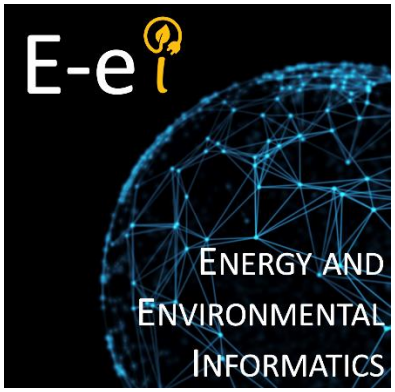
Distributed Photovoltaic System Capacity Estimation using Feeder Load Data based on Machine Learning

Lingxi Tang

Harris Manchester College

18 May 2022



Acknowledgements

I would first and foremost like to thank my two co-supervisors, Prof. David Wallom, for his consistent guidance and liaising with other stakeholders on the project's behalf, and Dr. Masaō Ashtine, for his valuable feedback, constant encouragement, and steadfast support. I would also like to thank Dr. Maomao Hu and Dr. Weiqi Hua for their generous help and support towards this project.

Finally, the successful completion of this project is only possible with the datasets and relevant information provided by colleagues from the University of Strathclyde, Dr. Bruce Stephen and Dr. Rory Telford. I would like to express my utmost gratitude to them for their help and kindness.

Abstract

Residential solar photovoltaic (PV) system installations are expected to continue growing due to their cost competitiveness and supportive government policies. However, excessive behind-the-meter solar panel installations can lead to technical issues, which pose a risk to the reliable operations of local power networks, as well as load prediction inaccuracies, which affect the efficient deployment of market-based demand response programmes. To address this growing concern by DNOs such as Scottish and Southern Electricity Networks in Oxfordshire, this thesis presents a simplified yet novel method for distributed PV system capacity estimation based on feature extraction and deep learning, trained using only feeder-level net-load data. The proposed model has the advantages of not requiring weather data and being able to accept time-series data of any time resolution.

A multi-step hyperparameter optimisation process first was performed on the artificial neural network. Experimentation processes were then conducted to test the proposed model's sensitivity to the time of data collection, specifically the season and day of the week. The method's sensitivity to the number of households served by the substation and the proportion of PV-equipped households is also explored. The results proved the feasibility of using machine learning to estimate installed PV capacity at a feeder level, without the use of weather data or satellite imagery. Additionally, it was shown that having more training data is the most important aspect to improving the performance of the proposed model. However, if constrained by data collection and storage capabilities, DNOs should prioritise collecting data and using the model during either the summer months or from Mondays to Thursdays. Most notably, conducting DPVSCE exclusively in summer only requires a third of the data to perform as well as a model which is deployable all year round. Finally, the results show that the model is most effective when there are more households associated with the substation feeder or more households are equipped with PV systems.

The work presented in this thesis was selected to be presented at the opening ceremony of The Energy Systems Accelerator in Oxford on 26th May 2022.

Content

Nomenclature

AE	Absolute Error
ANN	Artificial Neural Network
DNN	Deep Neural Network
DNO	Distribution Network Operator
DPVS	Distributed Photovoltaic System
DPVSC	Distributed Photovoltaic System Capacity
DPVSCE	Distributed Photovoltaic System Capacity Estimation
IFEEL	Interpretable Feature Extraction of Electricity Loads
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
PV	Photovoltaic
PVGIS	Photovoltaic Geographical Information System
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
SSEN	Scottish and Southern Electricity Networks
SVR	Support Vector Regression
Wp	Peak Power (Watt)

1 Introduction

According to the International Renewable Energy Agency and International Energy Agency, global photovoltaic (PV) capacity has increased rapidly in recent years, with an average of 98.8 GW of PV capacity installed annually from 2017 to 2021 [1]. As such, global installed PV capacity is expected to rise sixfold over the next decade, from a total of 480 GW in 2018 to 8,519 GW by 2050 [2]. Falling costs and strong supporting policies, such as government subsidies, have also encouraged the growth of distributed PV systems (DPVS) globally, with annual residential solar PV additions increasing from 6 GW in 2017 to 16 GW in 2019 and expected to maintain at 16 GW for the next three years [2]. In the UK alone, total residential DPVS deployment, categorised as PV systems with capacities of 10 kilowatts-peak (kWp) and below [3], have grown from about 0.7 GW to 3.2 GW over the last decade and accounts for 23.7% of all solar PV installations as of December 2021 [4].

In the UK, there are no requirements to seek prior approval for residential PV system installations, nor are there any post-installation registration requirements [5, 6]. This leads to many DPVS being unknown to or incorrectly registered with distribution network operators (DNO) as customers might not follow official registration processes or proceed to install additional DPVS after registration [7]. With a high penetration of unknown DPVS, a multitude of technical problems may arise, such as overvoltage, thermal loading issues and reverse power flow, all of which have potential to damage grid equipment and cause safety and reliability issues [7, 8]. Additionally, high penetration of DPVS will also reduce the accuracy of demand response capacity estimation, customer base load estimation and load forecasting, which can negatively impact the effectiveness of market-based DR programmes and in turn, the reliable operations of power networks [9, 10].

