-based methods to use more than one data source and consider the importance of the data sources. Secondly, although some of the previous

work has used clustering methods to group the days based on their weather and build a separate prediction PV power model for each cluster [69, 13, 79], there is still scope to improve these methods. For example, the previous work focused on single days and did not consider the continuity between days. Thirdly, most of the previous work used single machine learning methods. There is a need to investigate ensembles of prediction models, both static and dynamic.

In this thesis, we investigate the use of using machine learning techniques to directly and simultaneously predict the PV power output for the next day. More specifically, we focus on three main aspects which were not fully addressed by previous work.

1. **Investigating instance-based methods**. Previous work either failed to take advantage of multiple data sources or consider the importance of different data sources. We aim to extend the $k$-NN and PSF instance-based machine learning models to address this. This work is discussed in Chapter 3.

2. **Investigating clustering-based methods**. Previous work did not fully utilize clustering techniques for solar power forecasting tasks and did not consider the continuity between days. We aim to fully investigate these limitations. This work is discussed in Chapter 4.

3. **Investigating ensembles**. We propose, evaluate and compare a number of strategies for constructing static and dynamic ensembles. Dynamic ensembles for solar power forecasting, in particular, haven't received enough attention in previous research. This work is discussed in Chapter 5.

# Chapter 3

# Instance-based Methods

This chapter introduces two instance-based methods for solar power prediction, namely Data Source Weighted $k$ Nearest Neighbors and Extended Pattern Sequence Forecasting. These methods extend the standard k-Nearest Neighbor ($k$-NN) and Pattern Sequence Forecasting (PSF) algorithms, respectively, in order to utilize data from multiple sources.

## 3.1 Data Source Weighted $k$ Nearest Neighbors

$K$-NN is an instance-based prediction algorithm that can be applied to both classification and regression tasks. It has been applied to various time series prediction tasks, including energy related tasks such as electricity load and solar energy prediction [44, 45, 16, 46]. These approaches focused on a single data source, the target time series. Other approaches [13, 92] considered more than one data source to make forecasts, but treated these data sources as equally important.

Our aim is to develop a nearest neighbor method that: 1) uses multiple data sources and 2) assigns weights to them based on their importance for the prediction.

For solar power forecasting, there are three main data sources: historical PV data (PV), historical weather data (W) and weather forecast for the new day (WF). They can be used to represent a single day and find neighbors. For example, below we describe how the $k$-NN algorithm can be adapted and used in three scenarios - with WF, PV+W and PV+WF as feature vectors:

1. ***k*-NN using weather forecast (WF) as a feature vector**:

   When the weather forecast for the new day *d+1* is available, the features from the weather forecast can be used in a feature vector representing this day. Then, this feature vector is compared with the weather vectors in the historical data to find the neighbours. The PV vectors of the neighbours are averaged to form the PV prediction for day *d+1*.

2. ***k*-NN using historical data (PV+W) as a feature vector**:

   In this case, the PV and weather vector of the previous day can be used to find neighbours. For example, to predict $\widehat{PV_{d+1}}$, the PV data of day *d+1*, the the weather data $W_d$ and PV data $PV_d$ of the previous day *d* can be collected as a feature vector and used to find neighbours in the historical data. Then, the PV data of the days immediately after the neighbours are averaged to form the prediction.

3. **Using both historical data and weather forecast (PV+W+WF) as a feature vector**: Another way to represent the new day is by forming a feature vector that combines the PV, W and WF vectors. Specifically, to predict $\widehat{PV_{d+1}}$, the historical data for the previous day *d*, including both the weather $W_d$ and PV $PV_d$ data, and the weather forecast data for the new day $WF_{d+1}$ are collected in a vector, which is used to find neighbours. Then, the PV vectors of the days immediately after the neighbours are averaged to form the prediction.

To better utilize the three data sources and evaluate their importance for solar power forecasting tasks, we propose an extension of the standard *k*NN, called Data Source Weighted *k* Nearest Neighbors (DW*k*NN), as described below.

### 3.1.1 Methodology

DW*k*NN extends the traditional kNN algorithm by assigning weights to the different data sources based on the importance of these sources for the prediction. These weights are learned from previous data. In contrast to the traditional weighted neareast neighbor algorithms, DW*k*NN assigns weights to the features from the different sources, not to the nearest neighbours when combining their predictions.

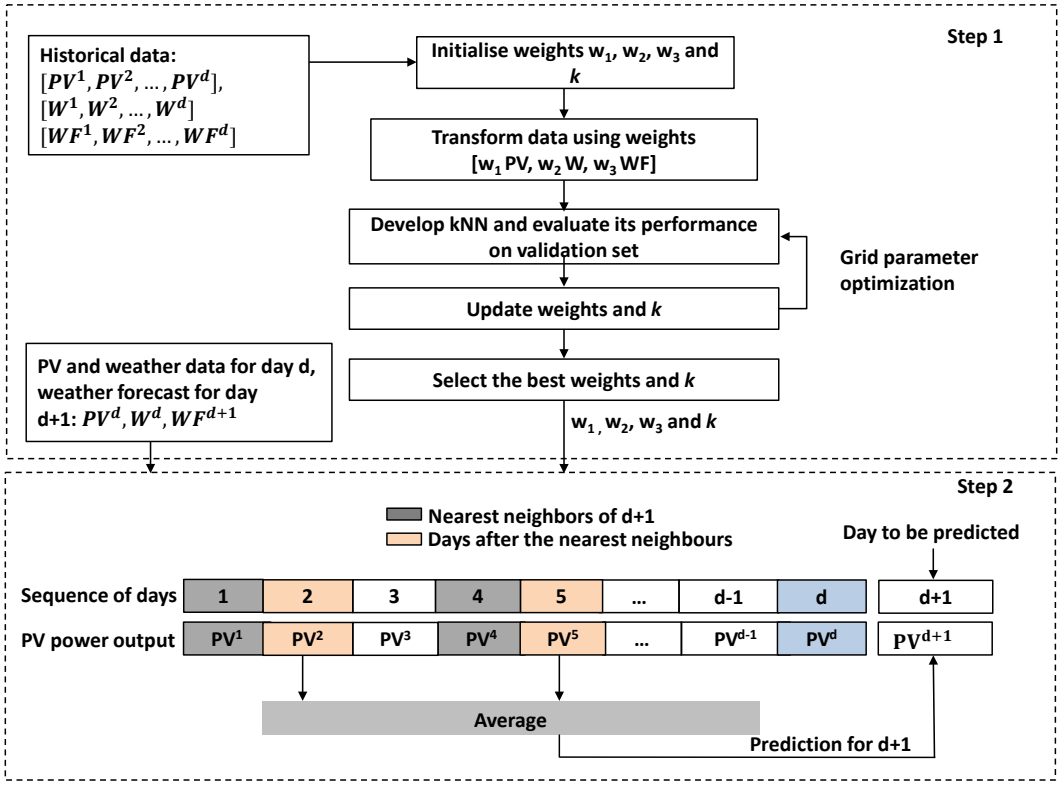


Figure 3.1: The DW$k$NN algorithm

DW$k$NN addresses the limitation of the traditional $k$-NN which doesn't consider the importance of different data sources. In this study, we focus on three data sources that are most commonly used for PV power output prediction tasks: historical PV data (PV), historical weather data (W) and weather forecast (WF).

The algorithm is summarized in Fig. 3.1 and consists of two main steps:

1. Finding the best weights for the features of the data sources and the best number of neighbors by using the historical data and applying a grid search method, and

2. Predicting the PV power output for the new day using the selected parameters from the previous step.

#### 3.1.1.1 Finding the Weights

The parameters that are optimized in the first step are the weights $w_1$, $w_2$ and $w_3$, reflecting the importance of each of the three data sources and the number of neighbors

*k*. To find the best parameters, a grid search method is first applied. The value of *k* is varied from 1 to 10 with an increment of 1, and the values of the weights are varied from 1 to 100%, with an increment of 1%. For each set of weights, the training and validation sets are transformed by multiplying the values of these sets with the corresponding weights:

$$new\ dataset = [w_1 PV, w_2 W, w_3 WF]$$

Then *k*NN, trained on the training data, is applied to predict all instances from the validation set and the accuracy is calculated. The parameters resulting in the highest accuracy on the validation set are selected.

#### 3.1.1.2 Predicting the New Day

In the second step, the PV power output for the next day *d+1* is predicted. To do this, firstly the *k* nearest neighbors of the previous day (day *d*) are selected using the chosen data weights and the number of neighbours *k* from the previous step. Then the PV power data of the days following the nearest neighbors is averaged to generate the prediction for the next day. Specifically, if $S = \{s_1, s_2, ..., s_k\}$ is the set of selected *k* days, then the prediction for $PV^{d+1}$ is given by:

$$\widehat{PV^{d+1}} = \frac{1}{k} \sum PV^{S_i+1}$$

where each $PV^{i \in S}$ is the 20-dimensional vector of half-hourly power outputs for day *i*.

To find the nearest neighbors for day *d*, four different representations of day *d* are used, depending on the data sources used, as shown in Fig. 3.2 - 3.5:

1. As shown in Fig. 3.2, when only the PV and W data sources are used $(PV + W)$, day *d* is represented as a feature vector consisting of its PV power and weather data $[PV^d, W^d]$ and it is compared with the previous days *i* represented as $[PV^i, W^i]$.

2. As shown in Fig. 3.3, when only the PV and WF data sources are used $(PV + WF)$, day *d* is represented as a feature vector including the PV power data for

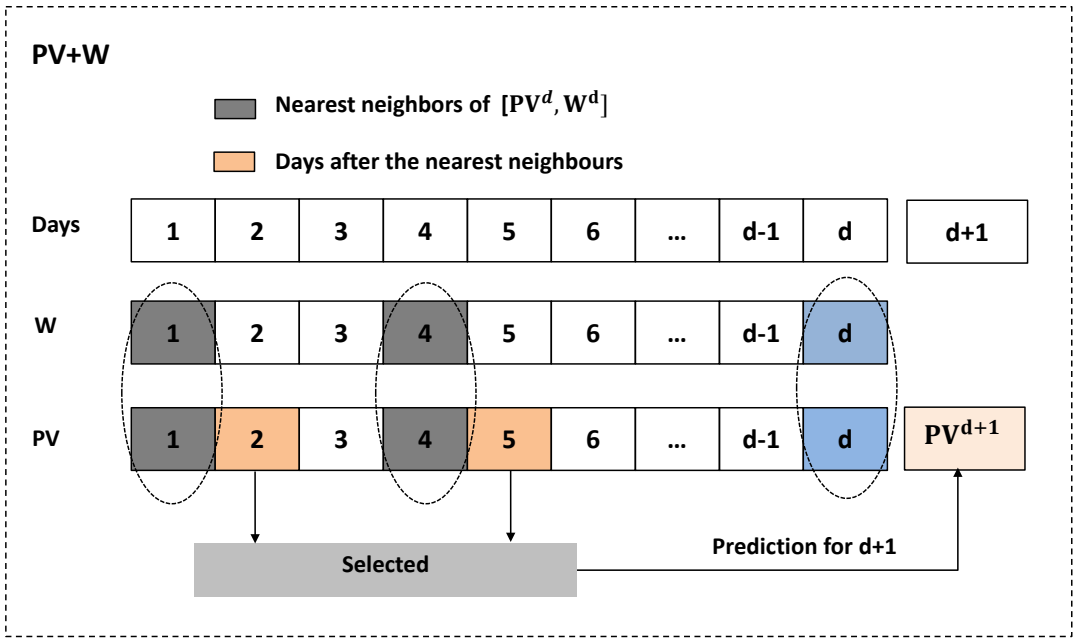


Figure 3.2: Representing days using historical PV and weather data

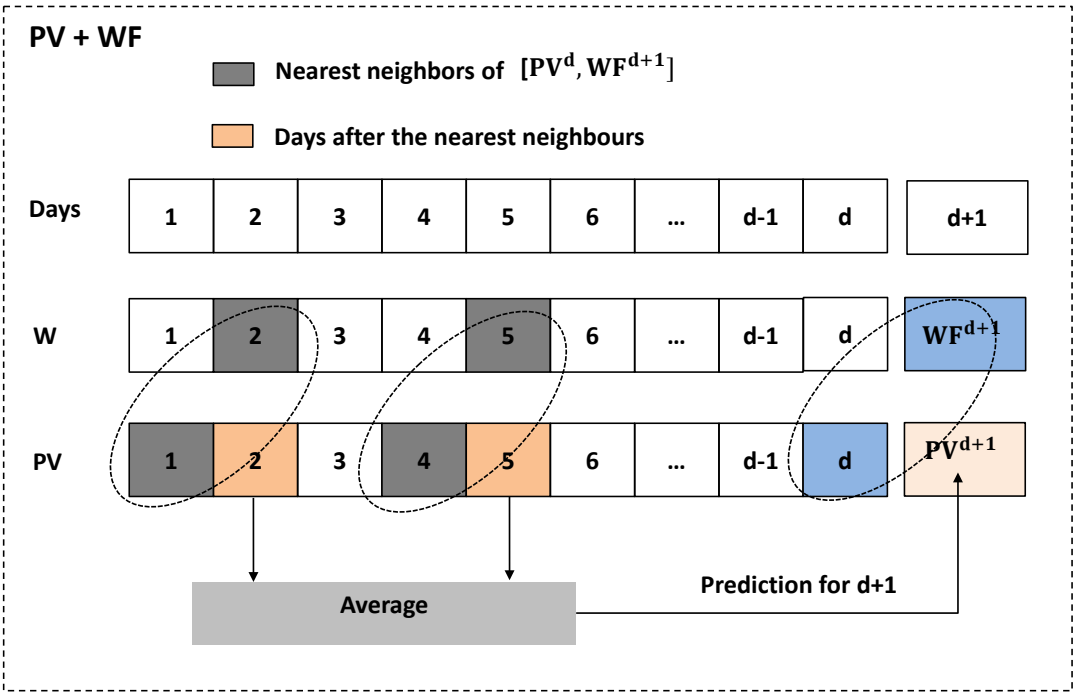


Figure 3.3: Representing days using historical PV data and weather forecast

day *d* and also the weather forecast for the next day *d+1* $[PV^d, WF^{d+1}]$, and is compared with the previous days *i* represented as $[PV^i, WF^{i+1}]$.

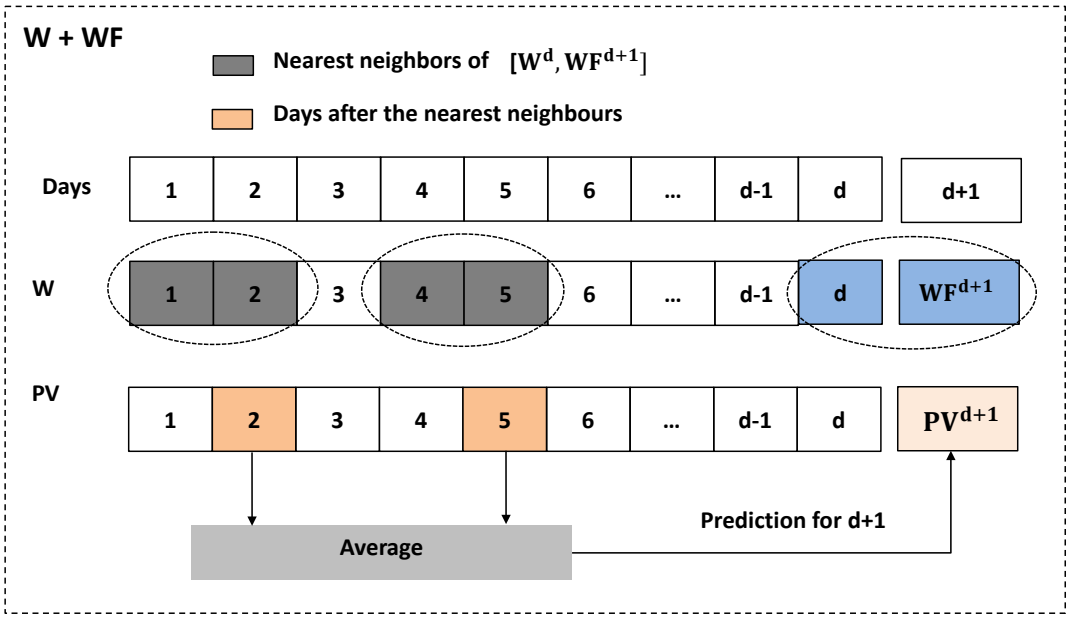


Figure 3.4: Representing days using historical weather data and weather forecast

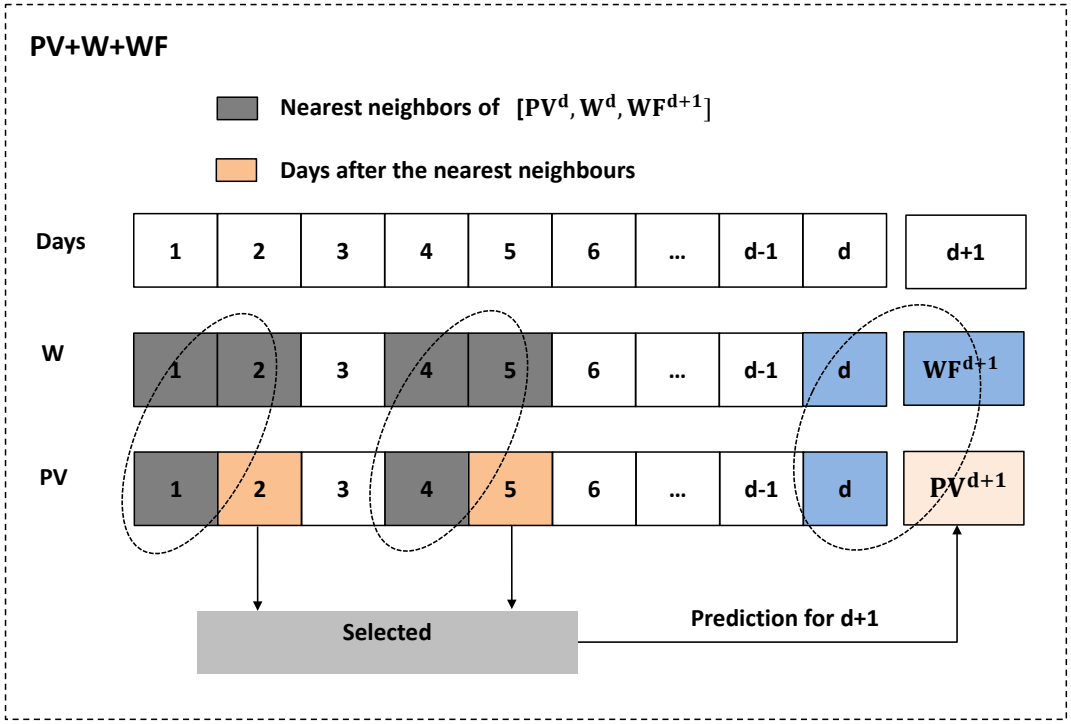


Figure 3.5: Representing days using historical PV and weather data, and weather forecast

3. As shown in Fig. 3.4, when only the W and WF data sources are used $(W+WF)$, day *d* is represented as a feature vector including the weather data for day *d* and also the weather forecast for the next day *d+1* $[W^d, WF^{d+1}]$, and is compared with the previous days *i* represented as $[W^i, WF^{i+1}]$.

4. As shown in Fig. 3.5, when all three data sources are used $(PV+W+WF)$, day *d* is represented as a feature vector including the PV power and weather data for day *d* and also the weather forecast for the next day *d+1* $[PV^d, W^d, WF^{d+1}]$, and is compared with the previous days *i* represented as $[PV^i, W^i, WF^{i+1}]$.

The nearest neighbours are found by using a suitable distance measure.

## 3.1.2 Case Study

To evaluate the performance of DW*k*NN, a case study is conducted using Australian PV and weather data for two years. The performance is compared with several other state-of-the-art methods including standard *k*-NN, Neural Networks (NN), Support Vector Regression (SVR), clustering based SVR, seasonal ARIMA and ES, as well as a persistence model as the baseline.

### 3.1.2.1 Experimental Setup

The goal for this study is to make half-hourly PV power output prediction for the next day, given historical PV power data, weather data and weather forecasts.

More specifically, given: (i) a time series of historical PV power outputs up to the day *d*: $[p^1, p^2, p^3, ..., p^d]$, where $p^i = [p^i_1, p^i_2, p^i_3, ..., p^i_{20}]$ is a vector of 20 half- hourly power outputs for the day *i*, (ii) a time series of weather vectors for the same days: $[W^1, W^2, W^3, ..., W^d]$, and (iii) the weather forecast for day *d+1* : $WF^{d+1}$, our goal is to forecast $PV^{d+1}$, the half-hourly power output for the next day *d+1*.

**A. Data Sources**

The data from the three different sources that we used is summarized in Table 3.1. The PV power data is collected from one of the largest roof-top flat-panel PV plants in Australia, located at the University of Queensland, Brisbane. It has over 5000 poly-crystalline silicon solar panels across four buildings and produces up to 1.22MW electricity.

Table 3.1: Data sources and feature sets for DW*k*NN study

| **Data source** | **Feature set** | **Num. features** | **Description** |
|---|---|---|---|
| PV data for the current day *d* and previous days | PV | 20 | Half-hourly PV values between 7am and 5pm |
| Weather data for the current day *d* and previous days | W | 14 | (1-6) Daily: min temperature, max temperature, rainfall, sun hours, max wind gust and average solar irradiance; (7-14) At 9 am and 3 pm: temperature, relative humidity, cloudiness and wind speed. |
| Weather forecast for the next day *d+1* | WF | 3 | Daily: min temperature, max temperature and rainfall |

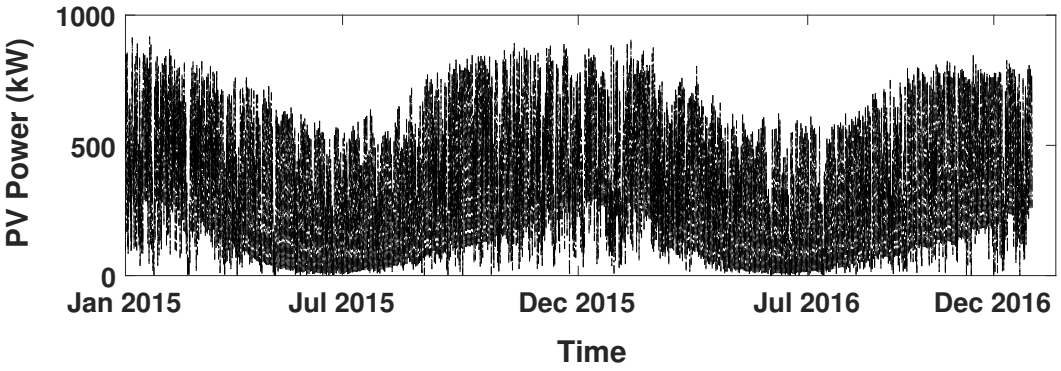


Figure 3.6: PV power data from Jan 2015 to Dec 2016

PV data for two years was collected: from 1 January 2015 to 31 December 2016 (as shown in Fig. 3.6). As expected, we can see that the PV solar power is highest for the summer months (from December to February in Australia) and lowest for the winter months (from June to August), consistent with the amount of solar irradiation. Further, only the data for a 10-hour interval during the day, from 7am to 5pm, is collected. Outside this time interval the PV power data is zero or close to zero due to the absent or very low solar irradiation, or was not available for all days. The PV power data was obtained from [93].

Weather data for these two years was also collected from the Australian Bureau of Meteorology. For each day, we collected 14 meteorological variables, as shown in Table 3.1. They include six daily measurements (minimum and maximum temperature,

rainfall, sunshine hours, maximum wind gust and average solar irradiance) and also eight measurements taken at two times during the day (9am and 3pm) - temperature, relative humidity, cloudiness and wind speed. The weather data is available from [94].

The third information source includes weather forecast data for the future days that are predicted. The weather forecast feature set WF shown in Table 3.1 includes three variables: the daily minimum temperature, the daily maximum temperature and the daily rainfall. All these are core measurements, widely available and included even in the most basic weather forecasts.

Since the weather forecasts were not available retrospectively for 2015 and 2016, we used the actual weather data with added 10% noise.

**B. Data Pre-processing**

The PV data was measured at 1-min intervals. As we predict at 30-min intervals, the 1-min data was aggregated into 30-min data by taking the average of the interval. It was then normalized to [0, 1]. In total there are 14,620 measurements $(= (365+366) \times 20)$ for the PV data.

The weather data consists of 14 daily measurements, thus there are $(365 + 366) \times 14 = 10,234$ measurements. It was also normalized to the range [0,1].

There was a small number of missing values (0.02% in the PV data and 0.82% in the weather data). To replace the missing values, the following method is applied, firstly to the weather data and then to the PV data. For missing values in the weather profile vector for day *d*, day *s* is found, the nearest neighbor of *d*, which does not have missing values. The similarity is calculated using the Euclidean distance and only the available weather features in $W^d$. We then replace the missing values in $W^d$ with the corresponding values in $W^s$.

A similar approach is followed for the missing values in the PV data. If day *d* has missing values in its PV vector, we firstly find day *s*, the most similar day to *d* in terms of weather by using the weather vectors and then replace the missing values in $PV^d$ with the corresponding values of $PV^s$.

**C. Training, Validation and Testing Sets**

The solar data and the corresponding weather data are divided into three non-overlapping subsets: training - the first 70% of the 2015 data, validation - the remaining 30% of

the 2015 data and testing - the 2016 data. The training dataset is used to build the prediction models; the validation dataset is used for parameter selection, and the testing dataset is used to evaluate the performance of the proposed method and other methods used for comparison.

**D. Performance Measures**

To evaluate the performance, we used two standard performance measures: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), defined as follows:

$$MAE = \frac{1}{D \times n} \sum_{i=1}^{n} \left| P^i - \widehat{P}^i \right|$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (P^i - \widehat{P}^i)^2}{D \times n}}$$

where $P^i$ and $\widehat{P}^i$ are the actual and forecast half-hourly PV power outputs for day $i$, $D$ is the number of days in the testing set and $n$ is the number of predicted daily values ($n$=20).

These performance measures were used in all case studies in the thesis.

**E. Distance Measure**

To find the nearest neighbors, we evaluated the performance of two distance measures on the validation set: Euclidean and Manhatten. The results showed that the Euclidean distance performed better. Therefore, this case study uses Euclidean distance.

#### 3.1.2.2 Methods for Comparison

We compare the performance of the proposed nearest neighbor algorithm DW*k*NN with the standard *k*NN, and also with: three machine learning methods using NNs and SVR, two statistical methods (ARIMA and exponential smoothing) and a persistence model used as the baseline.

**A. *k*NN**

To predict the power output for the next day *d+1*, *k*NN firstly selects the *k* neighbors

from the historical dataset whose data profiles are most similar to day $d$. It then computes the forecasts for $PV^{d+1}$ by taking the average of the PV data of the subsequent day for each of the $k$ selected similar neighbors. To find the neighbours, it uses the same day representation as DW$k$NN as shown in Fig. 3.2 - Fig. 3.5.

**B. NN**

The NN based prediction model is multi-layer perceptron with one hidden layer, trained with the the Levenberg-Marquardt version of the backpropagation algorithm. It has 20 output nodes corresponding to the half-hourly power outputs of the next day, and thus produces the 20 prediction values simultaneously. We trained two versions, each using different input data: NN (PV+W+WF) that has 37 input nodes, corresponding to the PV and weather data for the previous day, and the weather forecast for the next day, and NN (PV+W) which has 34 input nodes corresponding to the PV and weather data for the previous day.

**C. SVR**

Similarly to the NN prediction model, we developed a model based on SVR that uses the same input as NN - SVR (PV+W+WF) and SVR (PV+W). However, in contrast to NN which predicts all 20 outputs simultaneously, since SVR has one output, the SVR model consists of 20 separate SVRs, each predicting one half-hourly PV value for the next day.

**D. Clustering Based SVR**

This method groups the days into several clusters based on the weather data and then builds a separate SVR prediction model for each cluster. The rationale is that days with similar weather characteristics have similar PV power output, and by using this similarity and we can create specialized prediction models that are more accurate than a single model for all type of days. Clustering based methods have been used in [95, 70] showing good results.

We applied the k-means clustering algorithm to group the days based on their weather profile. The number of clusters was selected using clustering evaluation measures (Calinski-Harabasz, Silhouette coefficient and Davies-Bouldin). We found that the best number of clusters was two. Upon examination of the characteristics of the days in each cluster, we found that the first cluster contained clear sunny days, while

the other days (cloudy, rainy, etc.) were grouped in the second cluster.

To make a prediction for a new day, the new day is firstly mapped to the closest cluster based on its weather forecast features. This requires that the weather features used for the clustering are consistent with the available weather forecast features. Hence, for the clustering we also used the three WF features (daily minimum and maximum temperature, and daily rainfall). Then the trained SVR prediction model for the selected cluster is used to generate the prediction; it takes as an input the PV power for the previous day and outputs the PV power for the next day.

**E. Exponential Smoothing**

Exponential Smoothing (ES) is a very popular statistical method for time series forecasting. The predictions for future values are weighed combination of the previous values, where the more recent values are weighed higher than the older. We implemented the Holt-Winters ES, which is an extension of the standard ES. It decomposes the data into a trend and a seasonal component. We implemented seasonal ES with additive daily cycle since the seasonal variation for the PV power data is approximately constant. The parameters of the ES model were computed using an optimization procedure which minimizes the mean squared error for the training data.

**F. Seasonal ARIMA**

ARIMA is a well-established and widely used method for time series forecasting. It combines differencing with an Autoregressive (AR) and Moving Average (MA) components. Again, as the PV power data shows strong daily seasonality, we implemented the seasonal ARIMA, which is an extension of the standard ARIMA for modelling seasonal components. The seasonal ARIMA model can be written as: ARIMA (*p, d, q*) (*P, D, Q*) *m*, where *p* is the order of the AR lag, *q* is the order of the MA lag and *d* is the order of differencing; *m* is the length of the daily cycle (*m*=20 in our case), *P* and *Q* are the daily seasonal AR and MA lags, and *D* is the order of seasonal differencing.

To select the parameters (*p, q, d, P, Q* and *D*) of the ARIMA model for our solar power data, we experiment with different combinations of values. The best parameters were then selected by comparing the corrected Akaike Information Criterion ($AIC_c$) of the different ARIMA models.