While it is technically possible to conduct manual checking and monitoring of DPVS installations, the resource costs associated are undesirable and errors will be inevitable. As such, it is necessary to develop accurate methods for automatic DPVS capacity estimation (DPVSCE), so that DNOs (transitioning to Distribution System Operators or DSOs) can plan for and avoid potential technical and operational issues as mentioned above. Ultimately, progress in DPVSCE technology will contribute greatly to the continued growth and smooth integration of DPVS in distribution networks, potentially leading to significant economic, health and climate benefits [11]. As such, this thesis aims to contribute to automatic DPVSCE technology by developing a novel method based on feature extraction and machine learning and assessing the effectiveness of the proposed model in response to changing specific variables.

Research findings from this project will be disseminated to interested stakeholders via different means. This work will be translated into a research poster for presentation at conferences, as well as a paper, to be

submitted to a high-impacts journal for publication. From there, the goal is to contribute to the development of a cost-effective and scalable method for DPVSCE, to ensure the safe and sustainable growth of distributed renewable energy sources.

Following this introductory chapter, this report will be structured as follows: Section 2 first outlines recent research efforts in DPVSCE methods, the research gaps present in the field, and how the proposed method and subsequent model fills these gaps. This section will explain crucial relevant concepts featured in the proposed method, namely artificial neural networks (ANNs) and the Interpretable Feature Extraction of Electricity Loads, or IFEEL, package[1] [12]. Section 3 will introduce and explain the process of obtaining feeder net-load datasets used in this thesis. Together, Sections 2 and 3 provide the necessary background information required to understand the design principles and key components.

Details of the DPVSCE model and its design principles are then introduced in Section 4. After the proposed method has been introduced, Section 5 provides details on the sensitivity analysis to the training data's time of collection, the number of household properties served by the substation feeder and the proportion of PV-equipped households. Section 6 will present the experiment results and discuss the model's practical implications for industry applications, before Section 7 summarises the report and proposes potential next steps for future extensions of the project. Note that all scripts developed and used for this thesis can be found in the accompanying Github repository[2].

[1] IFEEL package github repository: https://github.com/chacehoo/IFEEL

[2] Thesis' accompanying github repository: https://github.com/LingxiTang/ML-DPVSCE

2 Literature Review and Key Concepts

Before introducing the proposed method, it is necessary to first understand the state of the art in DPVSCE technology, as well as important concepts that underpin the proposed method. This section will first present the most recent research in installed PV capacity estimation technology and highlight research gaps, followed by introducing ANNs and the IFEEL package, which will be featured heavily in the proposed method. From this subsection, readers will understand the heavy constraints of current DPVSCE methods and the industry's need for a simpler and more accessible alternative.

2.1 Related DPVSCE research

Understanding current research trends is key in ensuring that the proposed method is pioneering and addresses current research gaps. Recent research in DPVSCE methods have focussed on either being based on satellite imagery data or based on electricity load data. To provide a comprehensive review of DPVSCE methods, satellite image-based methods will first be introduced, followed by methods based on electricity load data.

Satellite image-based methods are based on the logic that solar PV panels must be captured by satellite images since solar PV arrays must be exposed to the sun for electricity generation, and so, information on installed DPVS can be obtained from these images. Recent and relevant research have been limited to two pieces of work at the time of writing. Charabi *et al.* [13] proposed using Geographic Information System (GIS) data to estimate the potential roof-PV generation capacity in a geographical region. In their work, the Arc-GIS software was heavily utilised to obtain roof area and solar radiation data from satellite images, which were then used to calculate the potential PV generation capacity in a residential area of Oman. Taking one step further towards DPVSCE, Malof *et al.* [14] presented a supervised machine learning (ML) algorithm to detect PV arrays using high-resolution colour satellite images. To process the satellite images into a compatible input format for ML methods, the authors manually annotated visible PV panels in the images, as shown by the red lines in Fig. 2.1. The annotated datasets were then used to train a random forest algorithm for PV panel detection. The information obtained from this method can be used to further infer DPVS capacity.

While these works clearly exhibit the feasibility of using satellite images to detect the location of PV panels and estimate their physical sizes, there are limitations in this approach compared to using electricity load data. First, these methods are dependent on the availability of satellite imagery, which may not be as reliable as electricity load data for DNOs. After all, DNOs have direct access to electricity consumption data [15] but can only obtain satellite images via third parties, which requires additional data purchase costs and

administrative delays. Secondly, satellite image-based DPVSCE assumes a constant relationship between physical size of PV panels and their electricity generation capacity. However, PV generation capacity per unit area can vary greatly among different arrays due to differences in efficiency, rendering the assumption limiting in scope. Finally, even if the capacity of location specific DPVS can be accurately estimated, there is limited information on the substation feeder to which it is connected [16]. Thus, the estimations will not be useful in planning for potential technical and operational issues at specific substations.