**G. Baseline Model**

Table 3.2: Accuracy of *k*NN

| Feature set | Best *k* | MAE (kW) | RMSE (kW) |
|---|---|---|---|
| PV (20) | 7 | 127.64 | 166.15 |
| W (14) | 7 | 125.66 | 166.78 |
| WF (3) | 5 | 119.64 | 158.69 |
| PV+W (34) | 7 | 130.33 | 175.32 |
| W+WF (17) | 5 | 120.45 | 157.41 |
| PV+WF (23) | 7 | 126.20 | 164.12 |
| PV+W+WF (37) | 8 | 117.21 | 158.10 |

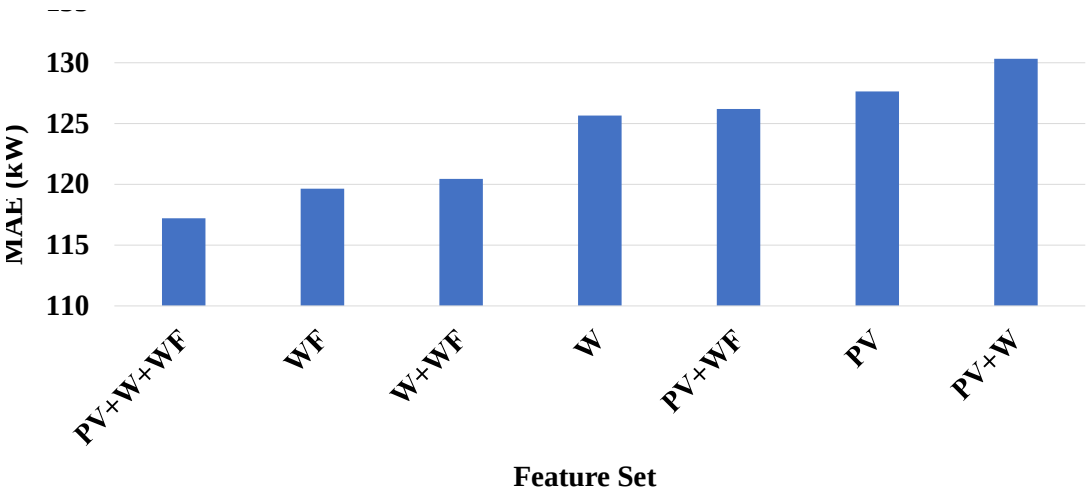


Figure 3.7: *k*NN performance (MAE) using different feature sets

As a baseline we use a persistencet model that considers the half-hourly PV power outputs from the previous day *d* as the prediction for the next day *d+1*. This means that the prediction for $\widehat{PV^{d+1}} = [pv_1^{d+1}, pv_2^{d+1}, ..., pv_{20}^{d+1}]$ is given by $PV^d = [pv_1^d, pv_2^d, ..., pv_{20}^d]$.

#### 3.1.2.3 Results and Discussion

**A. Performance of *k*NN**

Table 3.2 shows the accuracy results for *k*NN using the single and combined feature sets from the three different sources. Fig. 3.7 shows the MAE results from Table 3.2 in sorted order for visual comparison. As in DW*k*NN, the value of *k* for *k*NN was varied from 1 to 10 and the best *k* was selected based on the accuracy of the validation set.

The best results were achieved when all three data sources were combined (PV + W + WF), followed by the single WF and the combined weather and weather forecast

Table 3.3: Accuracy of DW*k*NN

| **Feature set** | **Num. features** | **Best weights and *k*** | **MAE (kW)** | **RMSE (kW)** |
|---|---|---|---|---|
| PV+W | 34 | PV=23%, W=77%, *k*=9 | 121.75 | 158.21 |
| W+WF | 17 | W=11%, WF=89%, *k*=9 | 119.12 | 148.21 |
| PV+WF | 23 | PV=13%, WF=87%, *k*=9 | 115.64 | 149.10 |
| PV+W+WF | 37 | PV=7.36%,W=24.64%, WF=68%, *k*=8 | 111.78 | 143.99 |

(W+WF).

The addition of WF features to any of the single and combined sets improves the accuracy (e.g. W+WF is better than W, PV+WF is better than PV and PV+W+WF is better than PV+W). On the other hand, the combination of PV and weather data is the least accurate dataset, followed by PV and PV+ WF. Just adding PV or W does not improve the accuracy, except when all three sources are combined (e.g.adding PV: W+PV is not better than W and WF+PV is not better than WF; adding W: PV+W is not better than PV and WF+W is not better than WF). Hence, the most useful data source for predicting the PV power output for the next day with *k*NN is WF, the weather forecast for the next day. However, the best performed combination uses the information from all three data sets (PV+W+WF), achieving MAE = 117.21 kW and RMSE = 158.10 kW.

**B. Performance of DW*k*NN**

Table 3.3 presents the accuracy results of DW*k*NN and also shows the weighting for each component learned by the algorithm when different sources of information were used. Fig. 3.8 shows the MAE values in sorted order for visual comparison. Note that there are no results in Table 3.3 for single PV, W, WF as DW*k*NN combines and weights different data sources, and hence it is not applicable to single sources.

The best accuracy is achieved by the full feature set using all data sources (PV + W + WF), followed by the other two feature sets that include the weather forecast, PV+WF and W+WF, and finally by PV+W. The weighting of the weather forecast features is the highest for all cases involving WF (from 68% for PV+W+WF to 89% for W+WF). The second highest weighting is for the weather features and the PV features have the lowest weighting in all cases involving PV. These results are consistent with

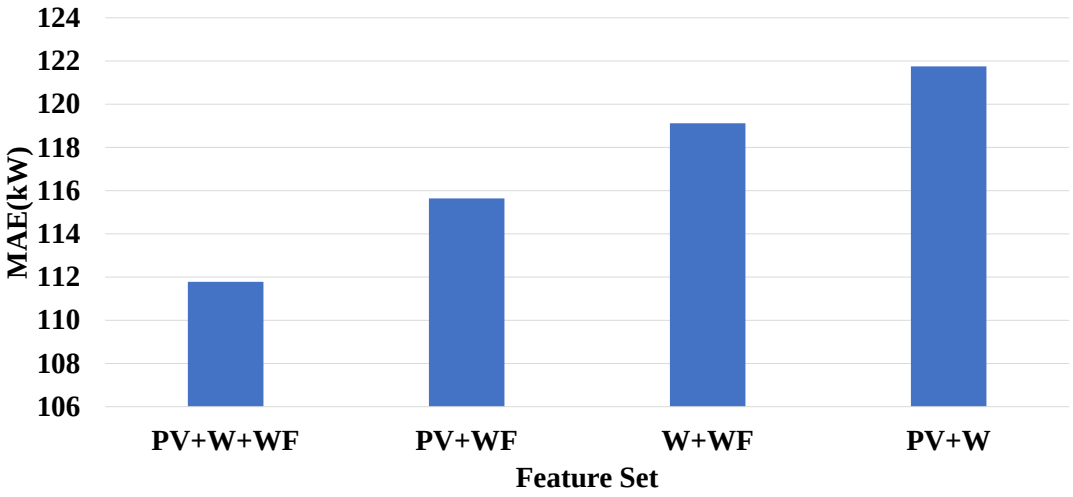


Figure 3.8: DW*k*NN performance (MAE) using different feature sets

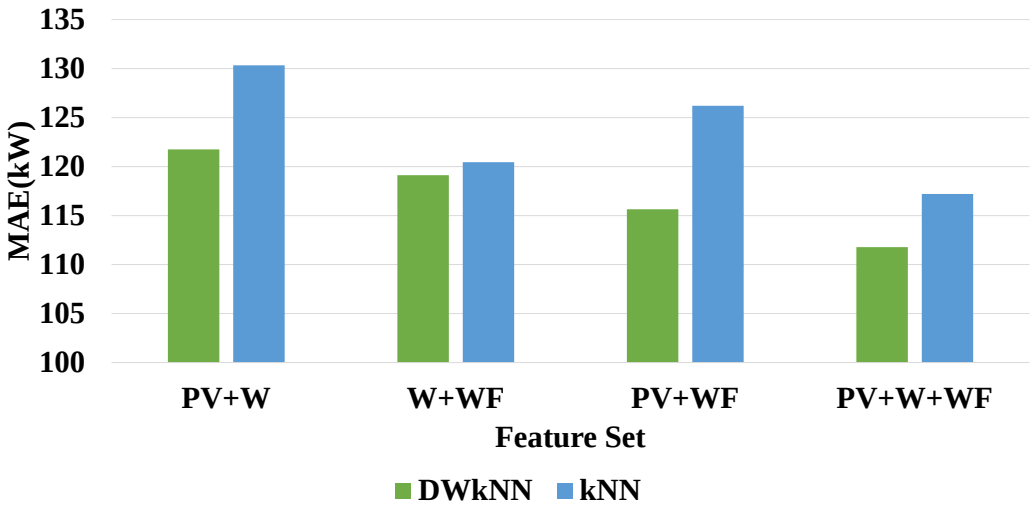


Figure 3.9: Comparison between DW*k*NN and *k*NN (MAE)

the *k*NN results and show again that the most important data source is the weather forecast for the next day. The results also show that when DW*k*NN is applied, the historical weather features are more important than the historical PV data - the weighting of W is always higher than the weighting of PV.

**C. Comparison between DW*k*NN and *k*NN**

We also investigated whether DW*k*NN improves the performance of Fig. 3.9 compares the two algorithms in terms of MAE. We can see that DW*k*NN improves the accuracy

Table 3.4: Statistical Significance Comparison Between *k*NN and DW*k*NN for MAE (Wilcoxon Rank Sum Test): -**-Stat. Sign. at $p \leqslant 0.001$, -*-Stat. Sign. at $p \leqslant 0.05$, -x-No Stat. Sign. Difference

| **Feature set** | $p$ **value** | **Statistical difference** |
|---|---|---|
| PV+W | 0.0023 | * |
| W+WF | 0.1425 | x |
| PV+WF | 0.0051 | * |
| PV+W+WF | 0.000072 | ** |

in all cases. The biggest improvement is achieved when historical PV data and weather forecasts are used (PV+WF), an improvement of 9.13% in terms of MAE and the smallest improvement is for W+WF. Both DW*k*NN and *k*NN perform the best when all three data sources are used and worst when only historical data (PV+W) is used. The second best data source is different - PV+WF for DW*k*NN and W+WF for *k*NN.

Table 3.4 shows the results of Wilcoxon rank sum statistical test, comparing DW*k*NN and the standard *k*NN. It can be seen that the differences in accuracy are statistically significant for all combinations of data sources except for PV+W.

**D. Comparison with Other Methods**

Table 3.5 shows the results of the methods used for comparison. The best performing model is SVR using all features (PV+W+WF), followed by NN with all features and SVR using the historical PV and weather features (PV+W). The addition of the weather forecast WF information improves the accuracy of both SVR and NN. The clustering based SVR is less accurate than the two non-clustering based SVR methods and performs similarly to the baseline. All methods outperform the baseline (persistent model) except ARIMA and ES.

Fig. 3.10 compares the accuracy (MAE) of the best DW*k*NN and *k*NN models with all methods from Table 3.5. We can see that the proposed DW*k*NN is the most accurate method, followed by *k*NN. The two nearest neighbor methods outperform the more sophisticated NN and SVR methods, the statistical methods and also the baseline.

**E. Comparison between WF and WF2**

The previous results (Tables 3.2, 3.3 and 3.5) showed that the addition of weather forecast information improves the results for all methods. The weather features selected for

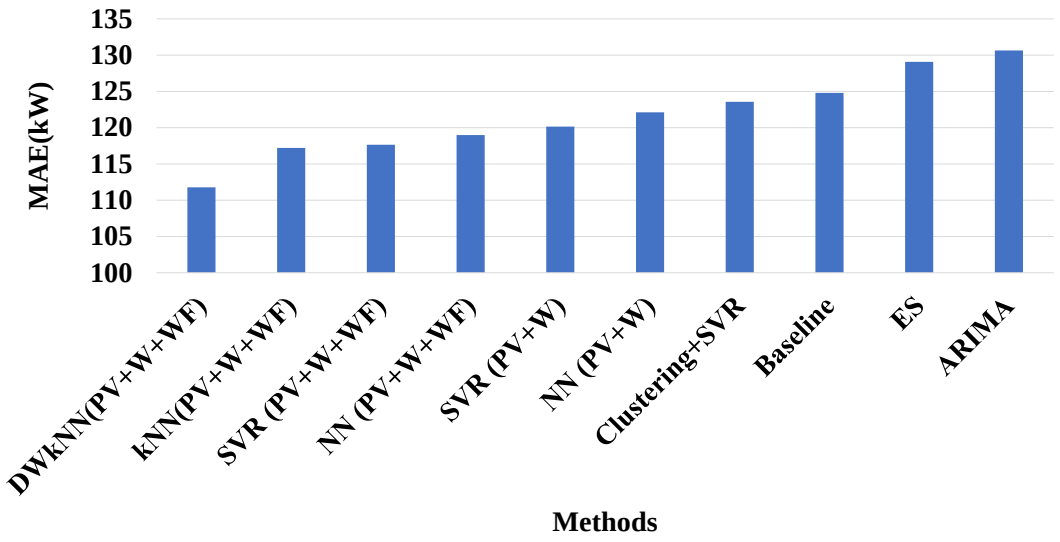


Figure 3.10: Comparison of all prediction methods (MAE)

Table 3.5: Accuracy of the Methods Used for Comparison

| Data source | MAE (kW) | RMSE (kW) |
|---|---|---|
| SVR (PV+W+WF) | 117.65 | 145.20 |
| NN (PV+W+WF) | 118.98 | 142.87 |
| SVR (PV+W) | 120.15 | 147.49 |
| NN (PV+W) | 122.12 | 146.85 |
| Clustering+SVR | 123.57 | 147.62 |
| ARIMA | 130.65 | 157.73 |
| ES | 129.08 | 158.26 |
| Baseline | 124.80 | 184.29 |

WF in this study are the daily minimum temperature, daily maximum temperature and daily rainfall (See Table 3.1). Apart from these three most commonly used features in weather forecasts, we also investigate if further improvements can be achieved by extending the WF feature set with one more feature - the daily average solar irradiance.

Although weather forecasts for solar irradiance are frequently used in research papers, they are not always available from meteorological bureaus. Our goal is to assess the usefulness of a solar irradiance forecast, for the cases when this information is easily available. In particular we consider a single value - the daily average solar irradiance; such single forecast value can be expected to be more easily available than

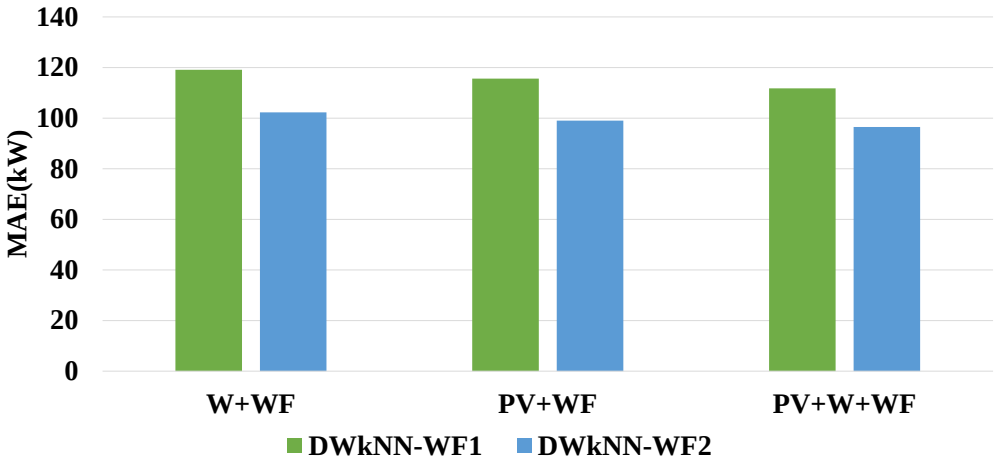


Figure 3.11: Comparision between DW$k$NN-WF and DW$k$NN-WF2

Table 3.6: Feature Set WF2

| **Data source** | **Feature set** | **Num. features** | **Description** |
|---|---|---|---|
| Weather forecast for the next day $d$+1 | WF2 | 4 | Daily: min temperature, max temperature, rainfall and average solar irradiance |

hourly forecasts. Table 3.6 describes the new feature set WF2, which is WF with the addition of the average daily Solar Irradiance (SI) forecast: WF2=WF+SI.

Table 3.7 shows the performance of DW$k$NN with WF2, together with the parameters learned by the algorithm (data source weights and $k$). We can see that SI has the highest weighting in all cases (around 70%) which shows the importance of the SI forecast feature for the PV power prediction. All three feature sets perform well but the best results are achieved when all three data sources are used PV+W+WF2.

To assess the impact of the SI forecast feature, Fig. 3.11 compares the performance of WF and WF2. We can see that there is a considerable improvement (15.82% - 16.78%) in all cases. Hence, the addition of the SI forecast to the temperature and rain forecasts for the next day improves the performance and the SI forecast should be used when it is available.

**F. Computational Time**

Table 3.7: Performance of DW$k$NN with WF2

| **Feature set** | **Num. features** | **Best weights and $k$** | **MAE (kW)** | **RMSE (kW)** |
|---|---|---|---|---|
| W+WF2 | 18 | W=9%, WF=20.03%, SI=70.97%, $k$=8 | 102.30 | 129.87 |
| PV+WF2 | 24 | PV=11%, WF=30.55%, SI= 58.45%, $k$=8 | 99.02 | 129.01 |
| PV+W+WF2 | 38 | PV=2.53%, W=8.47%, WF=19.59%, SI=69.41%, $k$=9 | 96.51 | 126.47 |

Apart from the prediction accuracy, another aspect of the model performance, the computational time is also evaluated in this study. As described in the previous sections, applying DW$k$NN includes two stages. The first stage involves parameter optimization and selection, taking about 10 minutes. The second stage involves finding the nearest neighbors and generating the prediction, taking only several seconds. As the parameter selection process is done off-line, the computational time for DW$k$NN is appropriate for both off-line and on-line PV power output prediction tasks.

#### 3.1.2.4 Conclusion

In this case study, we evaluated the performance of the proposed DW$k$NN algorithm on Australian data for two years. DW$k$NN considers the importance of the different data sources and learns the best weights for them from previous data. The goal was to directly and simultaneously predict the PV power output for the next day at 30-min intervals, using historical PV power data, historical weather data and weather forecasts for the next day.

The results show that DW$k$NN outperforms the standard $k$NN algorithm for all data source combinations considered. DW$k$NN is most accurate when using together the three data sources, and in this case in addition to $k$NN it also outperforms a number of other methods used for comparison - NN, SVR, clustering based SVR, ARIMA, ES and a persistence model used as a baseline.

The assessment of the importance of the three data sources shows that the weather forecast for the next day, which includes only three easily available features (the daily minimum and maximum temperatures and the daily rainfall), is the most useful data

source, followed by the previous weather data and the previous PV data. The addition of the weather forecast information to the other data sources improves the accuracy in all cases.

The results also show that further improvements can be achieved by extending the weather forecast feature set by adding the average solar irradiance forecast for the next day. This single additional feature improves the accuracy of DW*k*NN with approximately 16% and should be used when it is available. The overall best performing algorithm was DW*k*NN using the PV+W+WF2 feature set and achieving MAE = 96.51 kW and RMSE = 126.47 kW.

Overall, the results show that extending the standard *k*NN to consider the importance of the different data sources and learn the best weights for them is beneficial.

## 3.2 Extended Pattern Sequence-based Forecasting

Pattern Sequence-based Forecasting (PSF) [50] is another instance-based algorithm that has been successfully applied for time series forecasting. In particular, it has been applied for predicting electricity demand and electricity prices. In this chapter we describe our extension of the PSF algorithm and its application for solar power forecasting.

The standard PSF algorithm combines clustering with sequence matching. Fig. 3.12 shows how PSF works on a time series prediction task where the given data represents a sequence of days, each day represented as a feature vector. The prediction process consists of three main steps: (1) clustering and labelling; (2) searching for matching pattern sequences; (3) making the final prediction.

1. **Clustering and labelling**: PSF first uses a clustering technique to group the historical data (the vectors for each day) and labels them with the cluster number, e.g. $C1, C2$, etc.

2. **Searching for matched pattern sequences**: PSF then extracts the sequence of consecutive days with length $w$, from the day to be predicted backwards (excluding this day). It then searches the previous data to find the same sequence and forms a set of matched sequences.

3. **Making final prediction**: Once all matched sequences are collected, immediate

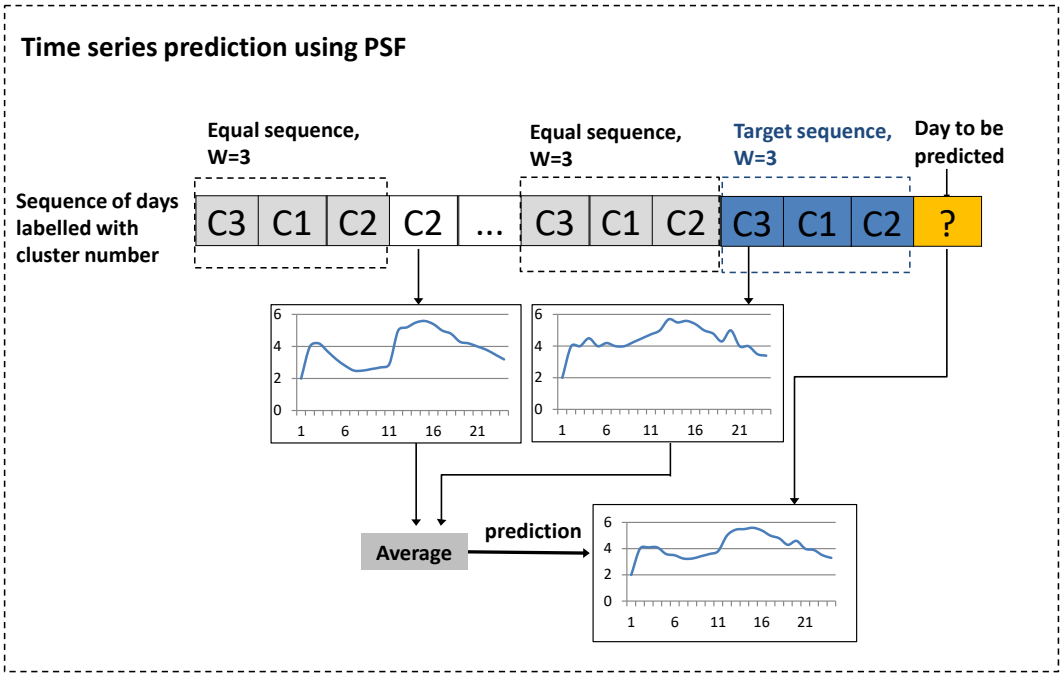


Figure 3.12: PSF for time series prediction

post-sequence days for each of them are extracted, and their vectors are averaged to form the final prediction for the new day.

Hence, PSF utilizes cluster labels rather than real values to find the matched sequences; the values are not used until the last step to make the final prediction. Also, similarly to DW$k$NN, PSF generates the prediction for all time points of the new day simultaneously, which is an advantage.

Even though PSF has been shown to be successful for predicting energy related time series, it has not been applied to solar power prediction. The solar data shows different patterns than electricity loads and electricity prices so it is worth to investigate the performance of PSF on solar data. Importantly, solar power generation depends on other data sources such as weather data, but the standard PSF is only applicable to a single time series. Therefore, there is a need to extend the traditional PSF algorithm to take advantage of information from more than one data source when it is available.

Section 3.2.1 describes how the standard PSF algorithm can be used for solar power prediction tasks and proposes two extensions which include other data sources - historical weather data and weather forecast data. Section 3.2.2 describes a case study to evaluate the performance of the proposed extensions and compares the stndard PSF

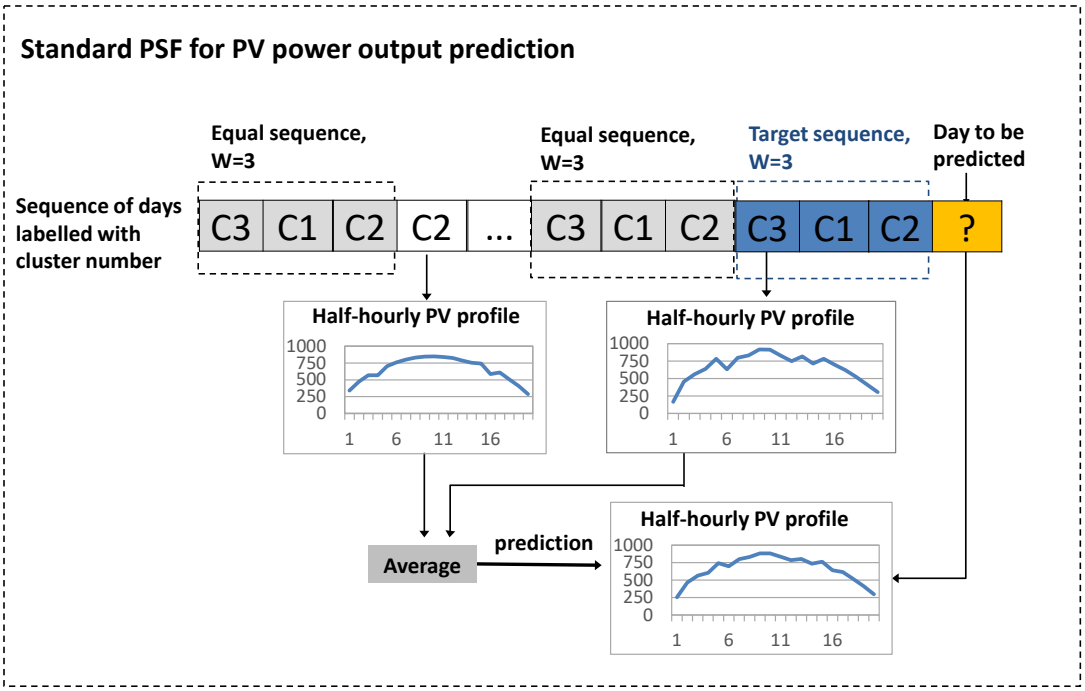


Figure 3.13: PSF for daily PV power output prediction

and the proposed extensions with other popular algorithms for PV power prediction.

### 3.2.1 Methodology

As before, we consider again the task of predicting the half-hourly PV power power output for the next day. As mentioned before, PSF is able to simultaneously predict all half-hourly PV values for the new day.

#### 3.2.1.1 PSF for PV Power Prediction

PSF firstly groups PV vectors for each day into several clusters using a clustering method such as K-means and labels the day with the cluster ID. More specifically, let $P^i$ be the 20-dimensional vector of the half-hourly PV power output for day $i$. PSF firstly clusters all vectors $P^i$ from the training data into $k_1$ clusters and labels them with the cluster number, e.g. $C1, C2$, etc. as shown in Fig. 3.13.

Then, when a new day *d+1* to predict comes, PSF extracts a sequence of consecutive days with length $w$, starting from the previous day $d$ and going backwards, and matches the cluster labels of this sequence against the previous days to find a set of

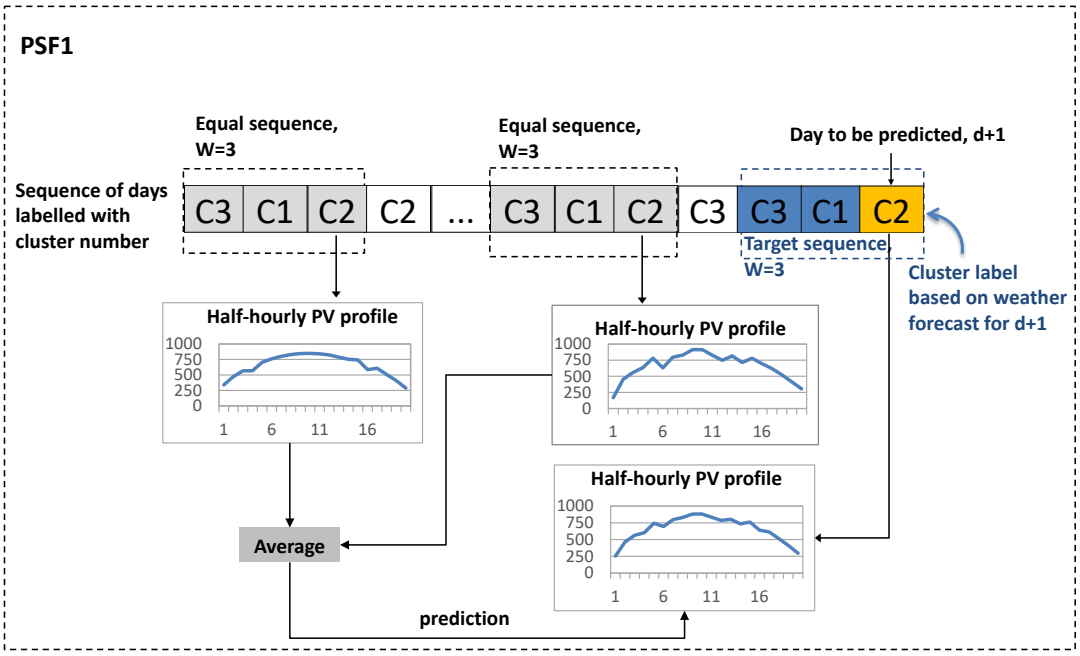


Figure 3.14: The proposed extension PSF1

equal sequences $ES_d$. It then follows a nearest neighbor approach - finds the post-sequence day for each sequence in $ES_d$ and averages the PV vectors for these days, to produce the final half-hourly PV predictions for day *d+1*. One thing to notice is that when searching for the equal sequences, only the cluster label sequences are used. The real PV profile of each day is not used until all the matched sequences are found and this helps to reduce the computational time.

#### 3.2.1.2 Extended Pattern Sequence-based Forecasting

This section introduces two extensions of the standard PSF algorithm, PSF1 and PSF2, which extend the standard algorithm by considering other data sources. PSF1 utilises the weather forecast, in addition to the historical PV data; Section 3.1 has shown the importance of the weather forecast. PSF2 uses a two-tier approach and utilises all three data sources: historical PV data, historical weather data and weather forecasts.

**A. PSF1**

The first extension of the standard PSF is PSF1, shown in Fig. 3.14. In contrast to the standard PSF, where the clustering and sequence matching are done using the PV data

only, in PSF1 this is done using the historical weather data for the previous days and the weather forecast WF for the new day. As mentioned before, the features available in the weather forecasts are typically fewer than those included in the historical weather data. To differentiate these two sets of weather features, in this study, W1 is referred as a full set of weather features typically available for historical weather data. W2 refers to a subset of W1 and W2 contains the features which are usually contained in the weather forecasts, such as the daily minimum and maximum temperature and daily rainfall.

The first step of PSF1 is also clustering. The days in the training data set are clustered using the weather data, W2 data into $k2$ clusters and then labelled using the cluster ID. To make a prediction for a new day *d+1*, PSF1 firstly uses the weather forecast for *d+1* (a vector containing the same features as W2, which facilitates the comparison) to find the cluster label for *d+1* by comparing it with the cluster centroids of the existing clusters, and assigning it to the cluster of the closest centroid. It then extracts a sequence of consecutive days with length *w*, from day *d+1* backwards (including *d+1*), and matches the cluster labels of this sequence against the previous days to find a set of equal sequences. It then obtains the PV power vector for the last day of each equal sequence and averages these vectors, to produce the prediction for *d+1*.

The main differences between PSF1 and the standard PSF are as follows:

1. **Data sources**

   PSF uses only historical PV data, for both the clustering of days and generating the prediction. It is limited to using a single data source.

   PSF1 uses data from three data sources: PV, weather (W2) and weather forecast (WF). W2 is used for the clustering of the previous days, WF is used to determine the cluster label of the new day and it is also included in the pattern sequence that is being matched. The PV data is used after the matched sequences have been collected, to generate the PV power prediction for the new day.

2. **Target sequences**

   Another difference between PSF and PSF1 is in the way the target sequences are formed.

   The target sequence in PSF (see Fig. 3.13) is a cluster label sequence of length *w*, starting from the day before the new day. More specifically, if the day to

predict is day *d+1*, the target sequence is formed using the cluster labels of days [*d-w+1*,..., *d*]. The label for day *d+1* is not available and cannot be used to form the target sequence. This means that no information about the new day is considered, which may weaken the performance of PSF.

On the other hand, in PSF1, the cluster label of the new day is determined and included in the target sequence. It is calculated by comparing the WF vector of the new day with the centroids of the W2 clusters, and is then used to form the target sequence (see Fig. 3.14). This means that information about the day that is being predicted contributes to the prediction. Specifically, to predict the new day *d+1*, the target sequence will use the cluster labels of days [*d-w+2*,..., *d+1*].

3. **Computation time**

   The computation time of the PSF algorithms mainly depends on two steps: clustering and searching for matching sequences. The later is similar as the sequences contain cluster labels and the search method is the same. However, since PSF1 uses the W2 weather data for the clustering, which usually contains much smaller number of features than the daily PV power output vectors (e.g. a vector of 20 half-hourly PV power output data or 600 PV power output data at 1-min intervals compared to 4 features in W2), PSF1 takes much less time for clustering than PSF; this means that PSF1 is faster to train than PSF.

**B. PSF2**

The previous section has discussed the advantages of PSF1 over PSF. However, PSF1 only utilizes a subset (W2) of all weather features (W1) as it needs to determine the cluster label of the new day based on the weather forecast features which are a subset of W1. To fully utilize the available weather features (W1), a second extension PSF2 is proposed and shown in Fig. 3.15. It involves a two-tier clustering approach, where the days are clustered in two different ways: based on the W1 and W2 weather data. PSF2 involves the following steps:

- Firstly, the training data is clustered using the W1 data (the full weather data, containing more features than W2) into $k1$ clusters. The days are labelled with the cluster ID, e.g. $C1$, $C2$, etc. as shown in Fig. 3.15. A sequence of consecutive days with length *w*, from day *d* backwards (including *d*), is formed as the first tier target sequence.

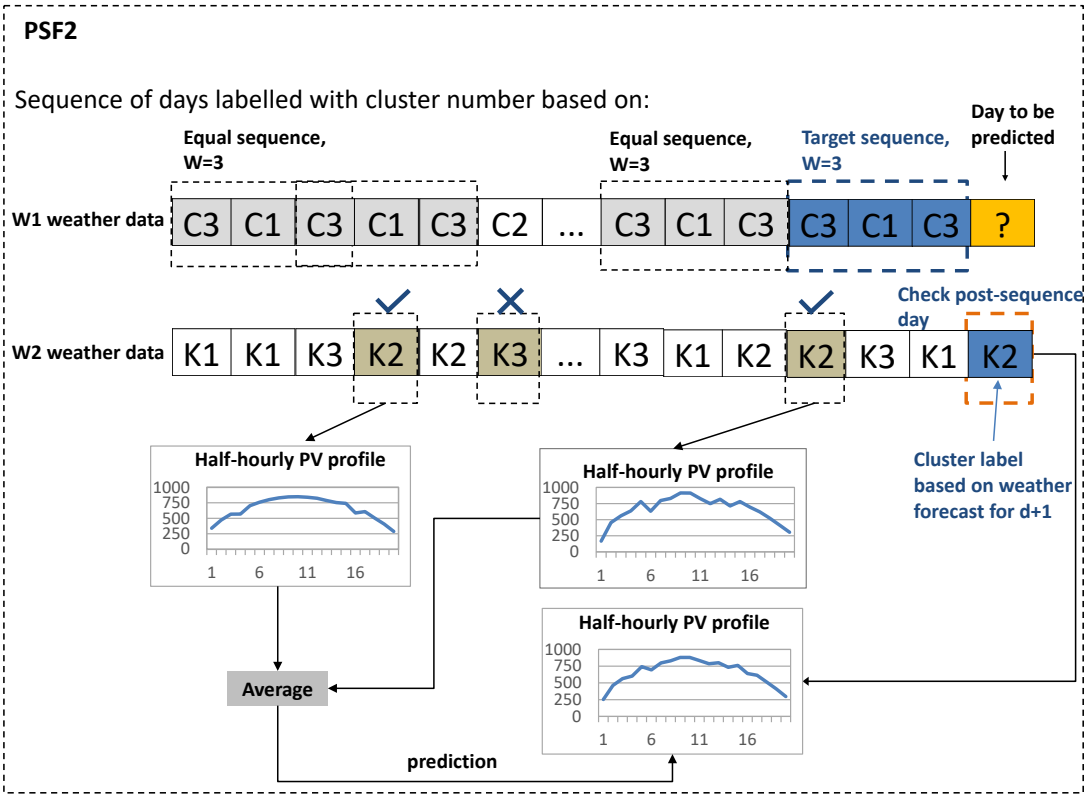


Figure 3.15: The proposed extension PSF2

- Secondly, the training data is clustered using the shorter weather vector W2 into $k2$ clusters, denoted with K1, K2, etc. The cluster label Kx for the new day *d+1* is found by obtaining the weather forecast for *d+1* and comparing it with the cluster centroids, forming the second tier of cluster labels. This process similar with the labeling process of PSF1.

- Thirdly, the cluster label of the post-sequence days for the equal sequences from the first clustering is checked, and if it is not Kx, the equal sequence is not included in obtaining the prediction for *d+1*. For the example from Fig. 3.15, the second left most equal sequence is not included as the cluster of the post-sequence day is K3 which is different from K2, the cluster for the new day.

- Finally, PSF2 averages the PV power vectors of the selected post-sequence days, to produce the prediction for day *d+1*.

In summary, PSF2 is an extension of PSF1. It makes a better use of the available

weather data than PSF1 by using the full weather vector W1 for the initial clustering (available for the previous days) and the shorter weather vector W2 to match the weather forecast for the new day, and refine the equal sequence selection from the previous step.

**C. Parameter selection**

For both PSF1 and PSF2, the number of clusters ($k1$ and $k2$) and the sequence length *w* are parameters that need to be determined.

A possible way to determine them is to follow the procedure proposed for the standard PSF algorithm [50]. It uses 12-fold cross validation on the training data (one year of data, where each fold is one month) to evaluate the performance of several clustering quality measures and sequence lengths, and to select the best parameters.

Specifically, for the number of clusters it calculates the the Silhouette, Calinski-Harabasz and Davies-Bouldin clustering quality measures and determines the best number of clusters for each of them. Then, it takes a majority vote over the three results to determine the final number of clusters; if necessary it uses the second best number of clusters.

To determine the sequence length *w*, different values of *w* are evaluated (e.g. from 1 to 10) and the one that minimizes the average error over the 12 folds is selected.

To select the sequence length *w*, different values are evaluated (from 1 to 10) using 12-fold cross validation, where one fold corresponds to one month. The best *w* is the one that minimizes the average error for the 12 folds.

### 3.2.2 Case Study

In the previous section, two extensions of PSF have been proposed. In this section, a case study is conducted to evaluate the performance of the proposed extensions. Similar to the case study in Section 3.2.3, the performance is evaluated using Australian data for two whole years (2015-2016).

The performance of the proposed extensions are also compared with other methods. As the models based on Neural Networks (NN) are the most popular for solar power prediction tasks and have shown good performance in the previous study, this section mainly chooses NN as the model for comparison with the proposed PSF extensions. The NNs used for comparison is extended by including more data sources as

inputs, which is consistent the proposed PSF extensions. The details of the NNs will be disused in Section 3.2.2.2. A persistent model is also used in this section as the baseline, which is exactly the same with the baseline model in Section 3.1.2.1.

Another target of this study is to evaluate the performance of models under different forecast noise levels. The results of the DW*k*NN case study has shown the importance of weather forecast for solar power prediction tasks. However, reliable weather forecast is not always available for the PV plants and for most cases the accuracy of weather forecasts is uncertain. Therefore, there is a need to evaluate the robustness of predicting models given the weather forecasts under different accuracy levels. This will be achieved by manually adding different levels of noise to the whether data.

#### 3.2.2.1 Experimental Setup

We consider the task of directly and simultaneously predicting the PV power output for the next day at 30-min intervals, given historical weather data, historical PV data and weather forecasts.

More specifically, given: (i) a time series of historical PV power outputs up to the day *d*: $[p^1, p^2, p^3, ..., p^d]$, where $p^i = [p^i_1, p^i_2, p^i_3, ..., p^i_{20}]$ is a vector of 20 half- hourly power outputs for the day *i*, (ii) a time series of weather vectors for the same days: $[W^1, W^2, W^3, ..., W^d]$, and (iii) the weather forecast for day *d+1* : $WF^{d+1}$, our goal is to forecast $PV^{d+1}$, the half-hourly power output for the next day *d+1*.

**A. Data Sources**

We use PV power and weather data for two years - from 1 January 2015 to 31 December 2016. The data sources and feature sets are described in Table. 3.8.

The PV data and W1 weather features are the same as in the previous case study for DW*k*NN algorithm (see Table. 3.1). In short, the half-hourly PV data of the daylight period from 7 am to 5 pm was collected for two years, so in total there are 14,600 measurements ($(366 + 365) \times 20$) for the PV data. The corresponding weather data is also collected and each day is represented as a 14-feature vector W1. Hence, there are in total 10,234 measurements for the W1 data (= $(365 + 366) \times 14$).

W2 is a subset of W1, and it includes only four features, the ones that are typically used in weather forecasts: the daily minimum temperature, the daily maximum temperature, the daily rainfall and the daily average solar irradiance.

Table 3.8: Data sources and feature sets for PSF study

| **Data source** | **Feature set** | **Num. features** | **Description** |
|---|---|---|---|
| PV data for the current day *d* and previous days | PV | 20 | Half-hourly PV values between 7am and 5pm |
| Weather data for the current day *d* and previous days | W1 | 14 | (1-6) Daily: min temperature, max temperature, rainfall, sun hours, max wind gust and average solar irradiance; (7-14) At 9 am and 3 pm: temperature, relative humidity, cloudiness and wind speed. |
| Weather data for the current day *d* and previous days | W2 | 4 | Daily: min and max temperature, rainfall and solar irradiance. **W2 is a subset of W1.** |
| Weather forecast for the next day *d+1* | WF | 4 | Daily: min temperature, max temperature, rainfall and average solar irradiance |

The weather forecast feature set WF includes the same four features as the weather set W2. Since the weather forecasts were not available retrospectively for 2015 and 2016, we used the actual weather data with added noise. We considered three different noise levels: 10%, 20% and 30%, to better evaluate the robustness of the prediction models.

**B. Data Pre-processing**

The small percentage of missing values (0.82% for the weather data and 0.02% for the PV data) were replaced using the same nearest neighbor-based method as discussed in Section 3.2.3.2. Then both the PV data and weather data were normalized to [0,1].

**C. Training, Validation and Testing Sets**

To evaluate the PSF algorithms, the data is divided into two sets: year 2015 for training and validation, and year 2016 for testing. As mentioned in the parameter selection section, the best number of clusters and sequence length is selected using a 12-fold cross validation and for this study we use the data of the first year (2015) to determine the parameters. Then, the second year (2016) is used to test the performance of the PSF based methods.

For the NNs used for comparison, the data for the fisrt year (2015) is further divided into two non-overlapping subsets: training the first 70% and validation: the remaining 30%. The training data was used to build the model, the validation set - to select the number of hidden neurons and other parameters. For consistent comparison with the PSF models, the data for the second year (2016) is used to evaluate the performance.

#### 3.2.2.2 Methods for Comparison

We compare the PSF methods with three NN methods and a persistence model commonly used as a baseline.

More specifically, the NN prediction models are multi-layer NNs with one hidden layer, trained with the Levenberg-Marquardt version of the backpropagation algorithm. The three NN models correspond to the three PSF models in terms of data source used, e.g. NN and standard PSF use only PV data while NN2 and PSF2 use data from all three data sources. The inputs and outputs for each model and their number are shown in Table 3.9. For example, NN2 uses as inputs the PV power for the previous day, the weather data for the previous day and the weather forecast for the next day and predicts the PV power for the next day; it has 20+14+4=38 input neurons and 20 output neurons.

The number of hidden neurons *h* was selected experimentally by varying it from 5 to 30, with an increment of 5, evaluating the performance on the validation set and selecting the best NN.

A persistence model ($B_{per}$) is used as a baseline . It considers the half-hourly PV power output from the previous day *d* as the prediction for the next day *d+1*. This means that the prediction for $\widehat{PV^{d+1}} = [pv_1^{d+1}, pv_2^{d+1}, ..., pv_{20}^{d+1}]$ is given by $PV^d = [pv_1^d, pv_2^d, ..., pv_{20}^d]$.

Table 3.9: Input and output of the neural models used for comparison

| NN model | Input | Output |
|---|---|---|
| NN | $PV^d(20)$ | $PV^{d+1}(20)$ |
| NN1 | $PV^d(20)$, $WF^{d+1}(4)$ | $PV^{d+1}(20)$ |
| NN2 | $PV^d(20)$, $W^d(14)$, $WF^{d+1}(4)$ | $PV^{d+1}(20)$ |

#### 3.2.2.3 Results and Discussion

**A. Selected parameters for PSF and NN Models**

As discussed in the previous section, the parameters of the three PSF methods (number of clusters and window size) are determined by using 12-fold cross validation, where each fold corersponds to one month, on the data for the first year. This is consistent with the procedure used in the standard PSF method [50].

More specifically, to select the best number of clusters $k, k_1$ and $k_2$, the clustering algorithm (k-means) is run and the quality of the clustering results is evaluated by computing the Silhouette, Calinski-Harabasz and Davies-Bouldin indexes and determine the best number of clusters based on each index. A majority vote is then across the three indexes to determine the final best number of clusters.

The clustering evaluation results are summarized in Table 3.10. As can be seen, $k$ =2 is selected for the standard PSF (it was voted as the best number by the three indexes). Recall that PSF uses only the PV data. This result is consistent with the PV data profiles of Australian data shown in Fig. 3.16 where we notice two distinct PV power profiles: 1) a smooth curve with highest values at noon and lowest at the beginning and end of the day and 2) a a flatter graph, highly variable during the day, without extreme peaks or lows. The former usually refers to a sunny day, while the latter usually refers to rainy or cloudy days.

For PSF1, the best number of clusters $k_2$ was 2; for PSF2, the best number of clusters for the first tier clustering (using all weather features) was $k_1$ =2 and for the second tier (using the subset of weather features) was $k_2$ =2.

The best window size $w$ was also determined using 12-fold cross validation on the data for the first year. Different values of $w$ are evaluated (from 1 to 10); the best $w$ is the one that minimizes the average error on the 12 folds. Table 3.11 shows the selected values of $w$ - $w$ = 2 for PSF and PSF2 and $w$=4 for PSF1. As $w$ reflects the length of pattern sequences to search for in the training data, the results indicate that relatively short sequences are more important than longer.

For the NN models, the number of hidden neurons was also selected using a validation set. The best performance on the validation set was achieved for $h$ = 15, 20 and 30, for NN, NN1 and NN2, respectively, and these values were used for the NN architectures to obtain the results on the testing set.

Table 3.10: Clustering evaluation results

| **Index** | **PSF** |
|---|---|
| Davies-Bouldin | Best $k$ = 2, Second best $k$ =3 |
| Calinski-Harabasz | Best $k$ = 2, Second best $k$ =3 |
| Silhouette | Best $k$ = 2, Second best $k$ =3 |
| | **Selected** $k$ = 2 |
| **Index** | **PSF1** |
| Davies-Bouldin | Best $k_2$ = 2, Second best $k_2$ =4 |
| Calinski Harabasz | Best $k_2$ = 3, Second best $k_2$ =4 |
| Silhouette | Best $k_2$ = 2, Second best $k_2$ =3 |
| | **Selected** $k$ = 2 |
| **Index** | **PSF2** |
| Davies-Bouldin | Best $k_1$ = 2, Second best $k_1$ =4; Best $k_2$ = 2, Second best $k_2$ =4 |
| Calinski-Harabasz | Best $k_1$ = 2, Second best $k_1$ =4; Best $k_2$ = 3, Second best $k_2$ =4 |
| Silhouette | Best $k_1$ = 2, Second best $k_1$ =4; Best $k_2$ = 2, Second best $k_2$ =3 |
| | **Selected** $k_1$= 2, $k_2$= 2 |

Table 3.11: Best *w* for PSF, PSF1 and PSF2

| **Methods** | **Best *w*** |
|---|---|
| PSF | 2 |
| PSF1 | 4 |
| PSF2 | 2 |

**B. Performance of All Methods**

Table 3.12 shows the performance of all methods. Fig. 3.17 - Fig. 3.19 present graphically the MAE results in sorted order for visual comparison. The main results can be summarized as follows:

- Among the PSF methods, PSF2 is the most accurate for all three noise levels. PSF2 outperforms PSF in all cases. Thus, the extensions introduced in PSF2 the use of weather and weather forecast data and 2-tier clustering and sequence matching were beneficial. This shows that the limitation of standard PSF can be addressed by taking more data sources into consideration and using them to form a different tier of pattern sequences.

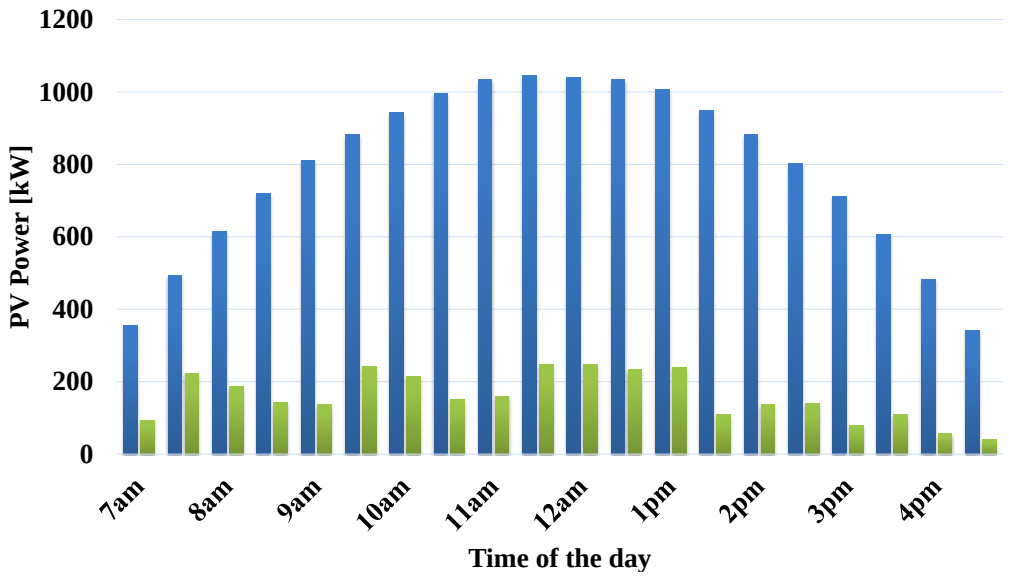


Figure 3.16: Two typical daily PV output profiles

- PSF2 outperforms PSF1 in all cases. It uses a more refined sequence matching and an additional data source the full weather vector W1. This result proves that with more features available in the weather features, the accuracy of prediction can be further improved.

- PSF1 performs similarly to PSF. PSF1 uses the weather forecast for sequence matching, while PSF uses the PV data. When the weather forecast is more accurate (10% noise), PSF1 performs slightly better than PSF, but as the accuracy of the weather forecast decreases (20% and 30% noise), PSF performs better. However, the differences are statistically significant only for 20% noise, hence overall the two methods perform similarly. One thing to notice is that PSF1 is the faster for training and making prediction as it uses a feature vector with smaller dimensionality (4 vs 20 features). Therefore, it can be seen that PSF1 has its own benefits for the PV output prediction tasks.

- In all cases NN2 is the most accurate model. It uses all data sources directly as inputs PV data, weather and weather forecast. The second best model PSF2 also uses all data sources but in a different way. The core part, sequence matching, is done using the weather and weather forecast data only, while the PV data is

Table 3.12: Accuracy of all methods

| Methods | 10% noise in WF | | 20% noise in WF | | 30% noise in WF | |
|---|---|---|---|---|---|---|
| | MAE (kW) | RMSE (kW) | MAE (kW) | RMSE (kW) | MAE (kW) | RMSE (kW) |
| **PSF** | 119.17 | 149.52 | 119.17 | 149.52 | 119.17 | 149.52 |
| **PSF1** | 118.12 | 151.52 | 120.05 | 156.61 | 123.04 | 154.10 |
| **PSF2** | 109.19 | 139.75 | 109.63 | 140.79 | 112.17 | 142.70 |
| **NN** | 116.64 | 154.16 | 116.64 | 154.16 | 116.64 | 154.16 |
| **NN1** | 111.16 | 149.33 | 117.47 | 158.86 | 126.14 | 173.85 |
| **NN2** | 94.75 | 133.65 | 95.76 | 134.62 | 96.89 | 135.69 |
| **Bper** | 124.80 | 184.29 | 124.80 | 184.29 | 124.80 | 184.29 |

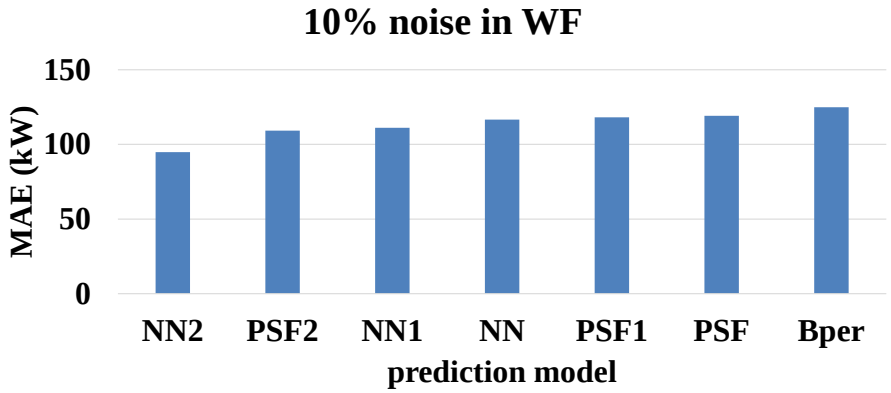


Figure 3.17: Performance of all methods under 10% noise level

used only in the last step. PSF2, however, is faster to train than NN2.

- All prediction models outperform the persistence baseline in all cases, except one (NN1 for 30% WF noise) but the difference in this case is not statistically significant.

**C. Influence of Noise**

As can be seen from Table 3.12 and Fig. 3.20, the performance of the PSF and NN methods using the weather forecasts (PSF1, PSF2, NN1 and NN2) is sensitive to the noise level. More specifically, the prediction error of these models increases in tandem with increasing the noise level. This result once again shows the importance of reliable weather forecasts for predicting the PV power.

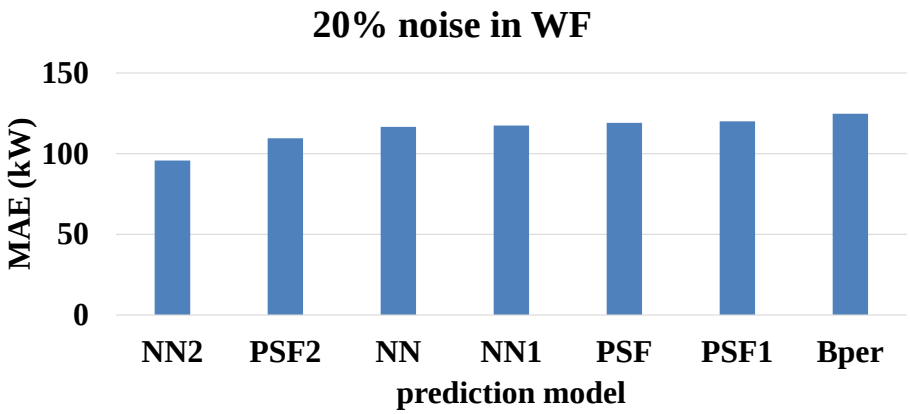


Figure 3.18: Performance of all methods under 20% noise level

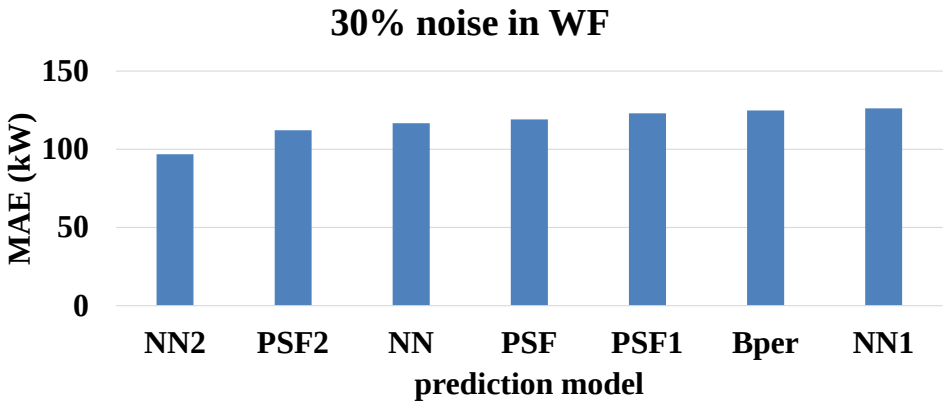


Figure 3.19: Performance of all methods under 30% noise level

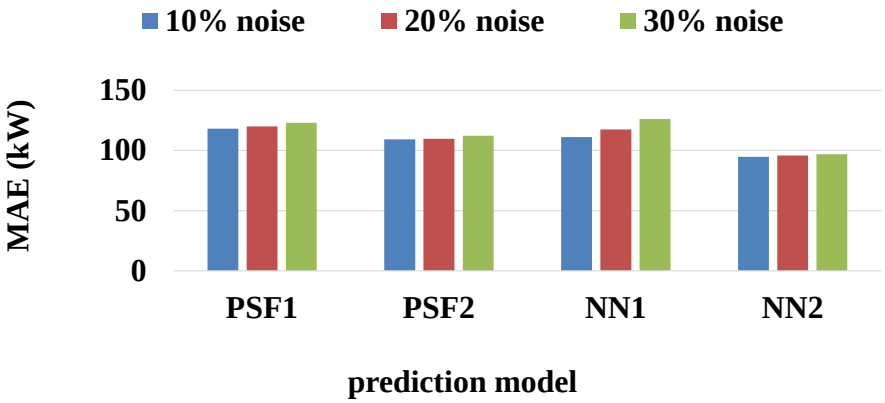


Figure 3.20: Comparison of models under different noise levels

For 10% noise in WF, PSF1 outperforms PSF; however, when the noise level increases to 20% and above, the accuracy of PSF1 decreases and becomes even lower than the accuracy of the standard PSF which doesn't use WF. Hence, when no accurate forecasts are available, it is better to use PSF instead of PSF1. Similarly, NN1 outperforms NN for noise level of 10%, but for noise levels of 20% and higher its accuracy decreases and becomes lower than NN's.

The accuracy of PSF2 and NN2 also decreases with the increase in noise, but only slightly. Compared to PSF1 and NN1, PSF2 and NN2 are less reliant on the WF features as they also use the W feature set, which may explain their better robustness to noise in WF.

In summary, PSF2 and NN2, which use the three features sets (PV, W and WF) are more robust to increasing the noise level in WF. On the other hand, PSF1 and NN1, which use only two feature sets (PV and WF), are more dependent on WF and the increased levels of noise in WF, and their accuracy drops below the accuracy of the standard PSF and NN for noise levels of 20% or higher.

#### 3.2.2.4 Conclusion

In this case study, we evaluated the performance of standard PSF method and the proposed extension, using Australian PV power and weather data for two years. We considered the task of predicting the next day PV power output at 30-min intervals, directly and simultaneously for all time points. The performance of the PSF methods was compared with three corresponding NN methods, and a baseline model.

The results show that the PSF extensions were beneficial - PSF2, which uses a 2-tier clustering and sequence matching, was more accurate than PSF in all cases. PSF1 performed similarly to PSF but was faster due to its smaller feature vector. Overall PSF2 was the second most accurate method, after NN2 - a neural network that uses directly the data from the three sources as inputs. However, PSF2 was faster to train than NN2. PSF2 and NN2 were also the most robust methods to higher levels of noise in the weather forecasts. Hence, we conclude that both PSF2 and NN2 are promising methods for solar power forecasting.

## 3.3 Summary

Chapter 3 explored the potential of instance-based methods for solar power prediction tasks and proposed two methods: DW*k*DD and extended PSF.

Section 3.1 introduced the DW*k*NN algorithm, which is an extension of *k*NN. It considers the importance of the different data sources and learns the best weights for them from previous data. The results of the case study using Australian data showed that DW*k*NN was more accurate than *k*NN and the other state-of-the-art methods and baselines.

Section 3.2 discussed the limitations of the PSF algorithm and proposed two extensions - PSF1 and PSF2, to deal with data from more than one data source. A case study using Australian data for two years from three sources (PV, weather and weather forecast) was conducted to evaluate the performance of the proposed extensions, and compare them with three corresponding NN models and a baseline. The results showed that the PSF extensions improved the accuracy compared to the standard PSF algorithm. Overall, PSF2 was the second most accurate method after NN2 but was faster to train.

Hence, based on the results, we can conclude that the proposed DW*k*NN and PSF extensions are promising methods for solar power forecasting, allowing to integrate data from multiple data sources.

# Chapter 4

# Clustering-based Methods

This chapter introduces two clustering-based methods for solar power prediction tasks: direct clustering-based and pair pattern-based.

Using machine learning methods such as NN and SVR and statistical methods such as ARIMA and ES is popular for building prediction models for solar power forecasting. However, most of these methods build a single prediction model for all weather types and their corresponding daily PV profiles. The main idea of the clustering-based methods is to group the days based on their weather characteristics and build a separate model for each cluster or pair of clusters. The motivation is that days with similar weather characteristics have similar PV power profiles; by considering this similarity we can build specilized prediction models that may be more accurate than a single general prediction model for all type of days.

For example, we can train separate models for different types of days: sunny, rainy cloudy, etc. Each model works as a independent prediction model which is able to capture the characteristics of the particular weather type. Fig. 4.1 shows the typical daily PV profiles of three different weather types from the same week. We can see that the power output is the highest for the sunny days and the lowest for the rainy days. The power output for the sunny days follows a smooth daily pattern, while it shows high variability for rainy and cloudy days depending on the weather conditions. This motivates us to group the days with similar weather conditions by using clustering technique and build prediction model for each group separately.

To make a prediction for a new day, we firstly obtain its weather forecast, determine the type of the day, and then select the corresponding trained prediction model for this type of day to make the PV power prediction. So the final prediction can be treated as

a collaborative work of prediction models for all different weather types. We call this method a direct clustering method and have published it in [95]. It will be discussed in Chapter 4.1.

However, the direct clustering-based method does not consider the relationship between two consecutive days. In particular, it only focuses on modelling the similarity of the PV data between the days in the same weather type cluster, and does not consider the relationship between two consecutive days in terms of weather and PV data. In order to address this limitation, we propose a new clustering-based approach that considers such relationships by extracting patterns for pairs of consecutive days. We have published this method in [42] and the details will be discussed in Chapter 4.2.

## 4.1 Direct Clustering-based Method

As described above, a single prediction model trained for all weather types fails to take advantage of the similarity of PV power profiles among the days with similar weather characteristics. Instead, we propose to train separate models for the days with different weather types and investigate if this can improve the prediction accuracy. In summary, the direct clustering-based methods include the following steps. Firstly, the days need to be partitioned into different groups based on their weather characteristics. To do this,

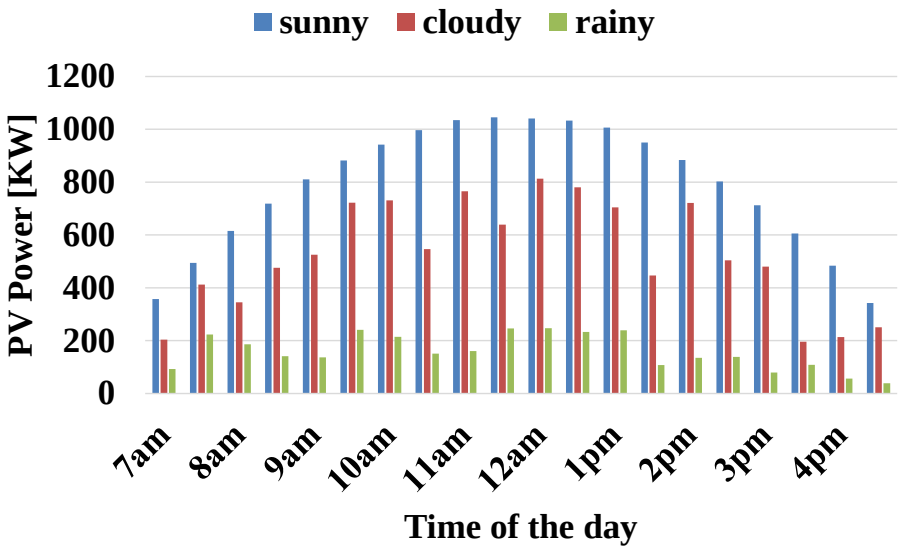


Figure 4.1: PV power outputs under different weather conditions

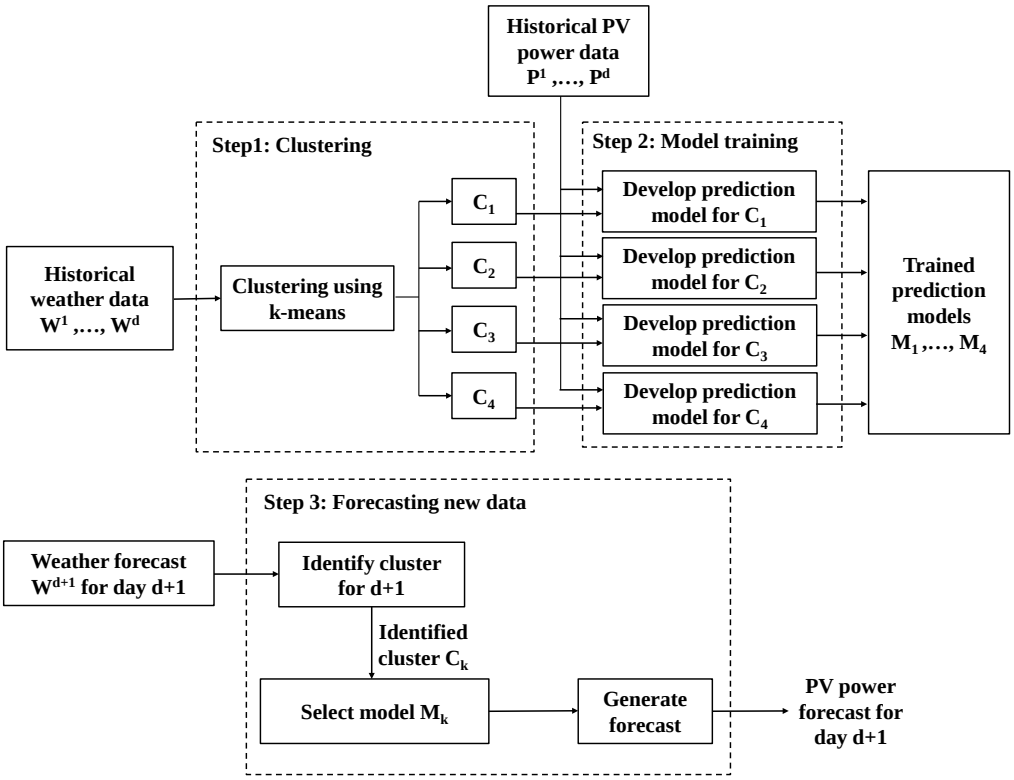


Figure 4.2: Main steps of the proposed clustering based forecasting approaches

after collecting the weather features of the days, clustering methods such as k-means can be used to group the days. Secondly, a separate machine learning model (e.g. NN and SVR) can be trained for this weather type. Finally, when a new day comes, the weather forecast report of this day is obtained and used to determine its weather type. Then, the trained model for this weather type is used to make the prediction for the new day.

## 4.1.1 Methodology

### 4.1.1.1 Main Steps

The proposed approach for solar power forecasting combines clustering and machine learning prediction algorithms. The key idea is to group the days into clusters based on their similarity in weather conditions and then develop a separate prediction model for each cluster. Fig. 4.2 shows the general structure of our approach. There are three main steps: clustering, training of prediction models, and generating predictions for

new days.

1. Clustering

   The first step is to partition the days into groups with similar weather using clustering methods. In this study, we consider two weather variables to represent the weather features for clustering, namely temperature and solar irradiance, which are easy to measure and available at PV stations. To find the most appropriate weather representation, we constructed five different weather vectors and evaluated their performances:

   - $W^d = SI^d_{avg}$ - mean solar irradiance for day *d*
   - $W^d = [SI^d_1, SI^d_2, ..., SI^d_n]$ - a vector of the half-hourly solar irradiance for day *d*
   - $W^d = T^d_{avg}$ - mean temperature for day *d*
   - $W^d = [T^d_{avg}, T^d_{max}, T^d_{min}]$ - mean, maximum, and minimum temperature for day *d*
   - $W^d = [T^d_1, T^d_2, ..., T^d_n]$ - a vector of the half-hourly temperature for day *d*

   To group the days into different clusters based on their similarity in weather conditions, we apply the k-means algorithm which was previously shown to be very effective for clustering of solar power data [6]. To select the number of clusters *k*, we calculate the Davies-Bouldin index, Calinski-Harabasz and Silhouette indexes and use majority voting to select the best *k* (*k*=4 for this study).

   Thus, five different groupings of the days from the training data into *k* clusters are obtained, each using a different weather representation.

2. Model Training

   The second step is to train a separate prediction model for each cluster. After the historical wether data is clustered and labelled, the days with the same cluster label are used to train a prediction model for this cluster. We used three prediction algorithms: *k*-NN, NNs, and SVR.

3. Predicting New Data

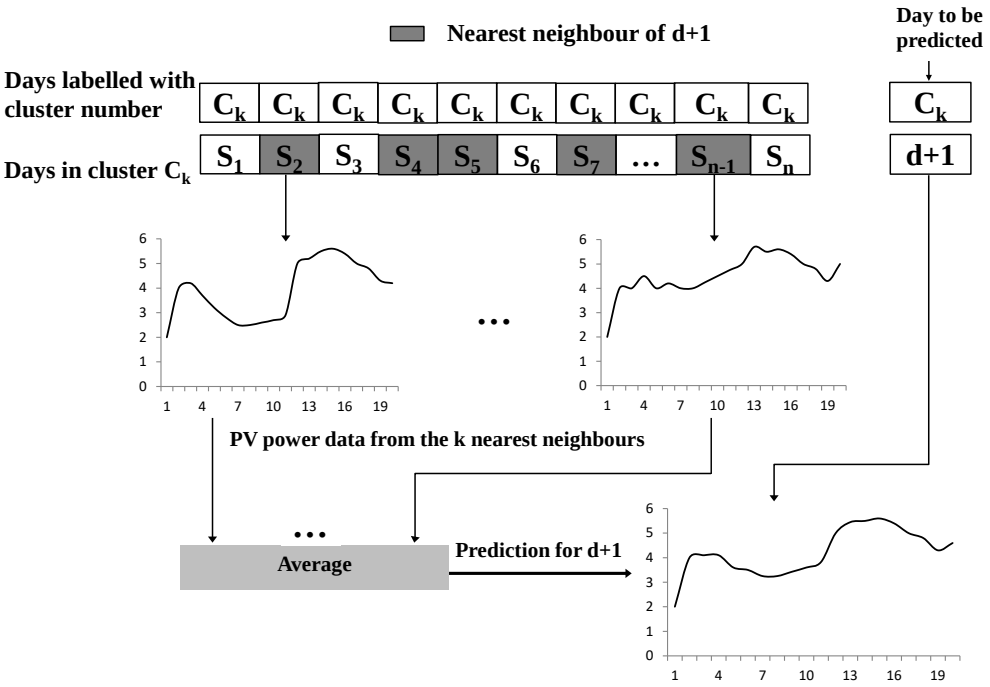


Figure 4.3: Forecasting using the clustering-based *k*-NN

To make a prediction for a new day, the weather forecast report of this day is firstly compared with the centroids of each cluster to determine the cluster label of the new day. Then, the trained prediction model for this cluster is selected to generate forecast for the day.

#### 4.1.1.2 Training of Prediction Models

In this study, we apply three prediction algorithms representing different machine learning paradigms: *k*-NN, NNs, and SVR.

**A. Using *k*-NN**

*k*-NN a is an instance based prediction algorithm. Given a new instance, it finds the *k* instances from the training set that are nearest to it and uses them to generate the prediction.

Fig. 4.3 shows the forecasting process for *k*-NN. To forecast the power output for the next day *d+1*, we firstly obtain the weather vector $W^{d+1}$ for this day from the

weather forecast report. *k*-NN then determines the cluster label $C_k$ for day *d+1* is determined by comparing the distance between $W^{d+1}$ and the centroids of all clusters based on a distance measure and selecting the cluster with smallest distance. *k*-NN then selects the *k* nearest neighbours for day *d+1* from cluster $C_k$ these are the days with the most similar weather vectors (smallest distance) to $W^{d+1}$. In our experiments, we used Euclidian distance and empirically set *k* = 5.

To generate the prediction (the half-hourly PV power output for day *d+1*), *k*-NN takes the average of the half-hourly PV output of the *k* neighbors: $p_t^{d+1} = \frac{1}{k}\sum_{j=1}^{k} p_t^j$, where $p_t^{d+1}$ is the power output at time *t* for day *d+1*, *k* is the number of nearest neighbour of the day *d+1*, $p_t^j$ is the power output of the neighbor *j*; *t*=1,...,20 and *j*=1,...,*k*.

**B. Using NNs**

To develop prediction models using NNs, we use multi-layer perceptron NNs with one hidden layer, 20 input and 20 output nodes. The 20 output nodes correspond to the half-hourly power outputs for the next day *d+1*, and produce the forecast simultaneously. The 20 input nodes correspond to the power outputs for day *S*, which is the closest in time day to *d+1* from the same cluster.

We train four different NNs, one for each cluster. The NN parameters (such as number of nodes in the hidden layer, transfer functions, learning rate and momentum) are determined experimentally. We train several NNs with different combinations of these parameters using the training dataset, and evaluate their performance on the validation dataset. The best performing NN for each cluster was then selected and used to predict the data from the testing dataset.

The forecasting steps of NNs are shown in Fig. 4.5. To predict the power outputs for day *d+1*, we firstly identify the cluster $C_k$ for *d+1* using the weather vector for *d+1*. The predictions for *d+1* are then computed by the trained NN for $C_k$, using as inputs the power outputs for the most recent day *S* from $C_k$.

**C. Using SVR**

We also implement SVR, which is another state-of-the-art machine learning algorithm, shown to obtain excellent performance for forecasting solar power data [16]. Similar to NNs, SVR is able to learn from training examples and form complex non-linear decision boundaries. However, unlike NNs, SVR finds a global rather than a local minimum during training, does not overfit as the decision boundary is defined by a

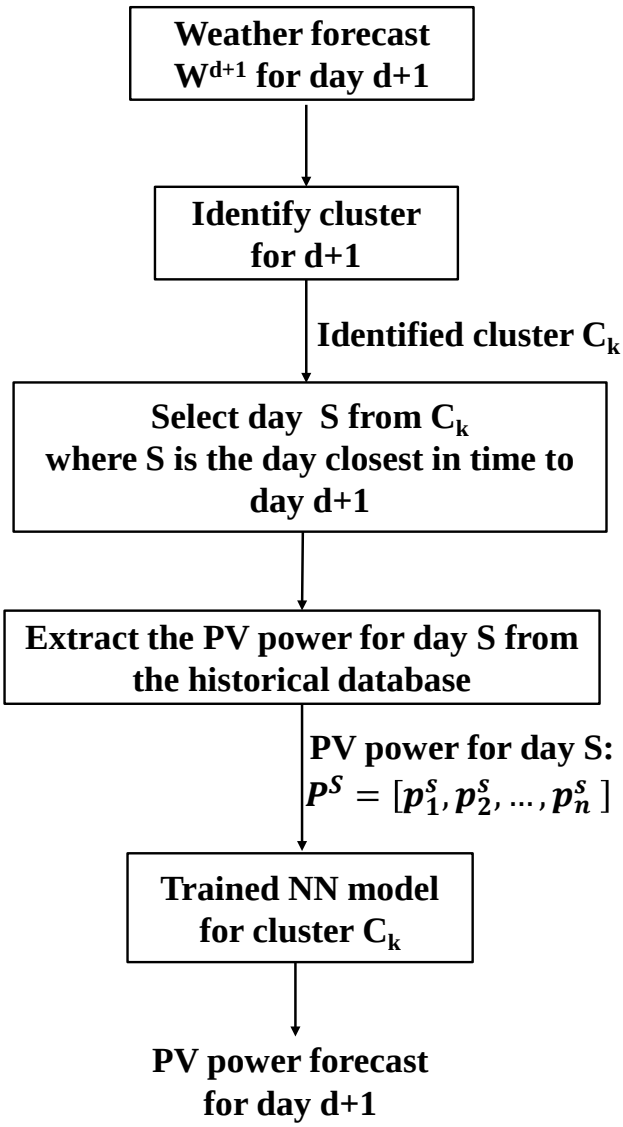


Figure 4.4: Forecasting using the clustering-based NN

small number of training examples and has a smaller number of parameters to tune.

To develop an SVR prediction model and generate the forecast for the next day, we followed the same methodology as for NNs, using the most recent day *S* from the same cluster as an input. However, in contrast to the single NN model for each cluster, that predicts all 20 power outputs for the next day, we created 20 SVR models for each cluster, each predicting one half-hourly value for the next day. Thus, for our case study we trained 80 SVR models in total (4 clusters $\times$ 20 models).

### 4.1.2 Case Study

To evaluate the performance of direct clustering-based methods, a case study is conducted using Australian PV and weather data for two years. The performance is compared with several other state-of-the-art non-clustering methods including *k*-NN, NN, SVR, seasonal ARIMA, ES and a persistence model as the baseline.

#### 4.1.2.1 Experimental Setup

As before, we consider the task of predicting the half-hourly PV power output for the next day, given historical PV power data, weather data and weather forecasts.

More specifically, given: (i) a time series of historical PV power outputs up to the day *d*: $[p^1, p^2, p^3, ..., p^d]$, where $p^i = [p^i_1, p^i_2, p^i_3, ..., p^i_{20}]$ is a vector of 20 half- hourly power outputs for the day *i*, (ii) a time series of weather vectors for the same days: $[W^1, W^2, W^3, ..., W^d]$, and (iii) the weather forecast for day *d+1* : $WF^{d+1}$, our goal is to forecast $PV^{d+1}$, the half-hourly power output for the next day *d+1*.

**A. Data Sources and Preprocessing**

We use the PV power data for two complete years - from 1 January 2013 to 31 December 2014. For each day, we only select data during the daylight period, from 7 am to 5 pm.

The original PV data is collected at 1-min intervals and contains $2 \times 365 \times 600 = 438,000$ measurements. There are 1,518 missing values, which accounts for approximately 0.35% of the total data. Each missing value is replaced by the average of the values from the previous 5 min. We aggregate data into 30-min intervals as our task is to make half-hourly predictions for the next day. Thus, we have 20 PV values for one day and 14,600 $(20 \times 365 \times 2)$ values in total for two years. The data is normalized to the interval [0-1].

With regards to the weather data, we only collected the temperature and solar irradiance of each single day to represent the day as these measurements are usually available at PV plants. The same pre-processing approach was applied as for the PV data. As the weather forecast data is not available retrospectively, we use the actual weather data with added 10% noise as the weather forecast.

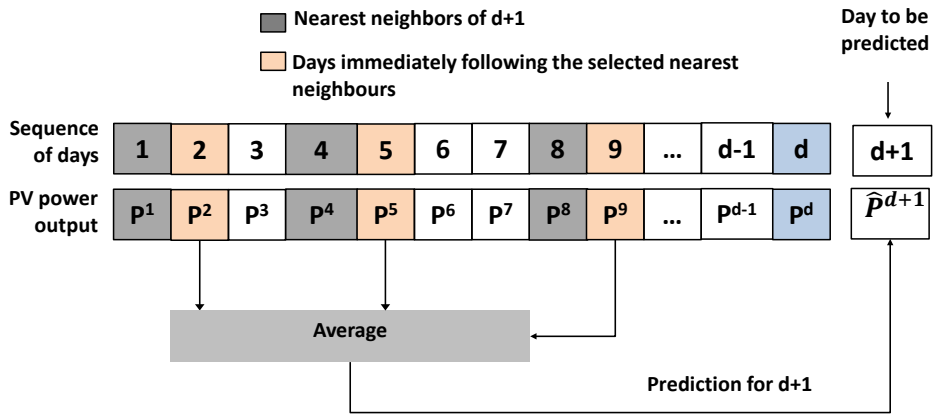


Figure 4.5: Forecasting using non-clustering based k-NN

**B. Training, Validation and Testing Sets**

We divide the solar power and corresponding weather data into three non-overlapping subsets: 50% (day 1 - day 365) for training, 25% (day 366 - day 550) for validation and 25% (day 551 - day 730) for testing. The training set is used to build the prediction models; the validation set is used for parameter selection and the testing set is used to evaluate the performance of the proposed clustering-based approach and the other methods used for comparison.

#### 4.1.2.2 Methods for Comparison

To compare the performance of our proposed clustering based forecasting approaches, we implemented three non-clustering based counterparts of the approaches described in the previous section and two other non-clustering methods based on ARIMA and ES. Unlike the proposed clustering based approaches that used both weather and PV power data, the non-clustering based approaches use only the PV power data (from one or more previous days) and do not require weather information. We also implemented two other persistence models as baselines.

**A. Methods Without Using Clustering**

1. **Non-clustering Based *k*-NN**

Fig. 4.5 shows the forecasting procedure of the non-clustering based *k*-NN. To predict the power output for the next day *d+1*, it firstly selects the *k* days from the historical dataset whose power profiles are most similar to day *d* (i.e. the day preceding to target day *d+1*), based on a distance measure between $P^d$ and all of $[P^1, P^2, ..., P^{d-1}]$. It then computes the forecasts for $P^{d+1}$ by taking the average of the subsequent day for each of the *k* selected similar days. Specifically, if $S = s_1, s_2, ..., s_k$ denotes the set of selected *k* days, then the prediction for $P^{d+1}$is given by: $P^{d+1} = \frac{1}{k}[p^{s_1+1}+p^{s_2+1}+, ..., p^{s_k+1}]$, where each $p^{i \in S}$ a 20-dimensional vector (i.e. $P^i = [p^i_1, p^i_2, ..., p^i_{20}]$) representing the 20 half-hourly power outputs for the day *i*.

2. **Non-clustering Based NN**

   The non-clustering based NN is similar to the clustering based NN, except that: 1) there is only one NN model for all days, since there is no clustering, and 2) as inputs it uses the PV power data from the most recent day d, not from the most recent day from the same cluster. To select the parameters of the non-clustering based NN, we followed the same procedure and experimental setting as for the clustering based NN.

3. **Non-clustering Based SVR**

   The non-clustering based SVR is similar to the clustering based SVR except that: 1) there is one set of 20 SVR models for all days, not a different set for each cluster of days, and 2) as inputs it uses the PV power data from the most recent day *d*, not from the most recent day from the same cluster. The parameters of the non-clustering SVR were selected using on same procedure as for clustering-based SVR.

4. **Seasonal ARIMA and Exponential Smoothing**

   Similar to the case studies in Chapter 3, we implemented a seasonal ARIMA and Exponential Smoothing (ES) for comparison. The parameters of ARIMA were selected by comparing the corrected Akaike Information Criterion ($AIC_c$) of the different ARIMA models. The parameters of the ES model were computed using an optimization procedure which minimizes the MSE for the training data.

**B. Persistence Models**

We also implemented two persistent models as baselines for comparison. The first baseline (**B1**) uses the clustering of days. It considers the power outputs from the closest day (in terms of time) with similar weather conditions as the predictions for the next day d+1. Specifically, the prediction for $P^{d+1} = [p_1^{d+1}, p_2^{d+1}, ..., p_{20}^{d+1}]$ given by $P^r = [p_1^r, p_2^r, ..., p_{20}^r]$ such that both *r* and *d+1* belong to same cluster $C_k$ and *r* is the closest to *d+1* among all the days in $C_k$.

The second baseline (**B2**) does not utilize the clustering. It considers the PV power outputs from the previous day *d* as the predictions for the next day *d+1*. This means that the prediction for $P^{d+1} = [p_1^{d+1}, p_2^{d+1}, ..., p_{20}^{d+1}]$ is given by $P^d = [p_1^d, p_2^d, ..., p_{20}^d]$

#### 4.1.2.3 Results and Discussion

**A. Performance of the Clustering Based Approaches**

Table 4.1 presents the accuracy results of the four clustering based approaches for the five different weather representations used for the clustering. The main results can be summarized as follows:

- The most accurate prediction model, in terms of all three accuracy measures, is the clustering based k-NN using the vector of half-hourly solar irradiance for the clustering. It achieved MAE = 59.81 KW and RMSE = 96.18 KW.

- The second most accurate prediction model is the clustering based *k*-NN using the vector of half-hourly temperature as an input for the clustering, which achieved MAE = 68.49 KW and RMSE = 109.36 KW. The other approaches achieved considerably lower accuracy: MAE = 82.55-140.18 KW and RMSE = 113.59-200.82 KW.

- There is no single best prediction algorithm  different algorithms perform best with different weather representations used for the clustering. For the clustering methods that use a single aggregated weather measurement (mean daily temperature for clustering method 1 and mean daily irradiance for clustering method 3, the best results are achieved using SVR as a prediction algorithm, followed by NN, while k-NN performs similarly or below the baseline.

- For the clustering methods that uses a 20-dimensional weather representation of all half-hourly values of the solar irradiance or temperature during the day

Table 4.1: Performance of the clustering based approaches

| **Clustering using:** | **MAE (KW)** | **RMSE (KW)** |
|---|---|---|
| **(1) Mean daily solar irradiance (1 feature)** | | |
| k-NN | 90.84 | 128.19 |
| NN | 93.94 | 130.2 |
| SVR | 82.55 | 113.59 |
| Persistent (B1) | 90.59 | 145.24 |
| **(2) Vector of half-hourly solar irradiance (20 features)** | | |
| k-NN | 59.81 | 96.18 |
| NN | 94.57 | 134.45 |
| SVR | 83.38 | 116.14 |
| Persistent (B1) | 90.43 | 144.52 |
| **(3) Mean daily temperature (1 feature)** | | |
| k-NN | 140.18 | 181.86 |
| NN | 133.28 | 177.73 |
| SVR | 116.7 | 155.17 |
| Persistent (B1) | 128.49 | 198.07 |
| **(4) Vector of half-hourly temperature (20 features)** | | |
| k-NN | 68.49 | 109.36 |
| NN | 137.62 | 183.72 |
| SVR | 119.51 | 158.34 |
| Persistent (B1) | 125.49 | 193.67 |
| **(5) Daily lowest, mean and highest temperature (3 features)** | | |
| k-NN | 120.70 | 162.02 |
| NN | 135.63 | 184.77 |
| SVR | 120.21 | 162.01 |
| Persistent (B1) | 130.94 | 200.82 |

(clustering methods 2 and 4), $k$-NN is the most accurate algorithm, considerably outperforming the others. For the case in between that uses three weather variables (clustering method 5), $k$-NN and SVR perform similarly and are the best algorithms.

- The performance of $k$-NN is more sensitive to the weather data used for the clustering than SVR and NN, and performs better with a longer weather vector. This is not surprising as nearest neighboring methods such as $k$-NN are more unstable and sensitive to noise, especially when only a subset of the examples are used for the prediction (the 5 nearest neighbors in our case). In addition, using only one aggregated weather variable is not sufficient to find the most similar days.

**B. Performance of the Non-Clustering Based Approaches**

Table 4.2 presents the accuracy results of the six non-clustering based approaches. Recall that they use only the previous PV data and do not consider the weather data. We can summarize the main results as follows:

- NN is the most accurate approach on all performance measures, achieving MAE=95.38 KW and RMSE=131.74 KW. The second best approach is SVR, followed by $k$-NN, ES, the persistent model and finally ARIMA.
- The improvement in MAE of NN compared to the other approaches is between 18% and 41%.
- All approaches except ARIMA outperformed the baseline persistent model.
- The machine learning approaches (NN, SVR and kNN) outperformed the statistical ones (SARIMA and ES).

**C. Comparison Between Clustering and Non-clustering Based Approaches**

By comparing the results from Table 4.1 and Table 4.2, we can see that the clustering based approaches outperform the non-clustering based approaches for clustering methods 1 and 2, for all prediction algorithms. For the other clustering methods the performance varies, which shows again the importance of the weather representation used for the clustering.

Table 4.2: Performance of the non-cluster based approaches

| Model | MAE (KW) | RMSE (KW) |
|---|---|---|
| k-NN | 119.17 | 180.29 |
| NN | 95.38 | 131.74 |
| SVR | 117.33 | 155.77 |
| ARIMA | 161.71 | 193.98 |
| ES | 123.46 | 168.75 |
| Persistent (B2) | 125.27 | 193.58 |

Fig. 4.6 visually compares the accuracy (MAE) of the clustering based approaches using the best clustering method (clustering method 2 - the 20-dimensional solar irradiance vector) and the non-clustering based approaches. We can see that the use of clustering improved the performance of all prediction algorithms the biggest improvement was achieved for *k*-NN (50%), followed by SVR (29%), the persistent model (28%), and NN (1%). The comparison of the RMSE results show a similar trend as MAE.

Thus, we can conclude that with appropriate weather representation, the clustering based approaches provide accurate prediction, outperforming their non-clustering based counterparts and also statistical non-clustering based methods such as ARIMA and ES.

**D. Clustering Based *k*-NN**

To better understand the performance of our best approach, the clustering based k-NN with clustering method 2, we calculated the accuracy for the days from each cluster separately as shown in Table 4.3. We can see that the best accuracy was achieved for cluster C1, closely followed by clusters C3 and C2, and finally cluster C4, which has considerably lower accuracy.

Fig. 4.7 shows the predicted and actual PV power output for typical consecutive days from each cluster. For days from C1 the PV power output is stable with a peak in the middle of the day reaching 850-900 KW, with very small differences between the actual and predicted values, mainly around the peak. For C3 the pattern is similar but the daily peak PV output is lower (600-700 KW) and there are more differences between the actual and predicted values for the last day. For C2 the pattern is also similar with a daily peak of 800 KW and slightly more differences between the actual

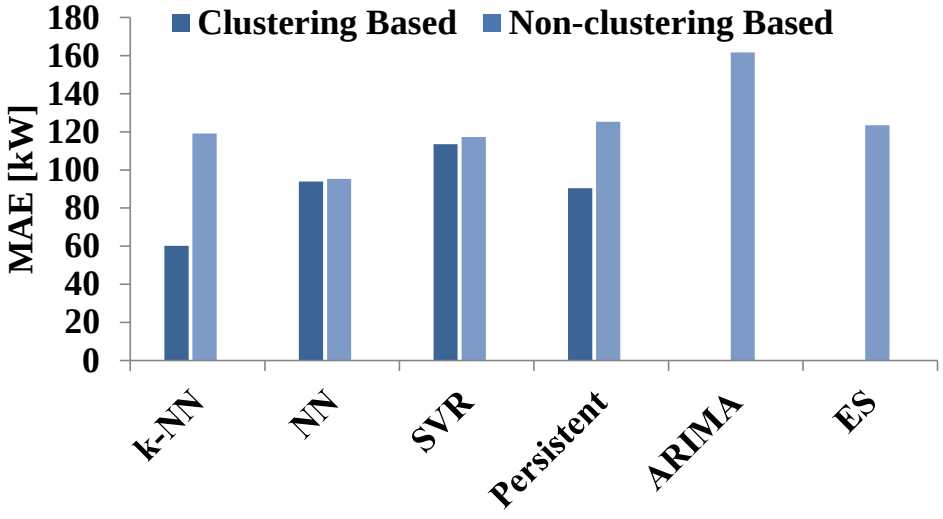


Figure 4.6: Comparison of clustering based and non-clustering based approaches

Table 4.3: Performance of *k*-NN for each cluster separately, with clustering method 2

| Cluster | MAE (KW) | RMSE (KW) |
|---|---|---|
| C1 | 48.77 | 76.31 |
| C2 | 61.47 | 84.54 |
| C3 | 52.94 | 76.92 |
| C4 | 133.19 | 206.43 |

and predicted values compared to C1. Cluster C4 is different there are random fluctuations in the generated PV output for each day; there is still one or several peaks in the middle of the day, reaching about 550-650 KW; this cluster is most difficult to predict as shown by the differences between the actual and predicted value. In summary, we can see that the four clusters represent days with different characteristics, with C1 most likely corresponding to sunny days with clear-sky, and C4 corresponding to days with changing solar irradiance due to rainy, cloudy, foggy or other conditions.

#### 4.1.2.4 Conclusion

In this case study, we considered the task of forecasting the PV power output for the next day at half-hourly intervals to evaluate the performance of direct clustering-based methods. The forecasting is done (1) directly, without the need to firstly predict the

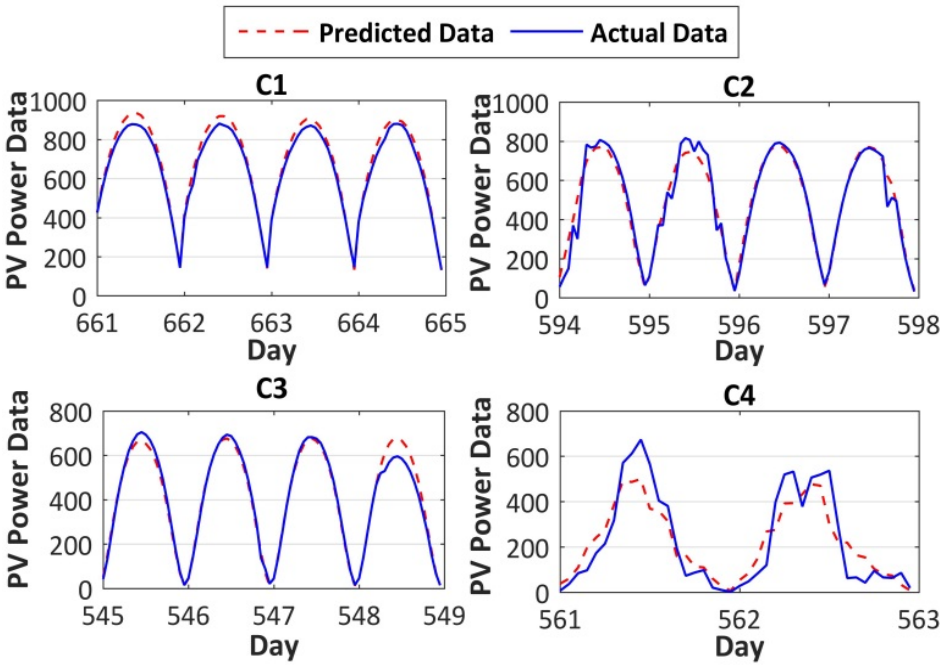


Figure 4.7: Actual vs predicted data for typical consecutive days from each cluster

solar irradiance and then convert it into PV output, and (2) simultaneously for all half-hourly intervals, rather than incrementally using the forecasts for the previous times.

The main idea behind our approaches is to cluster the days based on their weather characteristics and then built separate prediction models for each cluster using the PV data. We investigated if building such separate prediction models improves the accuracy, compared to building a single prediction model for all type of days.

We evaluated the performance of clustering based approaches that use five different weather representations based on temperature and solar irradiance, and three different prediction algorithms ($k$-NN, NN and SVR). We compared them with their non-clustering based counterparts, non-clustering based statistical methods such as ARIMA and ES, and clustering and non-clustering based persistent models used as baselines. Our evaluation was conducted using Australian data for two years.

We found that the most accurate prediction model was a clustering based $k$-NN, which uses a vector of half-hourly solar irradiance for the clustering. It achieved MAE = 59.81KW, RMSE = 96.18KW, significantly outperforming all other clustering and

non-clustering based methods and baselines. Our results also showed that the clustering based approaches did not always outperform their non-clustering based counterparts; the performance varied based on the weather representation used for the clustering, with solar irradiance representations favouring the clustering based approaches. In addition, our results showed that *k*-NN, NN and SVR, with and without clustering, performed better than ARIMA and ES.

## 4.2 Pair Pattern-based Method

The direct clustering-based prediction method proposed in the previous section does not consider the relationship between two consecutive days. In this section we describe an alternative clustering-based method - the Weather Pair Pattern-based (WPP) method. WPP builds specialized prediction models based on the cluster transition between consecutive days. It firstly partitions the days from the training data into clusters based on their weather characteristics and then uses the cluster label of the consecutive days to form pair patterns for each different type of cluster transition. A separate prediction model using NN or SVR is then built for each pair pattern group.

### 4.2.1 Methodology

Fig 4.8 summarizes the proposed WPP approach. There are four main steps:

1. Clustering of the days into $k$ groups based on the historical weather data and labelling the days with the cluster name;

2. Forming pairs of two consecutive days (pair patterns) based on the cluster labels, e.g. if there are two clusters C1 and C2, four pair patterns will be formed: {C1-C1}, {C1-C2}, {C2-C1} and {C2-C2};

3. Training a separate prediction model for each pair pattern using the historical PV data. It takes as an input the PV data of the previous day and predicts the PV data of the next day.

4. Predicting the PV power for a new day *d+1* by firstly identifying the cluster labels for days *d* and *d+1* based on the weather and weather forecast data respectively, and then using the prediction model for the corresponding pair pattern.

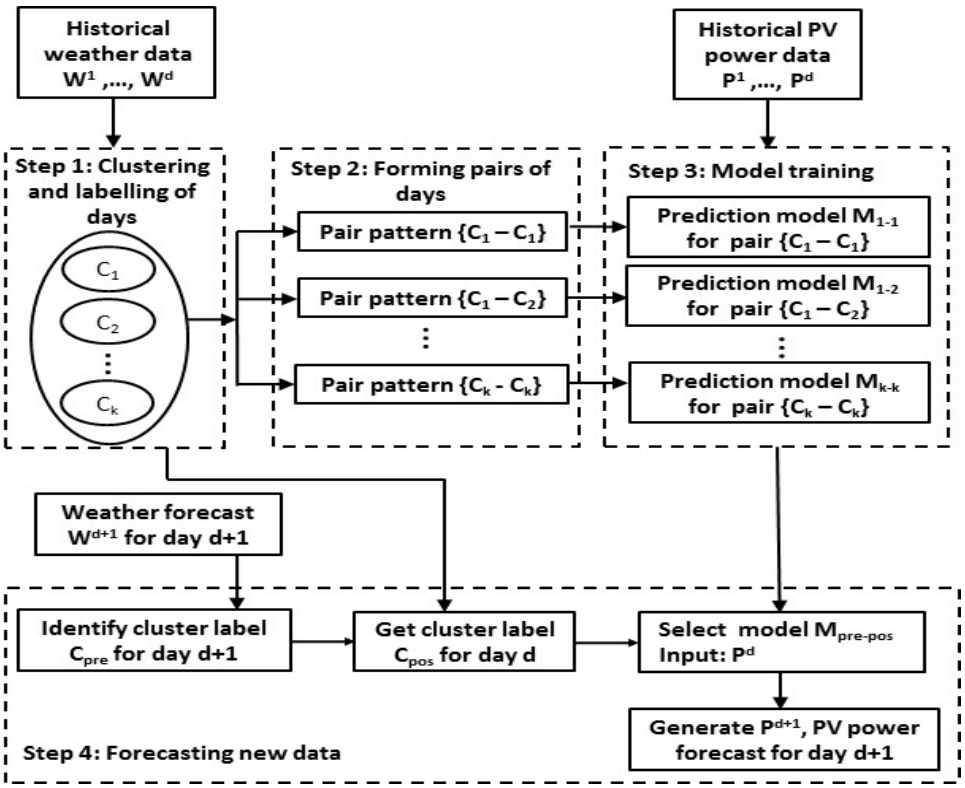


Figure 4.8: The WPP approach

For example, if day *d* belongs to C1 and day d+1 to C2, then the prediction model for {C1-C2} will be used.

#### 4.2.1.1 Clustering and Labelling of Days

Like the direct clustering-based methods, we use the *k*-means clustering algorithm to partition the days into *k* clusters based on the historical weather data, where each day is represented as a weather vector. We chose *k*-means as it is a classical and easy to implement algorithm, that has also shown good results in previous work on solar power prediction [8, 16].

For example, in the case study described in Section 4.2.2, we collect four weather variables: Solar Irradiance (SI), Temperature (T), Wind Speed (WS) and Humidity (H). These variables are typically available for most PV plants and are commonly used for PV prediction tasks. The daily weather data for day d can be represented as a

Table 4.4: Clustering evaluation results

| Indexes | Best k | Second Best k |
|---|---|---|
| Davies-Bouldin | 5 | 6 |
| Calinski-Harabasz | **2** | 3 |
| Silhouette | **2** | 3 |

12-dimensional vector:

$$W^d = [SI^d_{min}, SI^d_{avg}, SI^d_{max}, T^d_{min}, T^d_{avg}, T^d_{max},$$

$$WS^d_{min}, WS^d_{avg}, WS^d_{max}, H^d_{min}, H^d_{avg}, H^d_{max}]$$

To determine the number of clusters *k*, we calculate the Davies-Bouldin, Calinski-Harabasz and Silhouette indexes over the weather vector and then apply majority voting. The results are shown in Table 4.4 and based on them we select *k*=2 as the best number of clusters. We then partition the days into two clusters and label them as C1 and C2.

#### 4.2.1.2 Weather Type Pairs Forming and Model Training

After clustering and labelling of the days, weather type pairs are formed based on the labels of every two consecutive days. For example, if the label of day *d* is C1 and the label of day *d*+1 is C1, these two days will form a pair {day *d* to day *d*+1} and pair pattern {C1 - C2}. Since we have two clusters, the historical data can be partitioned into four pair patterns: {C1 - C1},{C1 - C2},{C2 - C1} and {C2 - C2}.

For each pair pattern group, a separate prediction model is trained using the PV power data of the days with this pattern. Specifically, for each pair of days, the PV vector of the first day is used as an input, while the PV data of the second day is used as a target. We implement prediction models using NN and SVR and compare their performance:

**A. Using NN**

Since we have two clusters, we train four NNs, one for each pair pattern group. Fig. 4.9 shows the structure of the NN model - a multi-layer perceptron with one hidden layer, 20 input and 20 output nodes. Each NN takes as an input the PV data of the previous

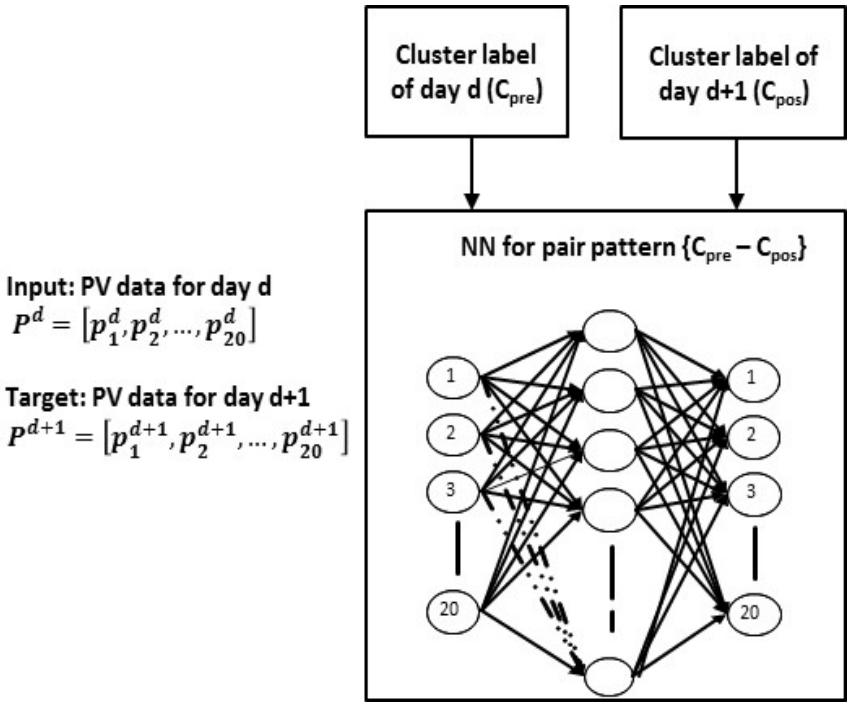


Figure 4.9: NN prediction model in WPP

day (first day in the pair) and as a target the PV data of the next day (second day in the pair). The input and output vectors are 20-dimensional corresponding to the 20 half-hourly PV power values of the day. Thus, the prediction for all half hours of the next day is generated simultaneously, not incrementally.

As a training algorithm we use the Levenberg-Marquardt algorithm. The NN parameters, including the number of hidden nodes, are selected based on the NN performance on the validation set. We train NNs with different combinations of parameters and then select, for each pair pattern, the NN with the best performance on the validation set.

**B. Using SVR**

We follow the same methodology as for the NN models, except that we train 20 SVRs for each pair pattern, one for each of the 20 half-hourly PV values for the next day, since SVR has only one output We follow the same methodology as for the NN models, except that we train 20 SVRs for each pair pattern, one for each of the 20 half-hourly PV values for the next day, since SVR has only one output.This process is similar with

the case of direct clustering-based methods.

#### 4.2.1.3 Forecasting New Data

As shown in Fig. 4.8, the last step of WPP is to make predictions for the new day *d+1*. Firstly, the weather forecast vector for day *d+1* is obtained and compared with the cluster centroids using a distance measure (we used the Euclidean distance) and day *d+1* is labelled with the cluster label of the closest cluster $C_{pos}$. The cluster label for day *d*, $C_{pre}$, is retrieved from the already labelled data from step 1. Finally, an already trained NN or SVR model for the pair pattern $\{C_{pre}$ - $C_{pos}\}$ is selected and used to predict the PV data for day *d+1* by taking as an input the PV data for *d*.

### 4.2.2 Case Study

We conducted a case study to evaluate the performance of the WPP-based method for PV power forecasting. We compared the performance of WPP with the direct clustering-based method, and also with non-clustering-based methods and persistence models.

#### 4.2.2.1 Experimental Setup

To make the comparison throughout the thesis consistent, in this case study we again consider the task of directly and simultaneously predicting the PV power data of the next day at 30-min intervals, using historical weather data, historical PV data and weather forecast.

More specifically, given: (i) a time series of historical PV power outputs up to the day *d*: $[p^1, p^2, p^3, ..., p^d]$, where $p^i = [p^i_1, p^i_2, p^i_3, ..., p^i_{20}]$ is a vector of 20 half- hourly power outputs for the day *i*, (ii) a time series of weather vectors for the same days: $[W^1, W^2, W^3, ..., W^d]$, and (iii) the weather forecast for day *d+1* : $WF^{d+1}$, our goal is to forecast $PV^{d+1}$, the half-hourly power output for the next day *d+1*.

**A. Data Sources and Preprocessing**

We use the same source of PV data as in the direct clustering-based methods' case study. More specifically, we use the PV power data for two complete years - from

1 January 2013 to 31 December 2014. For each day, we only select data during the daylight period, from 7 am to 5 pm.The original data is collected at 1-min intervals and contains $2 \times 365 \times 600 = 438,000$ measurements. There are 1,518 missing values, which accounts for approximately 0.35% of the total data. Each missing value is replaced by the average of the values from the previous 5 min. We aggregate data into 30-min intervals as our task is to make half-hourly predictions for the next day. Thus, we have 20 PV values for one day and 14,600 $(20 \times 365 \times 2)$ values in total for two years. The data is normalized to the interval [0-1].

As WPP requires weather features and weather forecasts to determine the cluster labels, we also collect weather data. Unlike the direct clustering-based methods' case study, we collect a different set of weather variables and form a 12-dimensional vector to represent the day. The four variables are Solar Irradiance (SI), Temperature (T), Wind Speed (WS) and Humidity (H); the vector consists of the maximum, minimum and average daily value of these four variables.

As the weather forecast data is not available retrospectively, we use the actual weather data with added 10% noise as the weather forecast.

**B. Training, Validation and Testing Sets**

We follow the same procedure to split the data into three different sets as we did in section 4.2.3 for consistent comparison. Specifically, We divide the solar power and corresponding weather data into three non-overlapping subsets: 50% (day 1 - day 365) for training, 25% (day 366 - day 550) for validation and 25% (day 551 - day 730) for testing. The training set is used to build the prediction models; the validation set is used for parameter selection and the testing set is used to evaluate the performance of the proposed clustering-based approach and the other methods used for comparison.

#### 4.2.2.2 Methods for Comparison

To comprehensively evaluate the performance of WPP, we implement 4 different groups of methods for comparison. This includes a comparison with the direct clustering-based method using the new weather features. Specifically, the four groups of methods are: group 1: clustering-based machine learning models (with NN and SVR as prediction algorithms), group 2: non-clustering-based machine learning models (NN, SVR and $k$-NN), group 3: statistical models (ARIMA and ES), and group 4: baselines -

three simpler (persistence) models, two using clustering and one not using clustering.

**Group 1: Clustering-Based Machine Learning Models**

Fig. 4.10 shows the main steps of the implemented clustering-based models. In step 1, the historical weather data is clustered into *k* groups (*k*=2 in our case) and labelled in the same way as in WPP. In step 2, a prediction model is trained for each cluster as opposed to each pair pattern as in WPP. The model takes as an input the PV power data of day *i* and as a target the PV power data of the next (closest in time) day *j* from the same cluster. Note that in contrast to WPP, the day *j* is not necessarily day *i+1*, the day following *i*, as only the days in the cluster are considered and *i+1* may be in another cluster. As prediction algorithms we used NN and SVR for consistency with WPP.

To predict the PV data for the new day *d+1* in step 3, the cluster label for *d+1* is firstly identified by comparing the weather forecast vector for *d+1* with the cluster centroids using the Euclidean distance and assigning the day to the cluster of the closest centroid, $C_x$. The prediction for *d+1* is then computed by the trained prediction model (NN or SVR) for $C_x$, using as an input the PV power data for the most recent day *s* (the day closest in time to day *d+1*) from cluster $C_x$.

In summary, both WPP and the Group 1 methods cluster and label the days based on the weather data. However, WPP forms pair patterns of consecutive days and builds a prediction model for each pair pattern  given the PV power for the previous day, it predicts the PV power for the next day: {train, target}=$\{P^i, P^{i+1}\}$. The Group 1 methods build a prediction model for each cluster using pairs of days closest in time from each cluster (i.e. the consecutive days for the cluster): {train, target}=$\{P^i, P^j\}$.

As we did with the direct clustering-based methods, we implement both NN and SVR versions:

1. Clustering-Based NN

   For each cluster, we train a feedforward NN with 20 input nodes and 20 output nodes which correspond to the half-hourly PV power data of day *i* and the closest day *j* from the same cluster, respectively. Similar to the training of WPP-based NN, we implement NN with different parameter combinations and select the one that performs best on the validation dataset.

2. Clustering-Based SVR

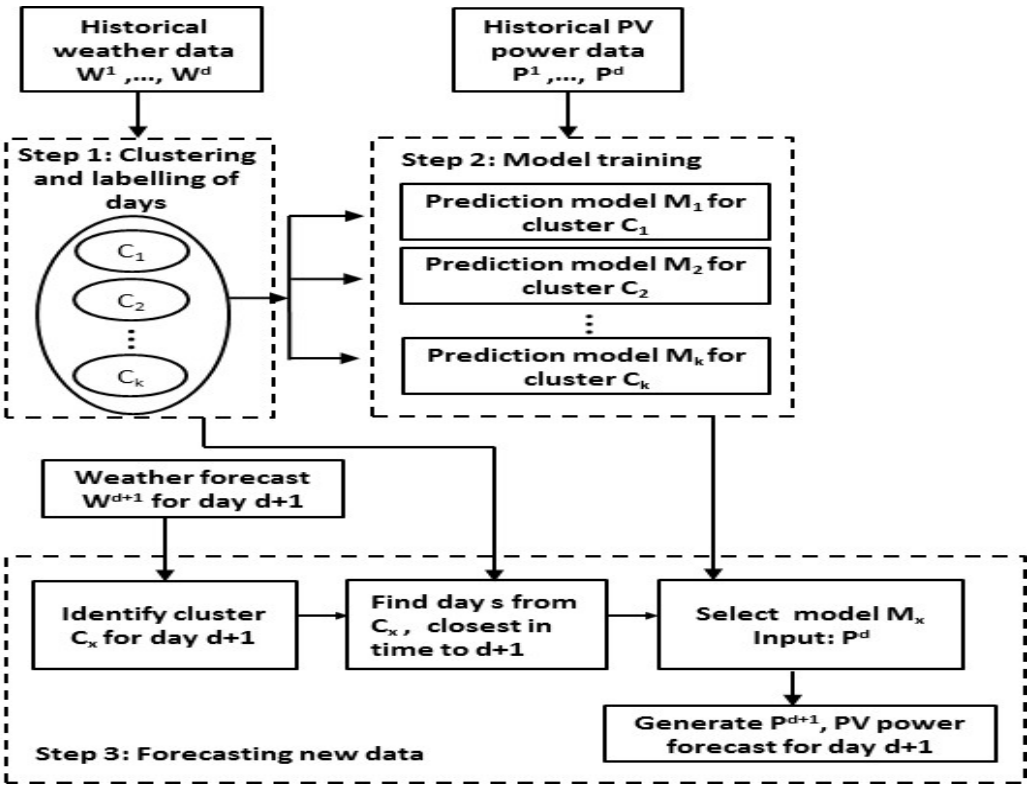


Figure 4.10: Clustering-based methods used for comparison

The training of clustering-based SVR follows the same methodology as for the clustering-based NN, using the PV power data of day *i* and the closest day *j* in the same cluster as inputs and targets respectively. However, in contrast to the NN method where we build one NN for each cluster, which predicts all 20 power outputs for the next day, here we create 20 SVR models for each cluster, each predicting one half-hourly value for the next day, since SVR has only one output.

**Group 2: Non-Clustering-Based Models**

Instead of clustering the days and building a prediction model for each weather type, we develop three general prediction models for all weather types using NN, SVR and kNN.

1. NN

   The NN has the same structure as the non-clustering-based NN. It takes as an input the PV data of day *i* and as an output the PV data of the following day *i+1*.

For parameter selection we used the same procedure as for the clustering-based NN.

2. SVR

   The non-clustering-based SVR is similar to the clustering based SVR except that: a) there is only one set of 20 SVR models for all days, not a different set for each cluster of days, and b) as inputs it uses the PV power data from the most recent day d, not from the most recent day j from the same cluster. The parameters of the non-clustering-SVR were selected using on same procedure as for the clustering-based SVR.

3. *k*-NN

   To predict the power output for the next day d+1 using *k*-NN, we firstly obtain the weather forecast vector $WF^{d+1}$ or day *d+1* from the weather forecast. Then the *k* nearest neighbours for *d+1* are determined - these are the days with the most similar weather vectors $WF$ (smallest distance) to $WF^{d+1}$. In our experiments, we used the Euclidean distance and the value of *k* was determined based on the performance on the validation set.

   To generate the PV power prediction for day *d+1*, *k*-NN takes the average of the half-hourly PV data of the *k* neighbors: $\widehat{p}_t^{d+1} = \frac{1}{k}\sum_{j=1}^{k} p_t^j$, where $\widehat{p}_t^{d+1}$ is the predicted PV value at time *t* for day *d+1*, *t*=1,,20 and *j* is the day index of the neighbors.

**Group 3: Statistical Models**

For comparison, we also implemented two statistical models: a seasonal ARIMA and a Holt-Winter ES.

**Group 4: Baselines**

We develop three persistence models, used as baselines.

The first baseline ($B_{persistent}$) does not use clustering. It considers the half-hourly PV power outputs from the previous day *d* as the prediction for the next day *d+1*. This means that the prediction for day *d+1*, $\widehat{p}^{d+1} = [\widehat{p}_1^{d+1}, \widehat{p}_2^{d+1}, ..., \widehat{p}_{20}^{d+1}]$, is given by $P^d = [p_1^d, p_2^d, ..., p_{20}^d]$. This baseline is called a persistence model.

The second baseline ($B_{cluster}$) uses the clustering of days and is a cluster based persistence model. To make a prediction for day *d+1*, it finds the most recent day (closest in time) to *d+1* from the same cluster, and predicts its half-hourly PV power outputs. This means that the prediction for day *d+1*, $\widehat{p}^{d+1} = [\widehat{p}_1^{d+1}, \widehat{p}_2^{d+1}, ..., \widehat{p}_{20}^{d+1}]$ is given by $P^d = [p_1^r, p_2^r, ..., p_{20}^r]$, where both days *r* and *d+1* belong to same cluster $C_x$ and day *r* is the closest to day *d+1* among all days in $C_x$.

The third baseline ($B_{pair}$) is a WPP based persistent model. To make a prediction for day *d+1*, it follows the WPP method to find the cluster labels for days *d* and *d+1* and their pair pattern $C_{pre}$ to $C_{pos}$. It then searches the historical data to find the most recent pair of consecutive days (day *s* to day *s+1*) with the same pair pattern and uses the PV power data of day *s+1* as the prediction. More specifically, the prediction for $\widehat{p}^{d+1} = [\widehat{p}_1^{d+1}, \widehat{p}_2^{d+1}, ..., \widehat{p}_{20}^{d+1}]$ is given by $P^d = [p_1^{s+1}, p_2^{s+1}, ..., p_{20}^{s+1}]$ such that the pair patterns from day *d* to day *d+1* and day *s* to day *s+1* are the same, and days *s* and *s+1* are the closest pair of days with the same pair pattern.

#### 4.2.2.3 Results and Discussion

**A. Overall Performance**

Table 4.5 shows the accuracy results (MAE and RMSE) for all methods and Fig. 4.12 shows the MAE results from Table 4.5 in sorted order for visually comparison.

The results show that the proposed WPP methods are the most accurate group of methods, with WPP using NN outperforming WPP using SVR and achieving MAE = 99.74 KW and RMSE = 136.78 KW. All pair-wise differences in accuracy between the two WPP methods and the other methods used for comparison are statistically significant at $p \leq 0.001$ except the difference between the WPP-based SVR and the clustering-based NN which is statistically significant at $p \leq 0.05$.

The second best group of methods is the clustering-based, followed by the two baselines $B_{pair}$ and $B_{cluster}$, then the non-clustering machine learning methods and ES, and finally the persistent baseline $B_{persistent}$ and ARIMA.

The first two groups of approaches both use clustering, either pair-based or standard, and significantly outperform their nonclustering-based counterparts (SVR and NN). This shows that grouping the days based on their weather patterns and building specialized prediction models for each group is useful.

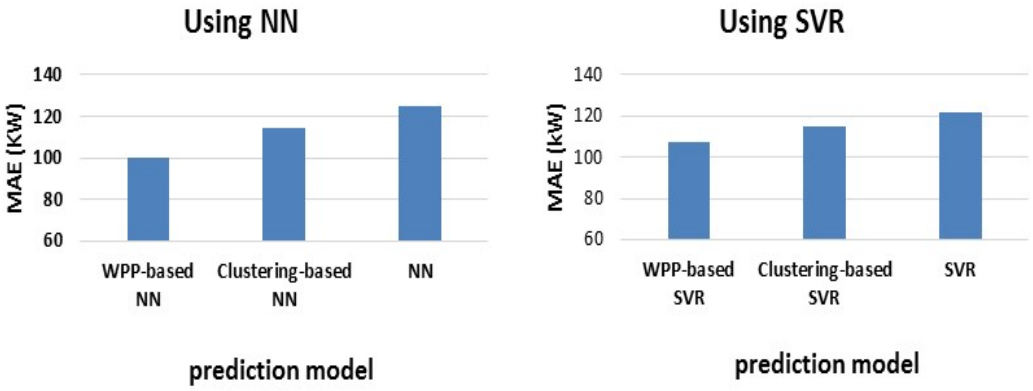


Figure 4.11: Comparision of the WPP, clustering-based and non-clustering-based approaches for NN and SVR separately

Fig. 4.11 compares the accuracy of WPP with NN and SVR with its clustering-based and non-clustering counterparts using these algorithms. It can be seen that the proposed WPP approach outperforms the other methods for both NN and SVR. When NN is used, WPP improves the accuracy (MAE) of the clustering based and non-clustering-based methods with 12.55% and 20.07% respectively. When SVR is used, this improvement is 6.55% and 11.58% respectively. We can also see that NN performs better than SVR in the clustering based methods.

The statistical methods show mixed results. While ES performs relatively well and similarly to the non-clustering based machine learning methods, ARIMA is the least accurate method, performing significantly worse than all other methods, including the baselines.

By comparing the performance of the baselines we can see that $B_{pair}$ and $B_{cluster}$ perform very well, and achieve higher accuracy than all non-clustering-based machine learning and statistical methods. Both $B_{pair}$ and $B_{cluster}$ are clustering-based baselines, which again shows the advantages of using clustering to group the days with similar weather. $B_{pair}$ is the best performing baseline and is closely related to the WPP approach. In summary, we can see that the proposed WPP approach, with both NN and SVR, is more effective than the standard clustering based methods and the non-clustering-based machine learning and statistical methods. It significantly improves the performance of its non-clustering-based NN and SVR counterparts. The

Table 4.5: Accuracy of all methods

| Methods | MAE (kW) | RMSE (kW) |
|---|---|---|
| Group 1: WPP-based | | |
| (P1) WPP-based NN | 99.74 | 136.78 |
| (P2) WPP-based SVR | 107.36 | 144.39 |
| Group 2: Clustering-based | | |
| (C1) Clustering-based NN | 114.06 | 158.58 |
| (C2) Clustering-based SVR | 114.88 | 149.70 |
| Group 3: Non-clustering-based | | |
| (N1) NN | 124.79 | 166.15 |
| (N2) k-NN | 126.58 | 183.18 |
| (N3) SVR | 121.42 | 159.63 |
| Group 4: Statistical | | |
| (S1) ARIMA | 164.22 | 196.69 |
| (S2) ES | 125.20 | 171.42 |
| Group 5: Baselines | | |
| (B1) Bpersistent | 129.46 | 197.36 |
| (B2) Bcluster | 120.73 | 182.69 |
| (B3) Bpair | 116.80 | 179.45 |

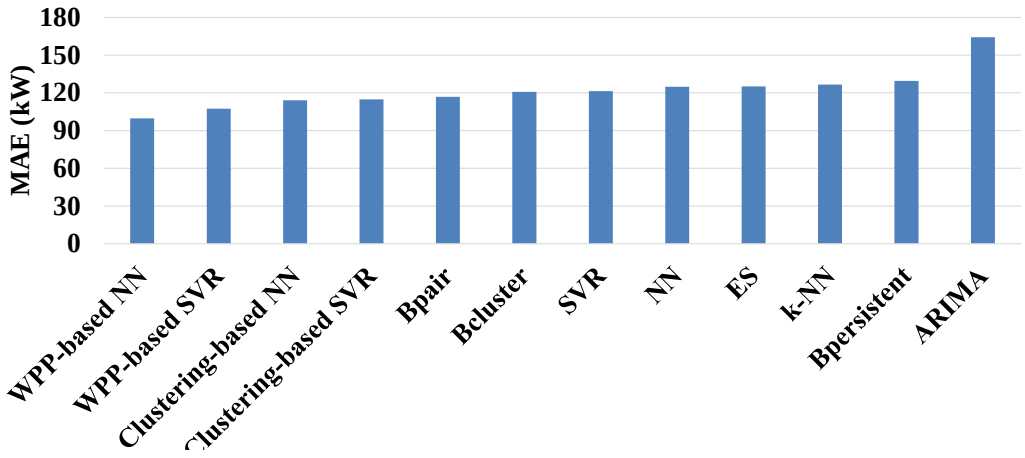


Figure 4.12: Performance of all prediction models (MAE)

pair-based persistent model, closely related to WPP, also shows good results and outperforms all non clustering-based methods.

**B. WPP Cluster Analysis**

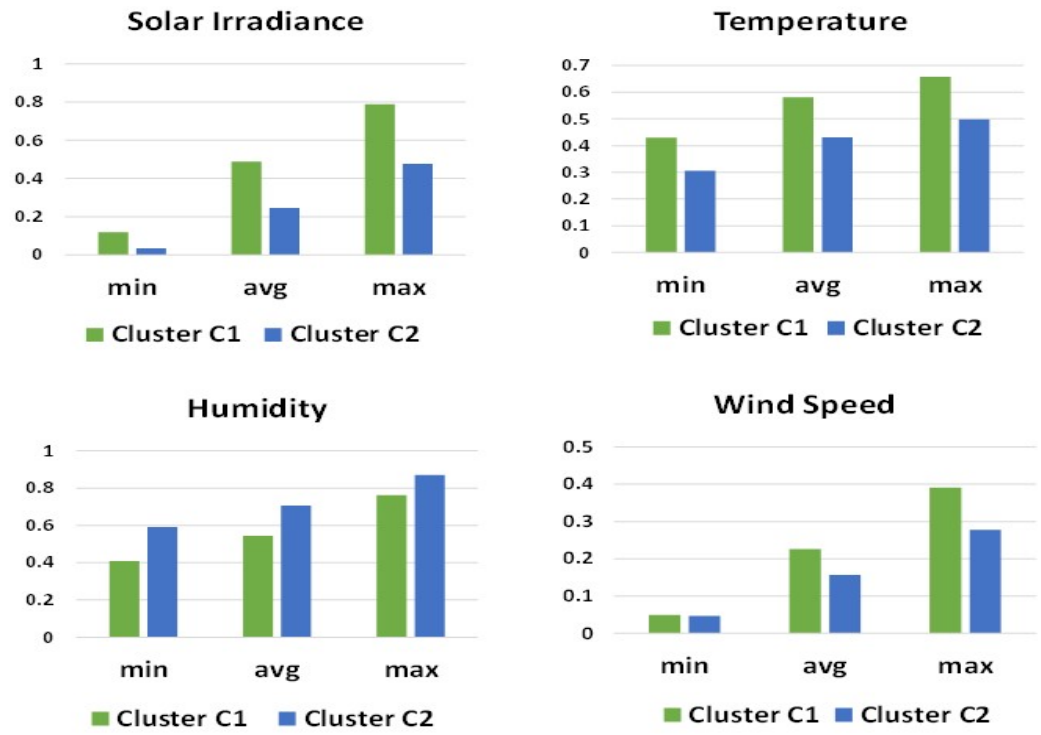


Figure 4.13: Comparison of the centroid values

To get a better understanding of the performance of WPP, we examine the cluster centroids which summarize the weather characteristics of the days in each cluster. Table 4.6 shows the centroids of the two clusters and Fig. 4.13 compares the centroid values. The days in cluster $C_1$ have considerably high solar irradiance and temperature than the days in cluster $C_2$, and also lower humidity and higher wind speed. Thus, the days in cluster $C_1$ are most likely to be clear sunny days, while the days in cluster $C_2$ are more likely to be cloudy or rainy.

**C. WPP Performance for Each Pair Pattern**

We also investigate the performance of WPP for each of the four pair patterns. Table 4.7 provides detailed information for each pair pattern prediction model using NN. We can see that most of the days in the training and validation set belong to pair patterns $\{C_1$ - $C_1\}$ and $\{C_2$ - $C_2\}$ , and the accuracy of the corresponding prediction models is the highest (MAE = 92.24 and 92.74 kW, respectively). The prediction models for pair patterns $\{C_2$ - $C_1\}$ and $\{C_1$ - $C_2\}$ are less accurate, especially the one for

Table 4.6: Centroids of the two clusters

| Cluster | Feature | Min | Average | Max |
|---|---|---|---|---|
| $C_1$ | Solar irradiance | 0.1190 | 0.4879 | 0.7899 |
| | Temperature | 0.4296 | 0.5814 | 0.6575 |
| | Humidity | 0.4087 | 0.5429 | 0.7632 |
| | Wind speed | 0.0492 | 0.2261 | 0.3914 |
| $C_2$ | Solar irradiance | 0.0350 | 0.2462 | 0.4770 |
| | Temperature | 0.3055 | 0.4302 | 0.4985 |
| | Humidity | 0.5915 | 0.7066 | 0.8708 |
| | Wind speed | 0.0468 | 0.1568 | 0.2778 |

Table 4.7: Detailed information for each pair pattern prediction model for WPP-based NN

| WPP Model | Training+ Validation Size | Testing Size | Best Hidden Layer Size | MAE (kW) | RMSE (kW) |
|---|---|---|---|---|---|
| C1 - C1 | 267 | 105 | 9 | 92.24 | 127.04 |
| C1 - C2 | 50 | 17 | 27 | 153.25 | 195.99 |
| C2 - C1 | 49 | 18 | 15 | 108.19 | 152.26 |
| C2 - C2 | 183 | 40 | 24 | 92.74 | 122.27 |
| Overall | 549 | 180 | N/A | 99.74 | 136.78 |

$\{C_1$ - $C_2\}$ , which can be explained with the smaller size of the training and validation sets, and also with the bigger weather variability when there is a transition between sunny and cloudy days, which makes predicting the PV power more difficult.

**D. WPP Performance for Each Cluster**

We further analyse the performance of WPP for each cluster. The days in the testing set are labelled with their weather cluster and Table 4.8 shows the per-cluster results

Table 4.8: Per-cluster comparison of WPP-based NN and Clustering-based NN

| Cluster | WPP-based NN | | Clustering-based NN | |
|---|---|---|---|---|
| | MAE (kW) | RMSE (kW) | MAE (kW) | RMSE (kW) |
| C1 | 94.62 | 131.17 | 107.53 | 155.79 |
| C2 | 110.79 | 148.15 | 128.17 | 164.43 |

for WPP using NN and the clustering-based NN which is used for comparison. We can see that both methods give higher accuracy for sunny days (cluster $C_1$) than for cloudy or rainy days (cluster $C_2$).

**E. Computational Time**

The training and prediction times of WPP are both acceptable. The training time was less than 5 minutes for both WPP with NN and SVR. As the training is done offline, this is an acceptable requirement for both offline and online applications of WPP. WPP was also fast at making predictions for new instances taking only a few seconds.

#### 4.2.2.4 Conclusion

In this study, we propose a new weather type pair pattern approach, called WPP, to directly and simultaneously predict the PV power for the next day at half-hourly intervals. This new method extends the direct clustering-based methods discussed in Section 4.2 and aims to better utilize the similarity of PV power profiles between consecutive days. WPP firstly partitions the days from the training data into clusters based on their weather characteristics and then uses the cluster label of the consecutive days to form pair patterns. A separate prediction model using NN or SVR is then built for each pair pattern group, and used to make predictions for the new days.

We evaluate the performance of WPP using PV and weather data for 2 years from a large PV plant in Australia. The results show that WPP was the most accurate method, outperforming the standard clustering-based methods, and also non-clustering- based methods using NN, SVR and *k*-NN, statistical (seasonal ARIMA and ES) and three persistence baselines. All differences in accuracy between WPP and the other methods are statistically significant.

The two highly accurate types of methods, WPP and the standard clustering-based, utilise clustering, which shows that grouping the days based on their weather patterns and building specialized prediction models for each group is useful. By utilizing the similarity between the weather and PV profiles of the consecutive days, WPP is able to improve the accuracy of the standard clustering-based methods which do not consider this information. The good performance of the pair-based baseline also reinforces the usefulness of this information. The best accuracy was achieved by the WPP based NN (MAE = 99.74 KW), which is an improvement of 12.55% and 20.07% compared to

the clustering-based NN and the non- clustering-based NN respectively.

## 4.3 Summary

This chapter discusses clustering-based methods for solar power prediction tasks. Most of the existing methods build a single prediction model for all weather types. In contrast, the clustering-based methods take advantage of the similarity of the PV profiles of days with similar weather patterns. To utilize this similarity, they partition the days into groups with similar weather characteristics using clustering and then build a separate, specialized, prediction model for each cluster.

Section 4.1 proposed a new approach for predicting the solar power based on direct clustering. The main idea is to cluster the days based on their weather characteristics and build a separate prediction model for each cluster using the PV solar data. The case study uses Australian PV data for two years and shows that the direct clustering-based method is more accurate than the general, single prediction model for all types of days.

Section 4.2 introduces the weather pair patterns (WPP) clustering-based method, which extends the direct clustering-based method by utilizing the relationship between consecutive days. WPP builds a separate prediction model for each type of cluster transition between two consecutive days. The case study shows that WPP outperforms the direct clustering-based method.

In summary, our results show that methods using clustering to partition the days into groups with similar weather characteristics and then build a separate prediction model for each group, were more beneficial for solar power forecasting, than methods using one prediction model for all types of days.

# Chapter 5

# Ensemble Methods

This chapter describes ensemble methods for solar power prediction. Most of the previous work on time series prediction has focused on using single prediction models. Ensemble methods combine the predictions of several prediction models in some way to form the final prediction, and have been shown to be very competitive [80, 82, 82]. One of the key ideas for building successful ensembles is to include diverse ensemble members [96], i.e. ensemble members with different expertise, which can be used to handle different scenarios [97, 98]. By combining the predictions of the ensemble members using appropriate combination strategies, the ensemble may achieve higher forecasting accuracy than the individual ensemble members.

Diversity can be generated in various ways: (i) by altering the training data for each ensemble member (e.g. by using different data sources or different data subsets; different pre-processing methods or by introducing noise); (ii) by altering the feature set for each ensemble member (e.g. by using feature subsets) or (iii) by altering the prediction models (e.g. using NNs with different topologies, different parameters or training algorithms).

Based on the adaptation strategy, ensembles can be divided into two main categories: static and dynamic:

- **Static ensembles** combine the predictions of the individual ensemble members in the same way (e.g. by taking the average of the individual predictions), regardless of the changes in the time series or the relative performance of the ensemble members.

- **Dynamic ensembles** combine the predictions of the ensemble members adaptively. For example, by tracking the error of the ensemble members on recent data and weighting their contribution for the new data accordingly or by predicting the performance of the ensemble members for the new data and weighting their contribution accordingly . In this way the ensemble is adapted to the changes in the time series and the changes in the relative performance of the ensemble members.

In this chapter, we propose several static and dynamic ensembles for solar power forecasting, and compare them with traditional ensembles such as Bagging, Boosting and Random Forest. Section 5.1 describes the static ensembles and the dynamic ensembles based on previous performance. Section 5.2 describes the dynamic ensembles based on predicted future performance using meta-learning. The main results of this chapter have been published in [99, 100].

## 5.1 Static Ensembles and Dynamic Ensembles Based on Previous Performance

As already mentioned, effective ensembles include diverse ensemble members. We propose three strategies to generate diverse ensemble members, and based on them create three types of static ensembles combining NNs. The final prediction is formed by taking the average of all individual predictions. In addition, we propose four strategies for constructing dynamic ensembles of NNs which adaptively weight the contribution of the ensemble members based on their recent performance.

### 5.1.1 Static Ensembles

The key idea of using ensembles for forecasting is to utilize the benefits generated by the diversity among the individual ensemble members. We generate diversity using three strategies: 1) by altering the training data, 2) by altering the feature set and 3) by combining these two strategies. Based on these strategies we construct the following static ensembles of NNs:

1. **Ensemble 1 (EN1).** It uses random example sampling of the training data to generate different training subsets for the ensemble members.

2. **Ensemble 2 (EN2).** It uses random feature sampling to generate data subsets with different features for the ensemble members.

3. **Ensemble 3 (EN3).** It combines the previous two strategies to generate data subsets with both randomly sampled examples and features.

As a base classifier (ensemble member) in all ensembles we use a single NN, trained with the Levenberg-Marquardt version of the backpropagation algorithm. It takes as an input the PV power of the previous day (all values of the day or less, depending on the ensemble type) and predicts the PV power of the next day. In this study, we still consider the task of predicting the half-hourly data of the next day simultaneously (maximum 20 data points for each single day).

Using the previous day as a prediction model was motivated by Wang et al. [95], where it was shown that the PV power of the previous day is a good prediction model for the PV power of the next day. Thus, a single ensemble member in all ensembles is a multi-layer perceptron with 1 hidden layer, 20 or less input nodes and 20 output nodes. The number of hidden neurons in all NNs was set to the average of the input and output neurons.

To make a prediction for a new day *d+1*, the predictions of the individual ensemble members are combined by taking the arithmetic average:

$$\hat{P}^{d+1} = \frac{1}{s}\sum_{j=1}^{S} \hat{p}_j^{d+1}$$

where $\hat{p}_j^{d+1}$ the prediction of ensemble member *j* for day *d+1*, and the index *j* is over all *S* ensemble members.

#### 5.1.1.1 EN1 - Random Example Sampling

Fig. 5.1 shows the structure of ensemble EN1 which uses the first strategy - random example sampling. EN1 consists of *S* NNs. Each NN is trained using a different training subset generated by sampling with replacement, with a given sampling rate $Rs$ (e.g. 75%). We firstly create a bootstrap sample for each NN which contains only $Rs\%$ of the d examples for the first year, which is the whole data used for training and validation. These examples are then randomly divided into training set (70%, used for

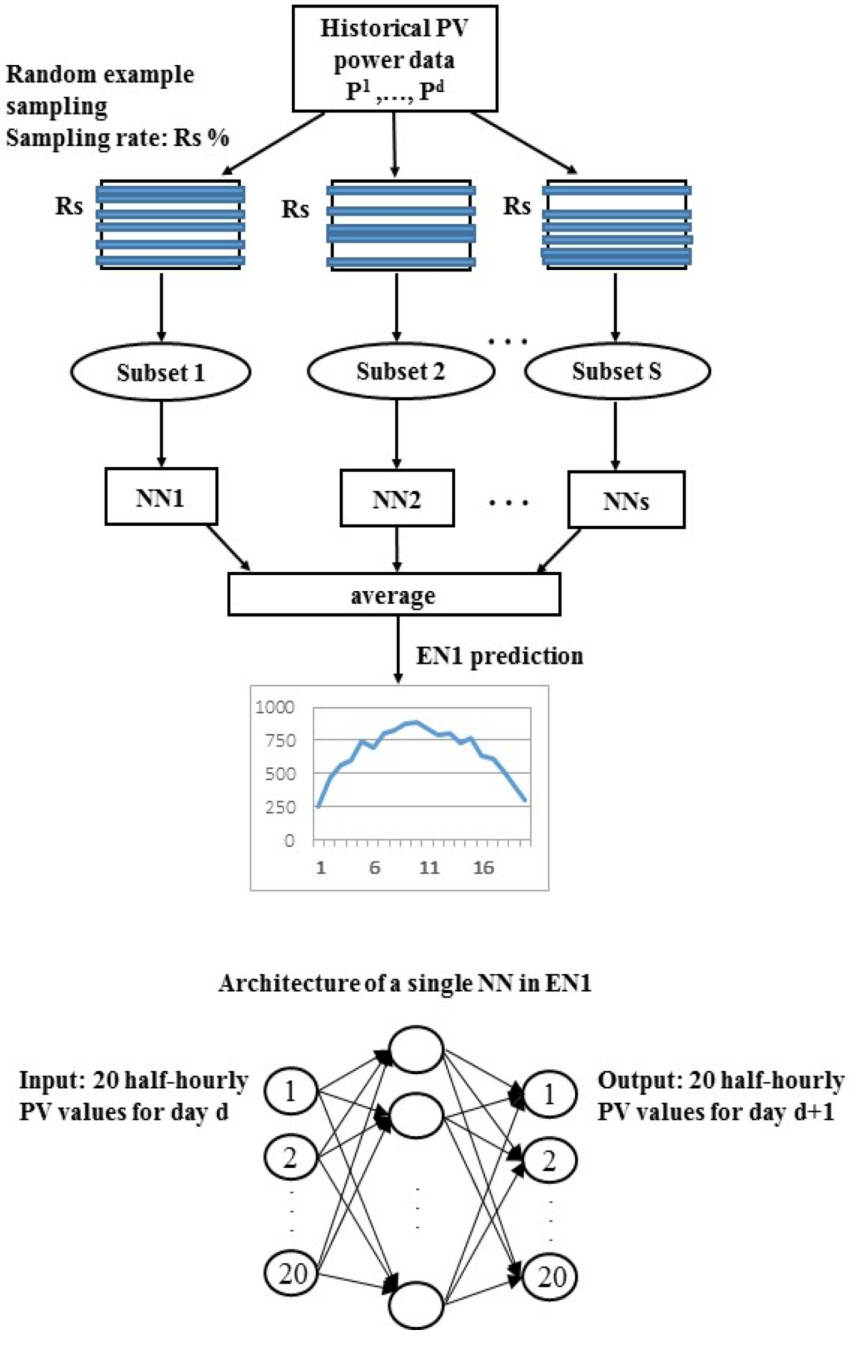


Figure 5.1: Ensemble EN1 using random example sampling

training of the NN) and validation set (30%, used for selecting the NN parameters). Thus, the training set for a single NN will contain a smaller number of examples than

the original training set and will have the same number of features.

The difference between our and the standard bootstrap sampling is that instead of sampling *n* instances, where *n* is the number of examples, we sample a smaller number, determined by the sampling rate. Since the sampling is with replacement, the maximum number of unique examples in each training set is $Rs\% \times d \times 70\%$.

**A single ensemble member:** A single NN in EN1 has a full set of nodes (20 for this study) and output nodes, corresponding to the half-hourly PV power of the previous and next day respectively.

**Parameters:** In our experiments we used $S = 30$ ensemble members (single NNs) and a sampling rate Rs=75%. The sampling rate was selected by varying Rs from 25% to 75% with an increment of 25%, evaluating the performance of the ensemble on the validation set and selecting the best Rs (the one with the highest accuracy).

**Number of unique examples:** We build 30 member NNs in the ensemble. The number of unique examples for the selected sampling rate varied from 46% to 59% for the all ensemble members.

#### 5.1.1.2 EN2 - Random Feature Selection

Fig. 5.2 shows the structure of the second static ensemble (EN2) which uses random feature selection. EN2 also consists of *S* NNs. However, unlike EN1 where each NN uses all features and a subset of all training examples, in EN2 each NN uses a subset of all features and all training examples. The number of features is determined by the feature sampling rate $Rf$. We firstly form a feature bootstrap sample for each single NN which contains all *d* examples but only $Rf\%$ of their features, which are randomly selected using sampling with replacement. Each sample is then randomly split into training and validation set containing 70% and 30% of the examples, respectively. Thus, the maximum number of unique features in each training set is $Rf\% \times f$, where $f$ is the number of all features.

**Single ensemble member:** A single NN in EN2 has *k* input nodes ($k \leq Rf\% \times f$) corresponding to the selected features (PV power outputs for the previous day) and 20 output nodes corresponding to all 20 half-hourly PV power outputs for the next day.

**Parameters:** As in EN1, we used $S = 30$ ensemble members and the same procedure for selecting the best feature sampling rate $Rf$ based on the validation set. The

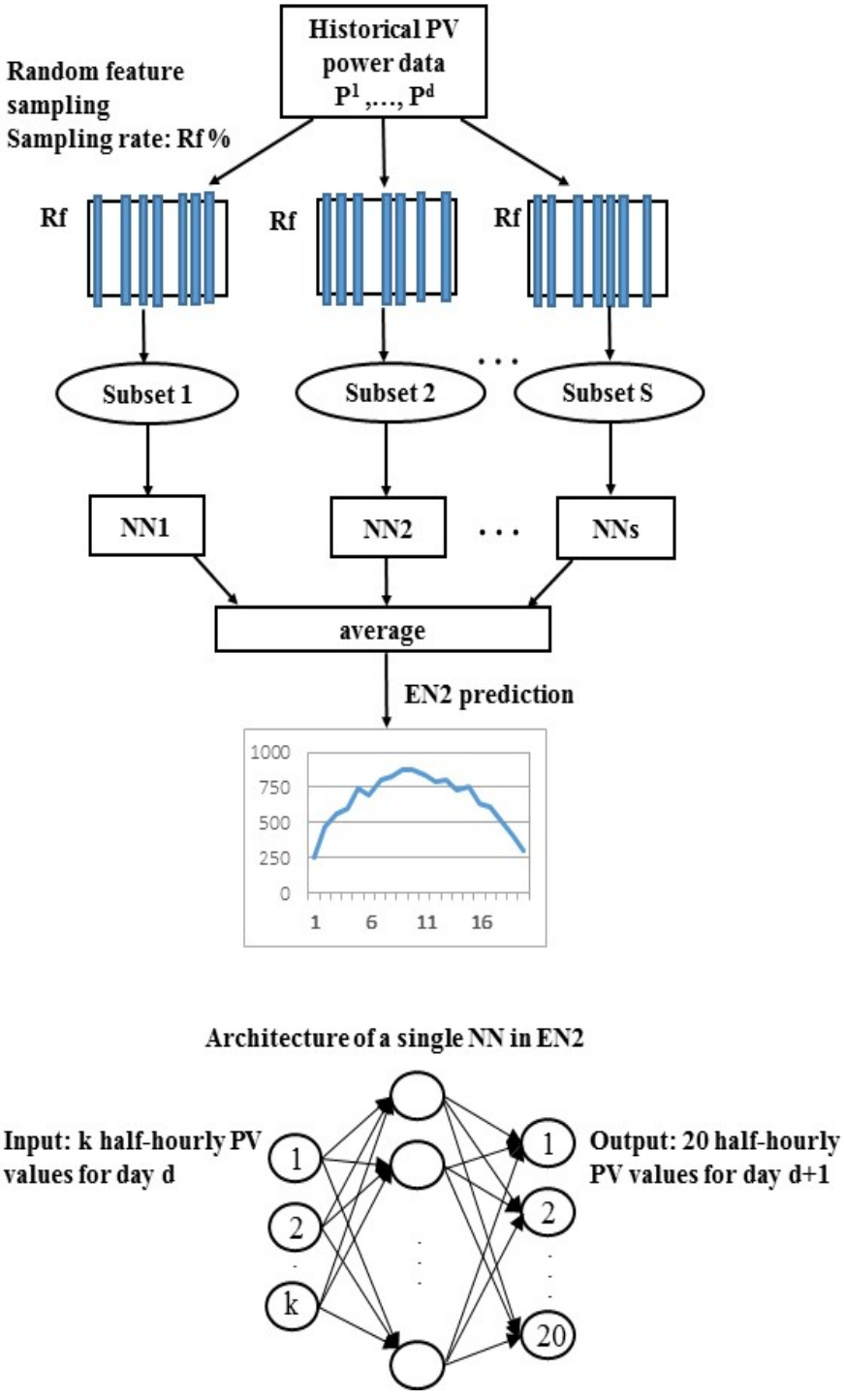


Figure 5.2: Ensemble EN2 using random feature sampling

selected best $Rf$ was 50%.

**Number of unique features:** The number of unique features for the selected sampling rate varied from 45% to 50% across the 30 ensemble members.

#### 5.1.1.3 EN3 - Random Sampling and Random Feature Selection

The third strategy is a combination of the previous two. As shown in Fig. 5.3, an ensemble EN3 with *S* NNs is built and each NN is trained using a subset of the training data as well as a subset of all features.

Firstly, a random sampling with replacement as in EN1 is applied to form training sets for each ensemble member. The best sampling rate Rs is selected using the same procedure as in EN1.

Secondly, three different feature sampling rates $Rf$ are applied to each training set to select only a subset of the features: $Rf1 = 25\%$ for the first 1/3 of the NNs, $Rf2 = 50\%$ for the second 1/3 of the NNs, and $Rf3 = 75\%$ for the last 1/3 of the NNs.

**Single ensemble member:** Each NN in EN3 has *k* input nodes corresponding to the sampled features (PV power outputs of the previous day) and 20 output nodes corresponding to all 20 half-hourly PV power output of the next day. As in EN1 and EN2, we used the same number of ensemble members: $S = 30$.

**Parameters:** The sampling rate $Rs$ is selected using the validation set as in EN1 and EN2; the best $R$s was 50%. The feature sampling rates were set as follows: $Rf1 = 25\%$, $Rf2 = 50\%$ and $Rf3 = 75\%$ as described above.

**Number of unique examples and features:** The number of unique examples for the selected sampling rate varied from 33% to 43% for the 30 ensemble members. The number of unique features was 20-30% for the first 1/3 of the ensemble members, 40-50% for the next 1/3 and 64-70% for the last 1/3.

### 5.1.2 Dynamic Ensembles Based on Previous Performance

The proposed ensembles EN1, EN2 and EN3 are static - the way they combine the predictions of the individual ensemble members do not change with time or as the time series evolves. We also explored if an adaptive combination would be beneficial.

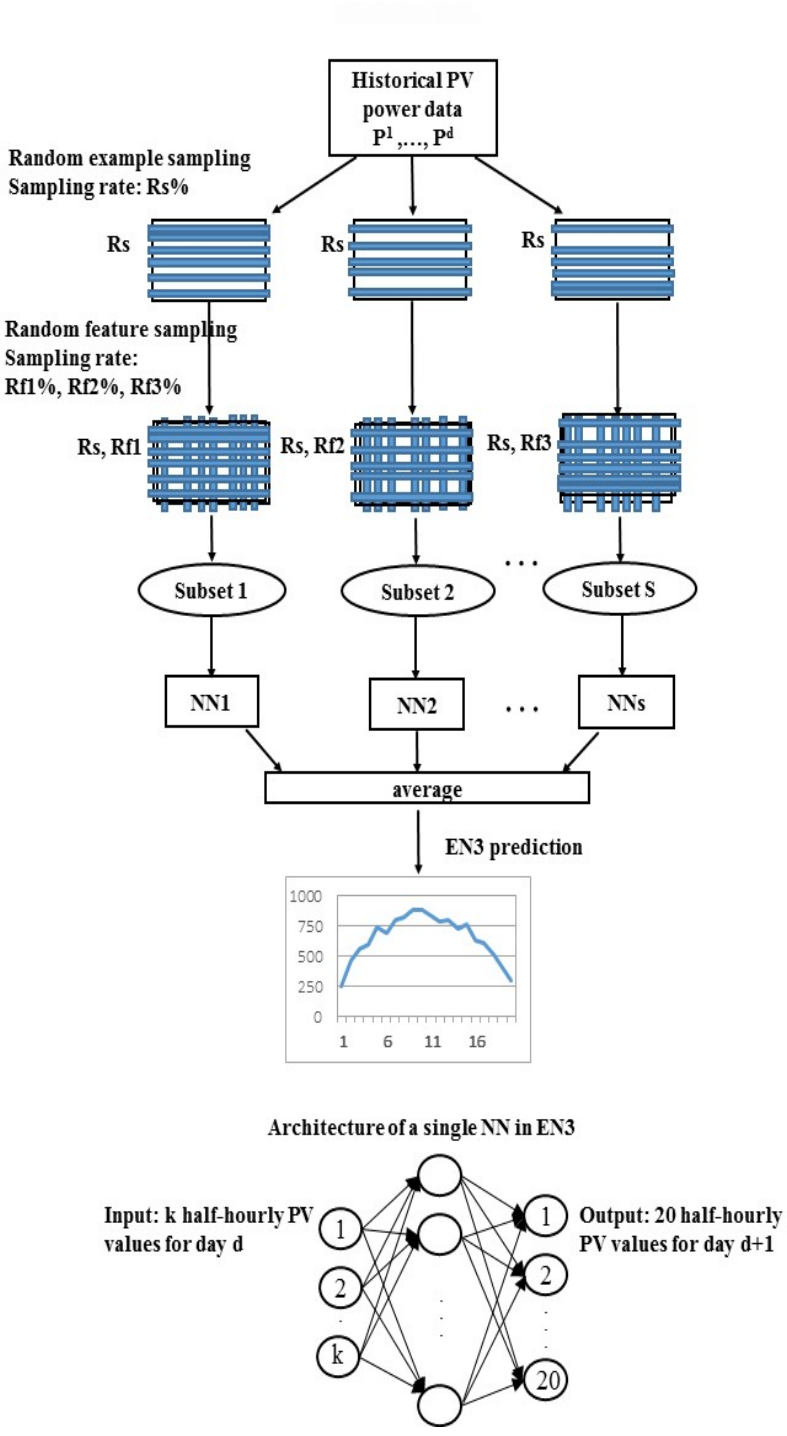


Figure 5.3: Ensemble EN3 using both random example and feature sampling

We developed four dynamic ensemble strategies that we applied to the best performing static ensemble.

The main idea is to dynamically weight the contribution of the ensemble members based on their previous performance. Thus, the predictions of the ensemble members are combined using weighted average instead of simple arithmetic average. To predict a new day *d+1*, each ensemble member $\text{NN}_i$ is assigned a weight $w_i^{d+1}$ calculated based on its most recent performance, i.e. its performance during the last *D* days. Higher weights are assigned to the more accurate ensemble members and lower weights to the less accurate ones. The weights are re-calculated for every new day that needs to be predicted. Thus, the weights of the ensemble members for day *d+1* depend on their performance during the previous *D* days which makes the ensemble dynamic.

The motivation behind using dynamic ensembles is that the different ensemble members have different areas of expertise, with their performance changing as the time series evolves over time. By tracking the error of the ensemble members on recent data, we can estimate their area of expertise and weight their contribution in the final prediction, so that the most suitable ensemble members for the new example will receive the highest weights. Thus, we adapt the ensemble to the changes in the time series and the expertise of the ensemble members.

In particular, we investigated two main strategies for calculating the weights of the ensemble members: linear transformation and non-linear transformation using the softmax function:

1. **Linear transformation for calculating weights.** The weight of the ensemble member $\text{NN}_i$ for predicting day *d+1* is calculated as:

$$w_i^{d+1} = \frac{1 - e_i^{norm}}{\sum_{j=1}^{S}(1 - e_j^{norm})}$$

where $e_i^{norm}$ is the total error of ensemble member $\text{NN}_i$ in the last *D* days, normalised between 0 and 1, and *j* is over all *S* ensemble members.

As an error $e$ we used the MAE, summed over the previous *D* days. The number of previous days *D* was set to 7, the length of a week. The subtraction of the error from 1 is necessary as higher errors should be associated with lower weights and vice versa. The denominator ensures that the weights of all ensemble members sum to 1.

2. **Non-linear transformation for calculating weights.** The weight of the ensemble member $\text{NN}_i$ for predicting day *d+1* is calculated as a softmax function of the negative of its error $e_i$:

$$w_i^{d+1} = \frac{exp(-e_i)}{\sum_{j=1}^{S} exp(-e_j)}$$

where $e_i$ is the total error of ensemble member $\text{NN}_i$ in the last *D* days, *j* is over all *S* ensemble members and *exp* denotes the exponential function.

As in the linear transformation above, we used the MAE as $e$ and *D* was set to 7 days. By taking the negative of the error, higher errors will be associated with lower weights and vice versa. The denominator again ensures that the weights of all ensemble members sum to 1.

In summary, by using the softmax function, the weight of an ensemble member decreases exponentially as its error increases. In contrast, by using the linear function, the weight of an ensemble member decreases linearly as its error increases. In addition, the least accurate ensemble member will be assigned a weight of 0, and thus, effectively will be dropped out from the prediction.

The final prediction of the dynamic ensemble is calculated by the weighted average of the predictions of the individual ensemble members:

$$\hat{P}^{d+1} = \sum_{j=1}^{S} \hat{P}_j^{d+1} \cdot w_j^{d+1}$$

We also considered combining the predictions of the *K* best ensemble members, based on their error in the last *D* days, instead of combining all ensemble members. This led to four different dynamic ensemble strategies - linear vs non-linear weight calculation and combining all vs combining the best *K* ensemble members.

### 5.1.3 Case Study

In this section, we describe the case study we conducted to evaluate the performance of the proposed static and dynamic ensembles. We used Australia data for two whole years, from 1 January 2015 to 31 December 2016. We compare the performance of

the proposed static and dynamic ensembles with a single NN, SVM, *k*-NN and a persistence model as baseline, as well as with the traditional ensembles bagging, boosting and random forests.

#### 5.1.3.1 Experimental Setup

As before, we consider the task of predicting the half-hourly PV power outputs for the next day at 30-min intervals. The prediction is based on using historical PV data only. Thus, it doesn't depend on weather information, which is an advantage as meteorological measurements for previous days and reliable weather forecasts for future days, for the location of the PV panels are not always available, while previous PV data is readily available.

More specifically, given a time series of historical PV power outputs up to the day *d*: $[p^1, p^2, p^3, ..., p^d]$, where $p^i = [p^i_1, p^i_2, p^i_3, ..., p^i_{20}]$ is a vector of 20 half- hourly power outputs for the day *i*, our goal is to forecast $PV^{d+1}$, the half-hourly power output for the next day *d+1*.

As in the previous chapters, we use again MAE and RMSE as evaluation measures.

**A. Data Sources and Preprocessing**

We collected and used PV data for two complete years: 2015 and 2016. We applied similar preprocessing methods as in the previous case studues to replace the missing values and normalize the data. The proportion of missing values of this dataset is 0.02%. The whole dataset contains $14,620 (= (365 + 366) \times 20)$ data points in total.

**B. Training, Validation and Testing Sets**

We divide the PV power data into three subsets:

1. Training - 70% of the 2015 data; it is used for model training
2. Validation - the remaining 30% of the 2015 data; it is used for parameter selection
3. Testing - the 2016 data is used to evaluate the accuracy of all models

#### 5.1.3.2 Methods for Comparison

We compare the proposed ensembles with two groups of methods: (i) single forecasting methods: a single NN, SVR, *k*-NN and a persistence model used as a baseline, and (ii) traditional ensemble methods: bagging, boosting and random forest.

**A. Single Forecasting Methods**

We build the same single forecasting models as we did for the case studies in Chapter 3 and 4. In short, we have four models built:

**NN model.** We build a NN model with one hidden layer of *m* nodes, where *m* was set to the average of the input and output nodes. It takes as an input the 20 half-hourly PV power data of the previous day *d-1* and predicts the 20 half-hourly PV data for day d.

**SVR model.** The SVR model is similar to the NN model, except that we train 20 SVRs, each predicting one of the 20 half-hourly value for the next day *d+1*. All SVRs take as an input the 20 half-hourly PV values of the previous day *d*.

***k*-NN model.** To forecast the PV power data of day *d+1*, *k*-NN firstly finds the *k* nearest neighbors of day *d* - these are the days from the training set with most similar PV power profile using the Euclidean distance. To compute the predicted PV power output for day *d+1*, it then finds the days immediately following the neighbors and averages their PV power.

**Persistence model.** As a baseline, we developed a persistence model which uses the PV power output of day *d* as the forecast for day *d+1*.

**B. Ensembles**

The second group includes classic ensemble methods: bagging, boosting and random forests, which are very successful tree-based ensembles. When used for forecasting, these methods combine regression trees.

**Bagging (Bagg)** generates diversity by altering the training data for each ensemble member. It creates a bootstrap sample with size *n*, where *n* is the number of training examples, for each regression tree, and then combines the predictions of the individual trees by taking the average.

**Boosting (Boost)** produces a series of prediction models iteratively, where each

Table 5.1: Accuracy of all static methods

| Method | MAE (kW) | RMSE (kW) |
|---|---|---|
| EN1 | 110.78 | 145.75 |
| EN2 | 111.31 | 145.34 |
| EN3 | 102.50 | 134.25 |
| NN | 116.64 | 154.16 |
| SVR | 121.58 | 158.63 |
| k-NN | 127.64 | 166.15 |
| Bagg | 109.87 | 146.40 |
| Boost | 118.08 | 158.80 |
| RF | 110.29 | 146.25 |
| P | 124.80 | 184.29 |

new prediction model focuses on the examples that were misclassified by the previous model. It uses a weighed vote based on previous performance to combine the predictions of the individual trees.

**Random Forest (RF)** uses two strategies to generate diversity in the trees that are combined: bagging and random feature selection. Specifically, it firstly uses bagging to generate bootstrap samples and then grows a regression tree for each sample. However, when selecting the best feature as a tree node at each step, it only considers $k$ randomly selected features, from all features available at the node. The individual tree predictions are combined by taking the average. RF can be seen as a bagging ensemble of random trees.

For consistency with EN1, EN2 and EN3, the number of trees in Bagg, Boost and RF was set to 30. As regression trees cannot predict all 20 values for the next day simultaneously, a separate ensemble is created for each half-hourly value, as in the SVR model. Thus, we create 20 ensembles of each type.

#### 5.1.3.3 Results and Discussion

**A. Static Ensembles**

Table 5.1 shows the accuracy results of all static forecasting methods and Fig. 5.4 presents graphically the MAE results in sorted order for visual comparison. The RMSE results follow the same trend.

The main results can be summarized as follows:

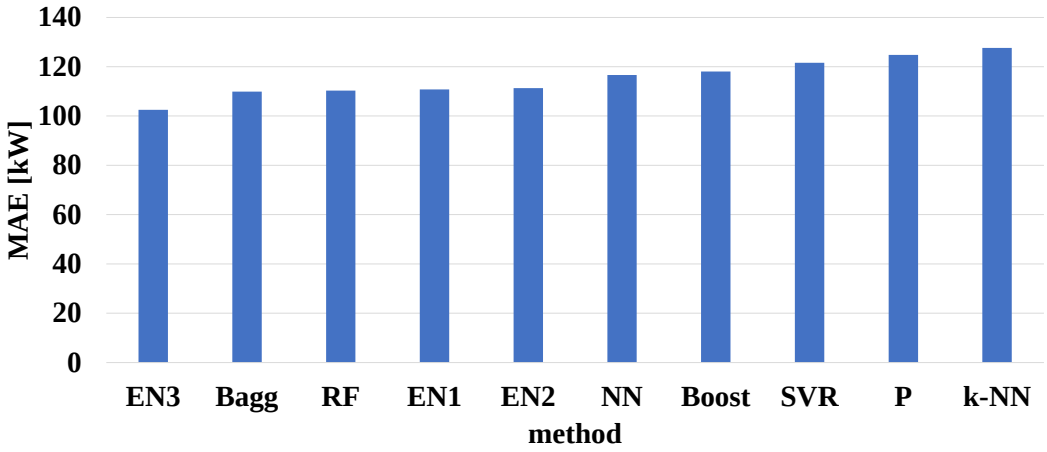


Figure 5.4: Comparison of forecasting methods (MAE)

- EN3 is the most accurate model, followed by Bagg and RF. EN3 has an additional advantage - it directly predicts the 20 half-hourly values for the next day, while Bagg and Boost required to build a separate ensemble for each value.
- EN1 and EN2 outperform the single NN which shows that the diversity brought by the random sampling (EN1) and random feature selection (EN2) was beneficial for improving the accuracy.
- EN1 and EN2 perform similarly, which indicates that the two diversity strategy had similar benefits.
- EN3 is more accurate than EN1 and EN2. This shows that combining both random sampling and random feature selection can further improve the performance of ensembles which utilize only one of these strategies.
- All ensemble methods (both the proposed and classical) are better than all single models.
- From the single prediction models, the best accuracy is achieved by NN and SVM, followed by $k$-NN and finally the persistence model. NN has an advantage over SVR as only one NN model is trained which predicts all values for the next day simultaneously.
- All forecasting models except $k$-NN outperform the persistence baseline model.

Table 5.2: EN3 dynamic ensembles - summary

| | Calculating weights | | Combining ensemble members | |
|---|---|---|---|---|
| **Method** | **Linear** | **Non-linear** | **All** | **Only best K** |
| EN3-lin | √ | | √ | |
| EN3-sofmax | | √ | √ | |
| EN3-bestK-lin | √ | | | √ |
| EN3-bestK-softmax | | √ | | √ |

- All differences in accuracy between EN1, EN2, EN3 and the other prediction models are statistically significant at $p \leq 0.05$, except the difference between EN1 and EN2.

**B. Dynamic Ensembles**

We applied the four dynamic ensemble strategies described in Section 5.1.2 to the best performing static ensemble EN3. These strategies are summarized in Table 5.2.

For the versions using the best *K* ensemble members, *K* was set to 10, 15 and 20, which is 30%, 50% and 67% of all ensemble members. The best results were achieved for $K = 15$ for EN3-bestK-lin and $K = 10$ for EN3-bestK-softmax.

Table 5.3 shows the accuracy results of the four dynamic ensembles. Fig. 5.5 presents graphically these results in sorted order and compares them with the accuracy of the static ensemble.

A pair-wise comparison for statistical significance of the differences in accuracy was conducted using the t-test. The results showed that all pair differences between the EN3 dynamic and static ensembles are statistically significant at $p \leq 0.05$ except the difference between two of the dynamic versions: EN3-lin and EN3-bestK-lin.

We can summarize the results as follows:

- All dynamic versions are more accurate than the static ensemble, and the differences are statistically significant.

- The most accurate version is EN3-bestK-lin which uses linear weight calculation and combines only the best *K* ensemble members. It is closely followed by EN3-bestK-softmax which also combines the best *K* ensemble members but uses a non-linear function to calculate the weights.

Table 5.3: Accuracy of EN3 dynamic ensembles

| Method | MAE (kW) | RMSE (kW) |
|---|---|---|
| EN3-lin | 101.64 | 130.79 |
| EN3-sofmax | 102.40 | 131.36 |
| EN3-bestK-lin | 100.46 | 130.61 |
| EN3-bestK-softmax | 100.69 | 130.57 |

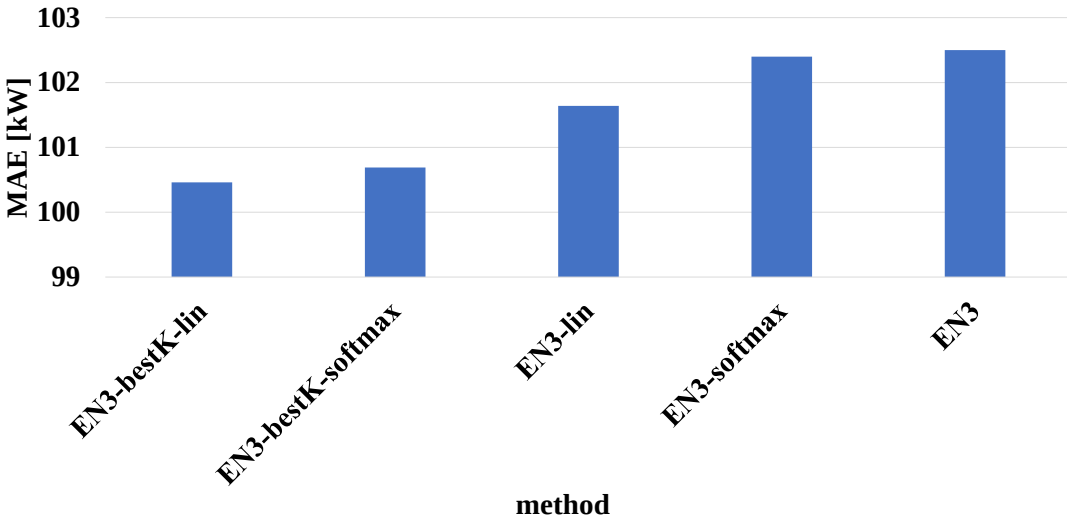


Figure 5.5: Comparison of EN3 dynamic ensembles (MAE)

- The dynamic ensemble versions using linear calculation of the weights outperformed the versions using the nonlinear softmax calculation of the weights.
- The versions combining only the best $K$ ensemble members outperformed the versions combining all ensemble methods, for both linear and nonlinear weights.

#### 5.1.3.4 Conclusion

In this case study, we considered the task of simultaneously forecasting the PV power output for the next day at half-hourly intervals, using only previous PV power data. We evaluated the proposed ensembles using Australian PV data for two years, and compared their performance with classical and state-of-the-art ensembles such as bagging, boosting and random forests, and also with four single prediction models (NN, SVM, $k$-NN and persistence).

Our results showed that all three static NN ensembles were beneficial they were more accurate than a single NN and all other single and classical ensemble methods used for comparison, and the differences were statistically significant. Ensemble EN3 which combines both random example sampling and random feature selection was the most effective ensemble. All four dynamic versions of EN3 resulted in further improvements in accuracy, which were also statistically significant. The most accurate dynamic ensemble (EN3-bestKlin) uses linear transformation to calculate the contributing weights of the ensemble members and combines only the best $K$ ensemble members. It achieved MAE = 101.64 kW and RMSE = 130.79 kW.

Hence, we conclude that EN3 and its dynamic versions are promising methods for solar power forecasting. This case study especially highlights the potential of dynamic ensembles for accurate solar power forecasting.

## 5.2 Dynamic Ensembles Based on Predicted Future Performance

The dynamic ensemble described in the previous section tracks the error of the ensemble members on previous data and converts this error into weights, used in the weighed combination of predictions for the new day. Here we investigate a different approach that is based on predicting the error of the ensemble members for the new day and converting this error into weights. The prediction is done using meta-learners, one for each ensemble member. The idea of using meta-learners was first proposed in [101]. We adapt this idea by developing meta-learners for our sampling-based ensemble members, investigating different weight calculation and different ensemble member combination strategies.

### 5.2.1 Methodology

There are three main steps in creating the dynamic meta-learning ensemble EN-meta as shown in Fig. 5.6:

1. Training ensemble members

2. Training meta-learners

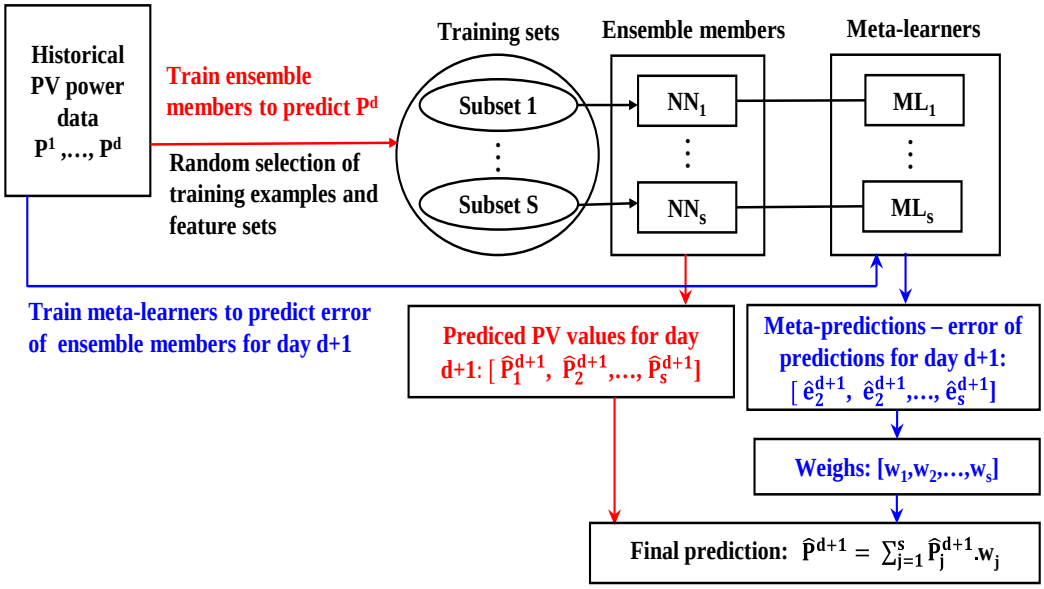


Figure 5.6: Structure of EN-meta

3. Calculating the weights of the ensemble members for the prediction of the new example.

We train the ensemble members to predict the PV power for the next day and their corresponding meta-learners (one for each ensemble member) to predict the error of this prediction. Thus, each meta-learner learns to predict how accurate the prediction of the ensemble member will be for the new day based on the characteristics of the day. The predicted errors are converted into weights (higher weights for the more accurate ensemble members and lower for the less accurate) and the final prediction is given by the weighted average of the individual predictions.

#### 5.2.1.1 Training Ensemble Members

We apply the meta-learning strategy to our best static ensemble model (EN3), which uses both random example sampling and random feature sampling to generate diversity. This is shown in Fig. 5.7.

**Random example sampling:** We create $S$ bootstrap samples, one for each NN, using random sampling with replacement and a pre-defined example sampling rate $Rs$. Each sample contains only $Rs\%$ of the $d$ examples for the first year, which is the whole data used for training and validation. These examples are then randomly

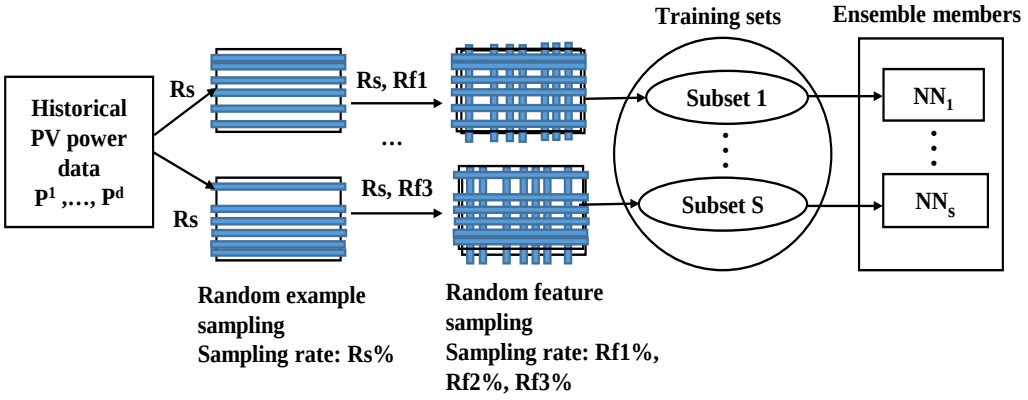


Figure 5.7: Training ensemble members

divided into training set (70%, used for training of the NN) and validation set (30%, used for selecting the NN parameters). Thus, the training set for a single NN will contain a smaller number of examples than the original training set and will have the same number of features. The best $Rs$ was selected by experimenting with different values and evaluating the performance on the validation set (best $Rs$ = 25%).

**Random feature sampling:** The *S* training sets from the previous step are filtered by retaining only some of their features and discarding the rest. This is done by using feature sampling with replacement with a pre-defined sampling rate $Rf$. We split the *S* training sets into three parts and applied $Rf1$=25%, $Rf2$=50% and $Rf3$=75% for each third.

A single ensemble member is a NN with $f$ input neurons ($f < 20$), corresponding to the sampled features (PV power of the previous day), and 20 outputs, corresponding to all 20 PV values of the next day. It had one hidden layer where the number of neurons was set to the average of the input and output neurons, and was trained using the Levenberg-Marquardt version of backpropagation algorithm. In this study, we still combined $S$=30 NNs.

#### 5.2.1.2 Training Meta-learners

Every ensemble member $NN_i$ has an associated meta-learner $ML_i$, which is trained to predict the error of $NN_i$ for the new day. Thus, $ML_i$, takes as an input the PV data for day *d* and predicts the forecasting error of $NN_i$ for day *d+1*. The error is then converted into a weight for $NN_i$ and used in the weighted average vote combining the

predictions of all ensemble members.

The motivation behind using dynamic ensembles is that the different ensemble members have different areas of expertise, with their performance changing as the time series evolves over time. We can learn to predict the error of an ensemble member for the next day based on its prior performance. Then we can use these predicted errors to weight the contributions of the ensemble members in the final prediction, so that ensemble members that are predicted to be more accurate are given higher weights. In this way we match the expertise of the ensemble members with the characteristics of the new day and adapt the ensemble to the changes in the time series.

For this study, we implemented and compared two sets of meta-learners: NN and $k$-NN. Both sets contain $S$ meta-learners, one for each ensemble member. Each meta-learner was trained to predict the Mean Absolute Error (MAE) of its corresponding ensemble member for the next day.

**NN meta-learners.** To train a NN meta-learner $\text{ML}_i$ for ensemble member $\text{NN}_i$, we firstly need to create the training data for it, and in particular to obtain the target output. Using the trained ensemble member $\text{NN}_i$, we obtain its prediction for all examples from the training set; the input is $P^d$ the PV power vector of the previous day $d$ but containing only the $f$ sampled features, and the output is $P^{d+1}$, the PV power vector for the next day *d+1* containing all 20 values. We then calculate $\text{MAE}^{d+1}$, the error for day *d+1*. A training example for $\text{ML}_i$ will have the form: [$P^d$, $\text{MAE}^{d+1}$], where $P^d$ is the input vector (containing the same $f$ features as $\text{NN}_i$) and $\text{MAE}^{d+1}$ is the target output. Thus, the meta-learner NN has $f$ input and 1 output neurons. We again used 1 hidden layer and the number of neurons was set to the average of the input and output neurons.

***k*-NN meta-learners.** In contrast to the NN meta-learners, there is no need to pre-train the $k$-NN meta-learners as the computation is delayed till the arrival of the new day. Specifically, to build a $k$-NN meta-learner for ensemble member $\text{NN}_i$ for the new day *d+1*, the PV data of the previous day $d$ is collected and processed by selecting the same subset of features $f$ as for $\text{NN}_i$. Then, the training set is searched to find the $k$ most similar days to day $d$ in terms of the f features. The errors (MAE) of the $\text{NN}_i$ for the days immediately following the neighbors are calculated and averaged to calculate $\text{MAE}^{d+1}$, the predicted error of ensemble member $\text{NN}_i$ for day *d+1*. To select the value of $k$, we experimented with $k$ from 5 to 15, evaluating the performance on the validation set; the best $k$ was 10 and it was used in this study.

#### 5.2.1.3 Weight Calculation and Combination Methods

Similarly, we conduct the same combination strategies, both linear and non-linear versions, to combine the predicted results generated by different ensemble members (As described in Section 5.1.2).

We also considered combining the predictions of only the *M* best ensemble members, based on their predicted error, instead of combining all ensemble members. To select the best M, we experimented with *M*=1/3, 1/2 and 2/3 of all ensemble members (30 in our study), evaluating the performance on the validation set. Hence there are four different strategies for combining the individual predictions linear vs non-linear weight calculation and combining all vs combining only the best *M* ensemble members.

### 5.2.2 Case Study

#### 5.2.2.1 Experimental Setup

To evaluate the proposed meta-learner based method, we conduct a case study using the same PV data as in Section 5.1.3, considering the same task: predicting the half-hourly PV power outputs for the next day at 30-min intervals. More specifically, given a time series of historical PV power outputs up to the day *d*: $[p^1, p^2, p^3, ..., p^d]$, where $p^i = [p^i_1, p^i_2, p^i_3, ..., p^i_{20}]$ is a vector of 20 half- hourly power outputs for the day *i*, our goal is to forecast $PV^{d+1}$, the half-hourly power output for the next day *d+1*.

For consistent comparison, we also use MAE and RMSE as evaluation measures.

The methods used for comparison can be categorized into three groups:

1. **Single Models.** This is the same with the single models in Section 5.1.3. We use the same structure and parameters.

2. **Classic Ensembles.** Similarly, we also implement the regression tree-based ensembles **Bagging (Bagg)**, **Boosting (Boost)** and **Random Forest (RF)**. For consistency with the proposed ensemble, the number of trees in Bagg, Boost and RF was set to 30. As regression trees cannot predict all 20 values for the next day simultaneously, a separate ensemble is created for each half-hourly value, as in the SVR model. Thus, we create 20 ensembles of each type.

3. **Static and Dynamic Ensembles Without Meta-learners**. To assess the contribution of the meta-learning component, we also compare the performance of the dynamic meta-learning ensemble EN-meta with two versions of this ensemble without meta-learning: static and dynamic, proposed in Section 5.1.

   The static ensemble is EN-meta without the meta-learning component and using the average of the individual predictions to form the final prediction. We refer to this ensemble as **EN-static**.

   The dynamic ensemble is an extension of **EN-static**: it uses weighed average for combining the individual predictions. The weighs of the ensemble members are calculated based on their previous performance (MAE error) in the last 7 days. The errors of the ensemble members are converted into weights using the same methods(linear and nonlinear; combining all or only the best *M* ensemble members).

   We evaluated the different versions using validation set testing; the best result were achieved for the version using a linear transformation and combining the best *M* ensemble members with *M*=15; we refer to this ensemble as **EN-dynamic**.

#### 5.2.2.2 Results and Discussion

**A. Performance of EN-meta**

Table 5.4 shows the accuracy results of EN-meta for the two different types of meta-learners and four weight calculation methods. The graph in Fig. 5.8 presents the MAE results in sorted order for visual comparison. We also conducted a pair-wise comparison for statistical significance of the differences in accuracy using the Wilcoxon ranksum test with $p \leq 0.05$. The results can be summarized as follows:

- **Overall performance:** The most accurate version of EN-meta is kNN-bestM-lin, which uses kNN meta-leaners, combines the predictions of only the best M ensemble members and uses linear transformation to convert the predicted errors into weights. It is followed by kNN-bestM-softmax, which differs only in the weight calculation function softmax instead of linear, and then by NN-bestM-softmax.

Table 5.4: Accuracy of EN-meta versions

| EN-meta | MAE (kW) | RMSE (kW) |
|---|---|---|
| **with NN meta-learners** | | |
| NN-lin | 88.40 | 115.35 |
| NN-softmax | 89.63 | 116.13 |
| NN-bestM-lin | 87.75 | 115.55 |
| NN-bestM-softmax | 87.68 | 115.29 |
| **with kNN meta-learners** | | |
| kNN-lin | 88.10 | 114.89 |
| kNN-softmax | 89.61 | 116.11 |
| kNN-bestM-lin | 86.77 | 114.57 |
| kNN-bestM-softmax | 87.34 | 115 |

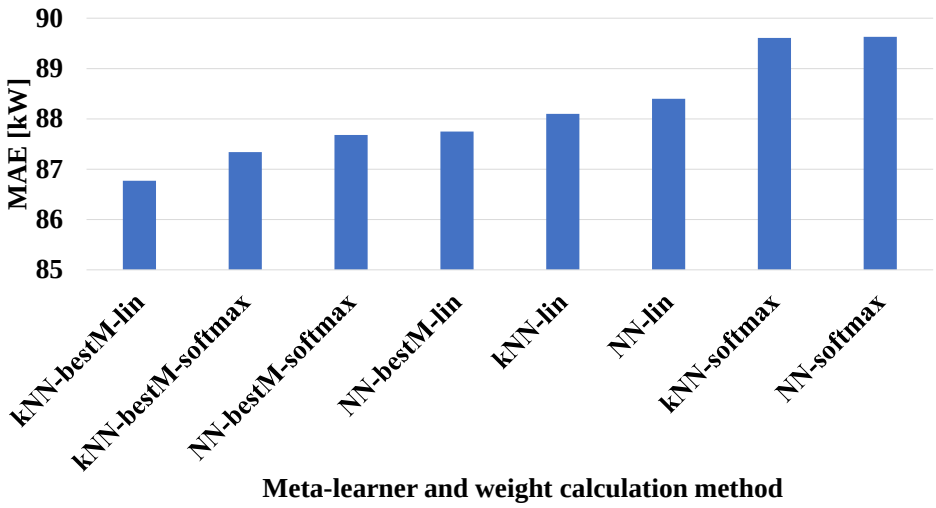


Figure 5.8: MAE comparison

- The pair-wise differences in accuracy between these three best models are not statistically significant but all other differences between the best model (kNN-bestM-lin) and the other models are statistically significant.
- **All vs best *M* ensemble members:** The EN-meta versions combining only the predictions of the best *M* ensemble members are more accurate than their corresponding versions which combine the predictions of all ensemble members and these differences are statistically significant.

Table 5.5: Accuracy of all models

| Method | MAE(kW) | RMSE(kW) |
|---|---|---|
| EN-meta | 86.77 | 114.57 |
| **Single models** | | |
| NN | 116.64 | 154.16 |
| SVR | 121.58 | 158.63 |
| kNN | 127.64 | 166.15 |
| Persistence | 124.80 | 184.29 |
| **Classic ensembles** | | |
| Bagg | 109.87 | 146.40 |
| Boost | 118.08 | 158.80 |
| RF | 110.29 | 146.25 |
| **EN-meta without meta learners** | | |
| EN-static | 102.50 | 134.25 |
| EN-dynamic | 100.46 | 130.61 |

- **Linear vs softmax weight calculation:** The EN-meta versions using linear weight calculations outperformed their corresponding versions using the softmax weight calculation in 3/4 cases but the differences are not statistically significant.

- **NN vs kNN meta-learners:** The EN-meta versions using kNN meta-learners were more accurate than their corresponding versions using NN meta-learners in all 4 cases but these differences are not statistically significant.

Based on these results we selected the best version (EN-meta-kNN-bestM-lin) for further investigation. We will refer to it as **EN-meta**.

**B. Comparison With Other Methods**

Table 5.5 compares the accuracy of EN-meta with the single models, classical ensembles and the two EN versions without meta-learners (static and dynamic). Fig. 5.9 graphically presents the MAE results in sorted order for visual comparison. The main results can be summarized as follows:

- The proposed EN-meta is the most accurate method. It considerable outperformed all other methods and all differences are statistically significant (Wilcoxon

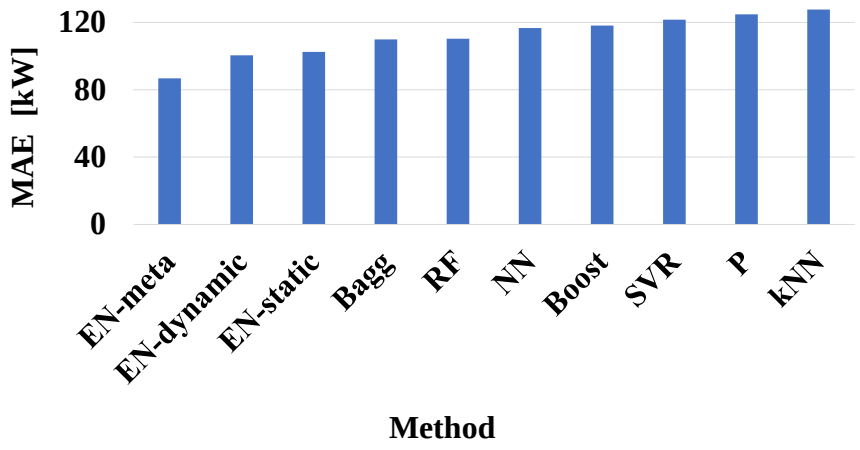


Figure 5.9: MAE comparison

sun-rank test, $p \leq 0.05$).

- The next best performing methods are EN-dynamic and EN-static, the EN-meta versions without meta-learners. This shows that the use of meta-learners was beneficial.

- EN-dynamic was more accurate than EN-static and the difference was statistically significant. This shows the advantage of tracking the error of the ensemble members on recent data and correspondingly weighting their contribution in the weighed vote.

- By comparing the two dynamic ensembles, EN-meta and EN-dynamic, we can see that the use of meta-learners and the more proactive approach of EN-meta for assessing the ensemble members - based on predicted error for the new day rather than error on previous days, gives better results.

- Bagg is the most accurate classic ensemble, followed by RF and Boost. All classical ensemble models outperform the single models, except for Boost which performs slightly worse than the single NN.

- From the single prediction models, NN was the best, followed by SVR, P and *k*-NN. All forecasting models except *k*-NN outperform the baseline P model.

#### 5.2.2.3 Conclusion

In this case study, we evaluated EN-meta - a meta-learning ensemble of NNs, where the ensemble members are generated using sampling techniques. The key idea is to pair each ensemble member with a meta-learner and train the meta-learner to predict the error for the next day of its corresponding ensemble member. The errors are then converted into weights and the final prediction is formed using weighed average of the individual predictions. EN-meta is a dynamic ensemble as the combination of predictions is adapted to the characteristics of the new day based on the expected error.

We investigated four strategies for converting the predicted error into weights and two types of meta-learners ($k$-NN and NN). We also compared the performance of EN-meta with three state-of-the-art ensembles (bagging, boosting and random forest), four single models (NN, SVM, $k$-NN and persistence) and two versions of EN-meta without meta-learners. The evaluation was conducted using Australian data for two years. Our results showed that EN-meta was the most accurate model, considerably and statistically significantly outperforming all other methods. The kNN meta-learners were slightly more accurate than NN and the most effective strategy was combining only the best $M$ ensemble members and using linear transformation to calculate the weights. The use of meta-learners to directly predict the error for the new day instead of estimating it based on the error on the previous days was beneficial.

## 5.3 Summary

This chapter proposed several methods for constructing static and dynamic ensembles of NNs for solar power forecasting, evaluated their performance on Australian data and compared them with classical ensembles (bagging, boosting and random forest) and single prediction models.

Firstly, in Section 5.1 we proposed three strategies for creating static ensembles based on random example and feature sampling, and four strategies for creating dynamic ensembles, by adaptively weighting the contribution of the ensemble members in the final prediction based on their recent performance. Our evaluation using Australian PV data for two years showed that all static ensembles were beneficial, outperforming the single models and classical ensembles used for comparison, with EN3 being the most accurate ensemble. All four dynamic versions of EN3 further improved

the accuracy of EN3.

Next, in Section 5.2 we investigated another type of dynamic ensemble (EN-meta) which uses predicted performance for the new day instead of actual performance on past days to calculate the weights of the ensemble members for the combined prediction. Specifically, instead of evaluating the error of an ensemble member on the most recent previous days and converting it into weight, EN-meta builds a meta-learner for each ensemble member, which forecasts the error for the new day, and this predicted error is converted into weight. Our case study showed that the dynamic ensemble using meta-learners was more accurate than the dynamic ensemble without meta-learning.

Therefore, based on the results, we can conclude that the proposed ensembles are promising methods for solar power forecasting tasks. Our research in particular highlights the potential of dynamic ensembles to generate accurate forecasts.

# Chapter 6

# Conclusions and Future Directions

## 6.1 Conclusions

In this thesis, we considered the task of using machine learning methods for predicting the PV power output for the next day. More specifically, we focused on directly and simultaneously predicting the PV power for the next day at 30-min intervals, using data from various data sources: previous PV data, previous weather data and weather forecasts.

In Chapter 3, we investigated two state-of-the-art instance-based methods for time series prediction tasks, *k*-NN and PSF, and discussed their limitations. We proposed two extensions, addressing these limitations, namely DW*k*NN and the extended PSF. DW*k*DD extends *k*-NN by considering the importance of the three data sources (historical PV power data, historical weather data and weather forecasts) and assigning weights to them instead of treating them equally in the final prediction. The best weights for the data sources are learnt from previous data. Our results using Australian data showed that DW*k*NN was more accurate than the standard *k*-NN and other state-of-the-art methods and baselines used for comparison. In the same chapter, we also discussed the limitation of the standard PSF algorithm, namely its inability to deal with time series data from more than one source, and proposed two extensions (PSF1 and PSF2) to address this limitation. We evaluated the PSF extensions using Australian data for two years showing that the proposed extensions were beneficial and were more accurate than the standard PSF algorithm. We concluded that both the DW*k*NN and

PSF extensions are promising methods for solar power forecasting, allowing to integrate data from multiple sources.

In Chapter 4, we focused on clustering-based methods, which take advantage of the similarity between the PV power profiles of days with similar weather profiles. We proposed two groups of methods: direct clustering-based and and pair patters clustering-based. The main idea of the direct method is to partition the days based on their weather characteristics and then build a separate prediction model for each cluster using the PV solar data. The results of our case study showed that the direct clustering-based method is more accurate than the general, single prediction model for all types of days. The second method, Weather Pair Patterns (WPP), extends the direct clustering-based method by utilizing the relationship between consecutive days. This method builds a separate prediction model for each type of cluster transition between two consecutive days. Our results showed that WPP outperformed the direct clustering-based method. In summary, the proposed methods which use clustering to partition the days into groups with similar weather characteristics and then build a separate prediction model for each group, were more beneficial for solar power forecasting than the methods using one prediction model for all types of days.

In Chapter 5, we investigated the potential of static and dynamic ensemble to improve the performance of single forecasting models. In particular, we created ensembles combining NNs and using only PV data as the weather information may not be available for the location of the PV plant. Firstly, in Section 5.1 we proposed a number of strategies for constructing static and dynamic ensembles of NNs. Specifically, we proposed three strategies for creating static ensembles based on random example and feature sampling, and four strategies for creating dynamic ensembles by adaptively weighting the contribution of the ensemble members in the final prediction based on their recent performance. Our evaluation using Australian PV data for two years showed that all static ensembles of NNs were beneficial, outperforming the single NN and all other single and classical ensemble methods used for comparison. We further investigated the most effective static ensemble (EN3); the results showed that all dynamic versions of EN3 further improved the accuracy, demonstrating the promise of dynamic ensembles.

In Section 5.2, we investigated another type of dynamic ensemble (EN-meta) which utilized meta-learners to improve the performance. It uses predicted performance for the new day instead of actual performance on previous days to calculate the weights

of the ensemble members for the combined prediction. More specifically, instead of evaluating the error of an ensemble member on the most recent previous days and converting it into weights, EN-meta trains a meta-learner for each ensemble member, which forecasts the error for the new day and converts this predicted error into weights. Our results showed that the dynamic ensembles using meta-learners were more accurate than the dynamic ensemble without meta-learning. Based on the results in this chapter, we can conclude that the proposed ensembles of NNs are promising methods for solar power forecasting, especially the dynamic ensembles.

In conclusion, this thesis addresses the task of directly and simultaneously forecasting the PV power output for the next day at 30-min intervals using machine learning methods. As a main contribution of this thesis, we have developed novel instance-based, clustering-based and ensemble methods; we applied them to accomplish this task - we evaluated them on Australian data and compared them with other machine learning methods, demonstrating their advantages. We explored the use of different data sources (historical PV data, historical weather data and weather forecasts), taking into account that sometimes only the historical PV data may be available for the location of the PV plant. The methods proposed in this thesis can be employed by engineers at PV plants for practical purposes or adopted by researchers for solar, energy-related or other time series forecasting tasks.

## 6.2 Recommendations

In this section we provide some general recommendations about the applicability of the three groups of methods that we have developed: instance-based, clustering-based and ensembles.

We note that direct comparison between the three groups is not possible as they use different data sources - the ensemble methods use only PV power data while the other two groups use PV power, weather and weather forecast data but differ in the weather and weather forecast features used. In addition, the clustering-based methods (which were developed first) were evaluated on data from 2013 and 2014, while the other two groups of methods were evaluated on newer data - 2015 and 2016, due to unavailability of the required weather data for the older years. Finally, the training/validation/test split for the clustering-based methods is slightly different than that for the other two

groups. Each group of methods is comprehensively evaluated and compared with a number of other methods and baselines in the respective chapter of the thesis.

For practical applications in industry, the characteristics of each group of methods should be taken into consideration, together with the specific forecasting scenario and data availability.

The methods discussed in Chapter 3 (DW$k$NN and extended PSF) are instance-based. Instance-based methods are also called lazy learning methods as the main computation is delayed till the forecasting stage when the nearest neighbours are found. Finding neighbours in the extended PSF is faster than in DW$k$NN as sequences of cluster labels are compared instead of weather or PV vectors. The training for both methods is fast as it only includes parameter selection. Hence, this group of methods are faster to train than the clustering-based and ensemble-based methods, but slower at forecasting. They may be suitable for scenarios where frequent updates of the model are required (e.g. for online forecasting) provided the historical window for finding neighbours is appropriately chosen, given the computing resources and time constraints. Also, both DW$k$NN and the extended PSF require meteorological variables to be collected. For better performance, accurate weather forecasts are especially important. So this group of methods should be applied in scenarios where meteorological measurements are readily available, frequent updates of the model are required and the nearest neighbours can be computed sufficiently fast given the computing resources and time resources.

The methods discussed in Chapter 4 (direct clustering and weather pair patterns) are clustering-based. They also require meteorological variables and are thus applicable when weather and weather forecast variables can be easily collected. Compared to the instance-based methods, the training stage of the clustering-based models is slower but the forecasting stage is very fast. This means that the clustering-based models are more suitable for scenarios when less frequent updates are required. Another characteristic of the clustering-based methods is that they split the training data into subsets based on the cluster label or the weather pair patterns. This suggests that the clustering-based methods should be used in scenarios where there is sufficient data in each cluster partition so that accurate prediction models can be built.

The methods discussed in Chapter 5 utilize ensembles of neural networks. They use only the PV power data, and do not require the availability of meteorological data. Similarly to the clustering-based methods, the training of the ensembles is slower and

more computationally expensive than for the instance-based models, but once the ensemble is trained, the forecasting process is fast. Ensembles are also slower to train than the clustering-based methods, since each ensemble combines the predictions of multiple neural networks, which are trained separately. Hence, frequent updates of the ensembles would take more time than for the other two groups of methods. So the best scenario for applying ensemble-based methods is when no frequent updates of the model are required and no meteorological information is easily available. Our results showed that when using only PV data, without any weather information, the ensemble-based methods were the most accurate models, with the dynamic meta-learning ensemble EN-meta achieving the best result.

## 6.3 Future Directions

There are several directions for future work.

First, methods for feature selection can be investigated to select a smaller set of informative features, before the application of the prediction algorithms. This can be done for the whole day or for specific time intervals during the day. In our work we didn't investigate feature selection methods - we used the whole daily PV and weather feature vectors. The feature selection can also be used in conjunction with the DW$k$NN model from Chapter 3 which considers the importance of the data sources and learns the best weights for them.

Second, our methods can be evaluated on other datasets, with different characteristics. Our evaluation was conducted on data collected from the Australian largest roof-top PV solar plant located in Queensland. The PV power profile of this data set is relatively stable with two clear patterns: a smooth curve in summer and fluctuation in winter. Future work could apply and extend our proposed methods to other data sets with more complex PV and weather profiles. This will help to evaluate the robustness of our methods for different types of PV and weather patterns.

Third, in our ensemble methods we investigated ensembles combining NNs. Our methods can be used to combine different types of prediction models and generate a heterogeneous ensemble, e.g. an ensemble combining LR, SVR and NN. In addition, dynamic ensembles can be built for different weather types or other scenarios by analysing the performance of the individual ensemble members. More sophisticated

strategies for building dynamic ensembles can also be used.

Fourth, the potential of using deep learning approaches such as recurrent neural networks, long short-term memory networks, convolutional and temporal convolutional neural networks should also be explored. Deep learning has shown excellent results in many applications and its application for solar power forecasting is a promising direction for future work.

Finally, our methods can also be applied to other energy-related time series such as wind and electricity demand prediction.

# Bibliography


[1] Solar Power Europe [Online], "Global Market Outlook For Solar Power 2015-2019," http://resources.solarbusinesshub.com/solar-industry-reports/item/global-market-outlook-for-solar-power-2015-2019.

[2] Climate Commission [Online], "The Critical Decade: Australia's Future - Solar Energy," http://www.climatecouncil.org.au/uploads/497bcd1f058be45028e3df9d020ed561.pd.

[3] Annette Hammer, Detlev Heinemann, Elke Lorenz, and B. Lckehe, "Short-term forecasting of solar radiation: a statistical approach using satellite data," *Solar Energy*, vol. 67, no. 1, pp. 139–150, 1999.

[4] Sophie Pelland, George Galanis, and George Kallos, "Solar and photovoltaic forecasting through postprocessing of the Global Environmental Multiscale numerical weather prediction model," *Progress in Photovoltaics: Research and Applications*, vol. 21, no. 3, pp. 284–296, 2013.

[5] Steven D. Miller, Matthew A. Rogers, John M. Haynes, Manajit Sengupta, and Andrew K. Heidinger, "Short-term solar irradiance forecasting via satellite/model coupling," *Solar Energy*, vol. 168, pp. 102–117, 2018.

[6] Philippe Blanc, Jan Remund, and Loic Vallance, "6 - Short-term solar power forecasting based on satellite images," in *Renewable Energy Forecasting*, George Kariniotakis, Ed., Woodhead Publishing Series in Energy, pp. 179 – 198. Woodhead Publishing, 2017.

[7] Hugo T. C. Pedro and Carlos F. M. Coimbra, "Assessment of forecasting techniques for solar power production with no exogenous inputs," *Solar Energy*, vol. 86, no. 7, pp. 2017–2028, 2012.

[8] Huan Long, Zijun Zhang, and Yan Su, "Analysis of daily solar power prediction with data-driven approaches," *Applied Energy*, vol. 126, pp. 29–37, 2014.

[9] Zibo Dong, Dazhi Yang, Thomas Reindl, and Wilfred M. Walsh, "Short-term solar irradiance forecasting using exponential smoothing state space model," *Energy*, vol. 55, pp. 1104–1113, 2013.

[10] Dazhi Yang, Vishal Sharma, Zhen Ye, Lihong I. Lim, Lu Zhao, and Aloysius W. Aryaputera, "Forecasting of global horizontal irradiance by exponential smoothing, using decompositions," *Energy*, vol. 81, pp. 111–119, 2015.

[11] Yanting Li, Yan Su, and Lianjie Shu, "An ARMAX model for forecasting the power output of a grid connected photovoltaic system," *Renewable Energy*, vol. 66, pp. 78–89, 2014.

[12] Ercan Izgi, Ahmet Oztopal, Bihter Yerli, Mustafa K. Kaymak, and Ahmet D. Sahin, "Shortmid-term solar power prediction by using artificial neural networks," *Solar Energy*, vol. 86, no. 2, pp. 725–733, 2012.

[13] Changsong Chen, Shanxu Duan, Tao Cai, and Bangyin Liu, "Online 24-h solar power forecasting based on weather type classification using artificial neural network," *Solar Energy*, vol. 85, no. 11, pp. 2856–2870, 2011.

[14] Christophe Paoli, Cyril Voyant, Marc Muselli, and Marie-Laure Nivet, "Forecasting of preprocessed daily solar radiation time series using neural networks," *Solar Energy*, vol. 84, no. 12, pp. 2146–2160, 2010.

[15] Zhaoxuan Li, S. M. Mahbobur Rahman, Rolando Vega, and Bing Dong, "A hierarchical approach using machine learning methods in solar photovoltaic energy production forecasting," *Energies*, vol. 9, no. 1, pp. 55, 2016.

[16] Hugo T. C. Pedro and Carlos F. M. Coimbra, "Nearest-neighbor methodology for prediction of intra-hour global horizontal and direct normal irradiances," *Renewable Energy*, vol. 80, pp. 770–782, 2015.

[17] Rich H. Inman, Hugo T. C. Pedro, and Carlos F. M. Coimbra, "Solar forecasting methods for renewable energy integration," *Progress in Energy and Combustion Science*, vol. 39, no. 6, pp. 535–576, 2013.

[18] Cristina Cornaro, Marco Pierro, and Francesco Bucci, "Master optimization process based on neural networks ensemble for 24-h solar irradiance forecast," *Solar Energy*, vol. 111, pp. 297–312, 2015.

[19] Maimouna Diagne, Mathieu David, Philippe Lauret, John Boland, and Nicolas Schmutz, "Review of solar irradiance forecasting methods and a proposition for small-scale insular grids," *Renewable and Sustainable Energy Reviews*, vol. 27, pp. 65–76, 2013.

[20] Pedro A. Jimenez, Joshua P. Hacker, Jimy Dudhia, Sue E. Haupt, Jose A. Ruiz-Arias, Chris A. Gueymard, Gregory Thompson, Trude Eidhammer, and Aijun Deng, "WRF-SOLAR: description and clear-sky assessment of an augmented NWP model for solar power prediction," *Bulletin of the American Meteorological Society*, vol. 97, no. 7, pp. 1249, 2016.

[21] Claudio Monteiro, Tiago Santos, L. Fernandez-Jimenez, Ignacio 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, 2013.

[22] Elke Lorenz and Detlev Heinemann, "1.13 - Prediction of solar irradiance and photovoltaic power," in *Comprehensive Renewable Energy*, Ali Sayigh, Ed., pp. 239 – 292. Elsevier, Oxford, 2012.

[23] Aidan Tuohy, John Zack, Sue E. Haupt, Justin Sharp, Mark Ahlstrom, Skip Dise, Eric Grimit, Corinna Mohrlen, Matthias Lange, Mayte Garcia Casado, Jon Black, Melinda Marquis, and Craig Collier, "Solar forecasting: methods, challenges, and performance," *IEEE Power and Energy Magazine*, vol. 13, no. 6, pp. 50–59, 2015.

[24] Handa Yang, Ben Kurtz, Dung Nguyen, Bryan Urquhart, Chi W. Chow, Mohamed Ghonima, and Jan Kleissl, "Solar irradiance forecasting using a ground-based sky imager developed at UC San Diego," *Solar Energy*, vol. 103, pp. 502–524, 2014.

[25] Chi W. Chow, Bryan Urquhart, Matthew Lave, Anthony Dominguez, Jan Kleissl, Janet Shields, and Byron Washom, "Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed," *Solar Energy*, vol. 85, no. 11, pp. 2881–2893, 2011.

[26] Zhenzhou Peng, Dantong Yu, Dong Huang, John Heiser, Shinjae Yoo, and Paul Kalb, "3D cloud detection and tracking system for solar forecast using multiple sky imagers," *Solar Energy*, vol. 118, pp. 496–519, 2015.

[27] Ricardo Marquez, Hugo T. C. Pedro, and Carlos F. M. Coimbra, "Hybrid solar forecasting method uses satellite imaging and ground telemetry as inputs to ANNs," *Solar Energy*, vol. 92, pp. 176–188, 2013.

[28] Richard Perez, Sergey Kivalov, James Schlemmer, Karl Hemker, David Renn, and Thomas E. Hoff, "Validation of short and medium term operational solar radiation forecasts in the US," *Solar Energy*, vol. 84, no. 12, pp. 2161–2172, 2010.

[29] Luis M. Aguiar, Brais Pereira, Philippe Lauret, Felipe Daz, and Mathieu David, "Combining solar irradiance measurements, satellite-derived data and a numerical weather prediction model to improve intra-day solar forecasting," *Renewable Energy*, vol. 97, pp. 599–610, 2016.

[30] Edward B. Ssekulima, Muhammad B. Anwar, Amer Al Hinai, and Mohamed S. El Moursi, "Wind speed and solar irradiance forecasting techniques for enhanced renewable energy integration with the grid: a review," *IET Renewable Power Generation*, vol. 10, no. 7, pp. 885–989, 2016.

[31] Xwegnon G. Agoua, Robin Girard, and George Kariniotakis, "Short-term spatio-temporal forecasting of photovoltaic power production," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 2, pp. 538–546, 2018.

[32] Dazhi Yang, Panida Jirutitijaroen, and Wilfred M. Walsh, "Hourly solar irradiance time series forecasting using cloud cover index," *Solar Energy*, vol. 86, no. 12, pp. 3531–3543, 2012.

[33] Chen Yang, Anupam A. Thatte, and Le Xie, "Multitime-scale data-driven spatio-temporal forecast of photovoltaic generation," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 1, pp. 104–112, 2015.

[34] James W. Taylor, "Short-term electricity demand forecasting using double seasonal exponential smoothing," *Journal of the Operational Research Society*, vol. 54, no. 8, pp. 799–805, 2003.

[35] James W. Taylor and Ralph D. Snyder, "Forecasting intraday time series with multiple seasonal cycles using parsimonious seasonal exponential smoothing," *Omega*, vol. 40, no. 6, pp. 748–757, 2012.

[36] Erasmo Cadenas, Oscar A. Jaramillo, and Wilfrido Rivera, "Analysis and forecasting of wind velocity in chetumal, quintana roo, using the single exponential smoothing method," *Renewable Energy*, vol. 35, no. 5, pp. 925–930, 2010.

[37] Andrew Kusiak and Zijun Zhang, "Short-horizon prediction of wind power: a data-driven approach," *IEEE Transactions on Energy Conversion*, vol. 25, no. 4, pp. 1112–1122, 2010.

[38] Patrick Mathiesen and Jan Kleissl, "Evaluation of numerical weather prediction for intra-day solar forecasting in the continental United States," *Solar Energy*, vol. 85, no. 5, pp. 967–977, 2011.

[39] Mashud Rana, Irena Koprinska, and Vassilios G. Agelidis, "Univariate and multivariate methods for very short-term solar photovoltaic power forecasting," *Energy Conversion and Management*, vol. 121, pp. 380–390, 2016.

[40] Hugo T. C. Pedro, Carlos F. M. Coimbra, Mathieu David, and Philippe Lauret, "Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts," *Renewable Energy*, vol. 123, pp. 191–203, 2018.

[41] Mashud Rana, Irena Koprinska, and Vassilios G. Agelidis, "2D-interval forecasts for solar power production," *Solar Energy*, vol. 122, pp. 191–203, 2015.

[42] Zheng Wang, Irena Koprinska, and Mashud Rana, "Solar power prediction using weather type pair patterns," 2017, pp. 4259–4266, IEEE.

[43] Mehmet Yesilbudak, Seref Sagiroglu, and Ilhami Colak, "A new approach to very short term wind speed prediction using k-nearest neighbor classification," *Energy Conversion and Management*, vol. 69, pp. 77–86, 2013.

[44] Mashud Rana, Irena Koprinska, Alicia Troncoso, and Vassilios G. Agelidis, "Extended weighted nearest neighbor for electricity load forecasting," 2016, vol. 9887, pp. 299–307.

[45] Tommaso Colombo, Irena Koprinska, and Massimo Panella, "Maximum Length Weighted Nearest Neighbor approach for electricity load forecasting," 2015, pp. 1–8, IEEE.

[46] Hugo T.C. Pedro and Carlos F.M. Coimbra, "Short-term irradiance forecastability for various solar micro-climates," *Solar Energy*, vol. 122, pp. 587 – 602, 2015.

[47] Yinghao Chu, Bryan Urquhart, Seyyed M.I. Gohari, Hugo T.C. Pedro, Jan Kleissl, and Carlos F.M. Coimbra, "Short-term reforecasting of power output from a 48 MWe solar PV plant," *Solar Energy*, vol. 112, pp. 68 – 77, 2015.

[48] Yinghao Chu and Carlos F.M. Coimbra, "Short-term probabilistic forecasts for Direct Normal Irradiance," *Renewable Energy*, vol. 101, pp. 526 – 536, 2017.

[49] Chaorong Chen and Unit T. Kartini, "K-nearest neighbor neural network models for very short-term global solar irradiance forecasting based on meteorological data," *Energies*, vol. 10, no. 2, pp. 186, 2017.

[50] Francisco Martinez-Alvarez, Alicia Troncoso, Jose C. Riquelme, and Jesus S. Aguilar Ruiz, "Energy time series forecasting based on pattern sequence similarity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 8, pp. 1230–1243, 2011.

[51] Imtiaz Ashraf and Avinash Chandra, "Artificial neural network based models for forecasting electricity generation of grid connected solar PV power plant," *International Journal of Global Energy Issues*, vol. 21, no. 1-2, pp. 119–130, 2004.

[52] Bhim Singh, Dilip T. Shahani, and Arun K. Verma, "Neural network controlled grid interfaced solar photovoltaic power generation," *IET Power Electronics*, vol. 7, no. 3, pp. 614–626, 2014.

[53] Chokri Ben Salah and Mohamed Ouali, "Comparison of fuzzy logic and neural network in maximum power point tracker for PV systems," *Electric Power Systems Research*, vol. 81, no. 1, pp. 43–50, 2011.

[54] Amit K. Yadav and S. S. Chandel, "Solar radiation prediction using Artificial Neural Network techniques: A review," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 772–781, 2014.

[55] Emanuele Ogliari, Francesco Grimaccia, Sonia Leva, and Marco Mussetta, "Hybrid predictive models for accurate forecasting in PV systems," *Energies*, vol. 6, no. 4, pp. 1918–1929, 2013.

[56] Shuaixun Chen, H. B. Gooi, and Mingqiang Wang, "Solar radiation forecast based on fuzzy logic and neural networks," *Renewable Energy*, vol. 60, pp. 195–201, 2013.

[57] Evaggelos G. Kardakos, Minas C. Alexiadis, Stylianos I. Vagropoulos, Christos K. Simoglou, Pandelis N. Biskas, and Anastasios G. Bakirtzis, "Application of time series and artificial neural network models in short-term forecasting of PV power generation," 2013, pp. 1–6, IEEE.

[58] Adel Mellit, Safak Saglam, and Soteris A. Kalogirou, "Artificial neural network-based model for estimating the produced power of a photovoltaic module," *Renewable Energy*, vol. 60, pp. 71–78, 2013.

[59] Florencia Almonacid, Pedro J. Prez-Higueras, Eduardo F. Fernndez, and L. Hontoria, "A methodology based on dynamic artificial neural network for short-term forecasting of the power output of a PV generator," *Energy Conversion and Management*, vol. 85, pp. 389–398, 2014.

[60] Adel Mellit and Alessandro M. Pavan, "A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy," *Solar Energy*, vol. 84, no. 5, pp. 807–821, 2010.

[61] Kahina Dahmani, Rabah Dizene, Gilles Notton, Christophe Paoli, Cyril Voyant, and Marie L. Nivet, "Estimation of 5-min time-step data of tilted solar global irradiation using ANN (Artificial Neural Network) model," *Energy*, vol. 70, pp. 374–381, 2014.

[62] Tiong T. Teo, Thillainathan Logenthiran, and W. L. Woo, "Forecasting of photovoltaic power using extreme learning machine," 2015;2016;, pp. 1–6, IEEE.

[63] Maria G. De Giorgi, Paolo M. Congedo, and Maria Malvoni, "Photovoltaic power forecasting using statistical methods: impact of weather data," *IET Science, Measurement and Technology*, vol. 8, no. 3, pp. 90–97, 2014.

[64] Gilles Notton, Christophe Paoli, Liliana Ivanova, Siyana Vasileva, and Marie L. Nivet, "Neural network approach to estimate 10-min solar global irradiation values on tilted planes," *Renewable Energy*, vol. 50, pp. 576–584, 2013.

[65] Badia Amrouche and Xavier Le Pivert, "Artificial neural network based daily local forecasting for global solar radiation," *Applied Energy*, vol. 130, pp. 333–341, 2014.

[66] Atsushi Yona, Tomonobu Senjyu, Toshihisa Funabashi, and Chul-Hwan Kim, "Determination method of insolation prediction with fuzzy and applying neural network for long-term ahead PV power output correction," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 2, pp. 527–533, 2013.

[67] Huan Long, Zijun Zhang, and Yan Su, "Analysis of daily solar power prediction with data-driven approaches," *Applied Energy*, vol. 126, pp. 29–37, 2014.

[68] Rasool Azimi, Mohadeseh Ghayekhloo, and Mahmoud Ghofrani, "A hybrid method based on a new clustering technique and multilayer perceptron neural networks for hourly solar radiation forecasting," *Energy Conversion and Management*, pp. 331–344, 2016.

[69] Jie Shi, Wei-Jen Lee, Yongqian Liu, Yongping Yang, and Peng Wang, "Forecasting Power Output of Photovoltaic Systems Based on Weather Classification and Support Vector Machines," *IEEE Transactions on Industry Applications*, vol. 48, no. 3, pp. 1064–1069, 2012.

[70] Adel Mellit, A. Massi Pavan, and Vanni Lughi, "Short-term forecasting of power production in a large-scale photovoltaic plant," *Solar Energy*, vol. 105, pp. 401–413, 2014.

[71] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[72] Adel Mellit, Alessandro M. Pavan, and Mohamed Benghanem, "Least squares support vector machine for short-term prediction of meteorological time series," *Theoretical and Applied Climatology*, vol. 111, no. 1, pp. 297–307, 2013.

[73] Makbul A. M. Ramli, Ssennoga Twaha, and Yusuf A. Al-Turki, "Investigating the performance of support vector machine and artificial neural networks in predicting solar radiation on a tilted surface: Saudi Arabia case study," *Energy Conversion and Management*, vol. 105, pp. 442–452, 2015.

[74] Kasra Mohammadi, Shahaboddin Shamshirband, Chong W. Tong, Muhammad Arif, Dalibor Petkovi, and Sudheer Ch, "A new hybrid support vector machine wavelet transform approach for estimation of horizontal global solar radiation," *Energy Conversion and Management*, vol. 92, pp. 162–171, 2015.

[75] Jilong Chen, Guosheng Li, and Shengjun Wu, "Assessing the potential of support vector machine for estimating daily solar radiation using sunshine duration," *Energy Conversion and Management*, vol. 75, pp. 311–318, 2013.

[76] Bjrn Wolff, Jan Khnert, Elke Lorenz, Oliver Kramer, and Detlev Heinemann, "Comparing support vector regression for PV power forecasting to a physical modeling approach using measurement, numerical weather prediction, and cloud motion data," *Solar Energy*, vol. 135, pp. 197–208, 2016.

[77] Betul B. Ekici, "A least squares support vector machine model for prediction of the next day solar insolation for effective use of PV systems," *Measurement*, vol. 50, no. 1, pp. 255–262, 2014.

[78] Lanre Olatomiwa, Saad Mekhilef, Shahaboddin Shamshirband, Kasra Mohammadi, Dalibor Petkovi, and Ch Sudheer, "A support vector machinefirefly algorithm-based model for global solar radiation prediction," *Solar Energy*, vol. 115, pp. 632–644, 2015.

[79] Hong-Tzer Yang, Chao-Ming Huang, Yann-Chang Huang, and Yi-Shiang Pai, "A weather-based hybrid method for 1-day ahead hourly forecasting of PV power output," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 3, pp. 917–926, 2014.

[80] Mashud Rana, Irena Koprinska, and Vassilios G. Agelidis, "Forecasting solar power generated by grid connected PV systems using ensembles of neural networks," in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–8.

[81] Mashud Rana, Irena Koprinska, and Vassilios G. Agelidis, "Solar power forecasting using weather type clustering and ensembles of neural networks," 2016, pp. 4962–4969, IEEE.

[82] Muhammad Qamar Raza, Nad Mithulananthan, and Alex Summerfield, "Solar output power forecast using an ensemble framework with neural predictors and Bayesian adaptive combination," *Solar Energy*, vol. 166, pp. 226 – 241, 2018.

[83] Zhaoxuan Li, S. M. Mahbobur Rahman, Rolando Vega, and Bing Dong, "A hierarchical approach using machine learning methods in solar photovoltaic energy production forecasting," *Energies*, vol. 9, no. 1, pp. 55, 2016.

[84] Moufida Bouzerdoum, Adel Mellit, and A. Massi Pavan, "A hybrid model (SARIMASVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant," *Solar Energy*, vol. 98, pp. 226–235, 2013.

[85] Yuan-Kang Wu, Chao-Rong Chen, and Hasimah A. Rahman, "A novel hybrid model for short-term forecasting in PV power generation," *International Journal of Photoenergy*, vol. 2014, pp. 1–9, 2014.

[86] Zibo Dong, Dazhi Yang, Thomas Reindl, and Wilfred M. Walsh, "Satellite image analysis and a hybrid ESSS/ANN model to forecast solar irradiance in the tropics," *Energy Conversion and Management*, vol. 79, pp. 66–73, 2014.

[87] Alberto Dolara, Francesco Grimaccia, Sonia Leva, Marco Mussetta, and Emanuele Ogliari, "A physical hybrid artificial neural network for short term forecasting of PV plant power output," *Energies*, vol. 8, no. 2, pp. 1138–1153, 2015.

[88] Alessandro Gandelli, Francesco Grimaccia, Sonia Leva, Marco Mussetta, and Emanuele Ogliari, "Hybrid model analysis and validation for PV energy production forecasting," 2014, pp. 1957–1962, IEEE.

[89] Jorge M. Filipe, Ricardo J. Bessa, Jean Sumaili, R. Tome, and Joao N. Sousa, "A hybrid short-term solar power forecasting tool," 2015, pp. 1–6, IEEE.

[90] Cyril Voyant, Marc Muselli, Christophe Paoli, and Marie-Laure Nivet, "Numerical weather prediction (NWP) and hybrid ARMA/ANN model to predict global radiation," *Energy*, vol. 39, no. 1, pp. 341–355, 2012.

[91] Ricardo Marquez and Carlos F. M. Coimbra, "Forecasting of global and direct solar irradiance using stochastic learning methods, ground experiments and the NWS database," *Solar Energy*, vol. 85, no. 5, pp. 746–756, 2011.

[92] Jun Liu, Wanliang Fang, Xudong Zhang, and Chunxiang Yang, "An improved photovoltaic power forecasting model with the assistance of aerosol index data," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 2, pp. 434–442, 2015.

[93] "University of Queensland Solar Data [Online]," `http://www.uq.edu.au/solarenergy/`.

[94] Australian Bureau of Meteorology, "Climate Data [Online]," `http://www.uq.edu.au/solarenergy/`.

[95] Zheng Wang, Irena Koprinska, and Mashud Rana, "Clustering based methods for solar power forecasting," 2016, pp. 1487–1494, IEEE.

[96] Gavin Brown and Ludmila I. Kuncheva, ""Good" and "Bad" diversity in majority vote ensembles," in *Multiple Classifier Systems*, Neamat El Gayar, Josef Kittler, and Fabio Roli, Eds., Berlin, Heidelberg, 2010, pp. 124–133, Springer Berlin Heidelberg.

[97] Javier Antonanzas, Natalia Osorio, Rodrigo Escobar, Ruben Urraca, Francisco J. Martinez-de Pison, and Fernando Antonanzas-Torres, "Review of photovoltaic power forecasting," *Solar Energy*, vol. 136, pp. 78–111, 2016.

[98] Maimouna Diagne, Mathieu David, Philippe Lauret, John Boland, and Nicolas Schmutz, "Review of solar irradiance forecasting methods and a proposition for

small-scale insular grids," *Renewable and Sustainable Energy Reviews*, vol. 27, pp. 65–76, 2013.

[99] Zheng Wang, Irena Koprinska, Irena Koprinska, Alicia Troncoso, and Francisco Martinez-Alvarez, "Static and dynamic ensembles of neural networks for solar power forecasting," 2018, pp. 1–8, IEEE.

[100] Zheng Wang and Irena Koprinska, "Solar power forecasting using dynamic meta-learning ensemble of neural networks," in *Artificial Neural Networks and Machine Learning – ICANN*. 2018, pp. 528–537, Springer International Publishing.

[101] Vítor Cerqueira, Luís Torgo, Fábio Pinto, and Carlos Soares, "Arbitrated ensemble for time series forecasting," in *Machine Learning and Knowledge Discovery in Databases*. 2017, pp. 478–494, Springer International Publishing.