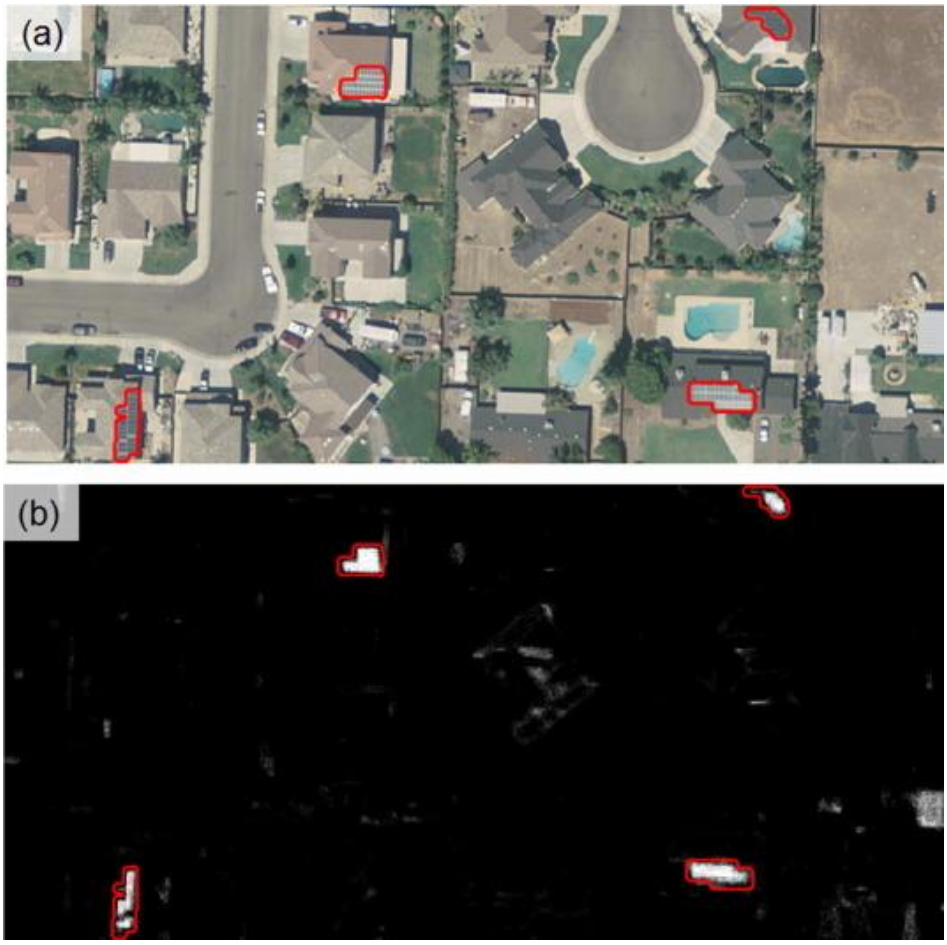


Figure 2.1: Results of satellite image-based PV panel detection algorithm proposed by Malof *et al.* [14], with (a) the original aerial image, and (b) the confidence map of PV panel detected. The red lines indicate the true PV panel locations.

Due to the heavy constraints of using satellite images for DPVSCE, there are more research efforts in DPVSCE methods using electricity load data. A few examples will be highlighted as follows. For instance, Wang *et al.* [9] proposed a two-stage PV detection and estimation method based on support vector machine. The first stage adopted a support vector classification model to detect the presence of DPVS from a household net-load curve. The second stage involves estimating the capacity of detected DPVS using a support vector regression (SVR) model with specific extracted features. Another SVR-based approach performs DPVSCE based on features extracted from the difference between consumer net-load curves in different weather conditions [10]. Zhang and Grijalva [8] presented a data-driven approach for PV detection and capacity

estimation. They adopted a change-point detection algorithm to identify unauthorised PV installations based on unusual energy consumption behaviours. The size of the installed DPVS was estimated based on the relationship between local cloud cover index and smart meter measurements. The methods discussed so far prove the capabilities of using load data to perform DPVSCE. However, they are only specific to DPVSCE at an individual household level, instead of an aggregated substation feeder level. Unfortunately, household-level DPVSCE is inferior to feeder-level DPVSCE in terms of delivering value in network planning, as operations and maintenance decisions are mostly made at a feeder level, rather than household-level [7]. Another limitation of household-level DPVSCE is its requirement of individual household load data which can be difficult to obtain due to the costs associated with fitting individual households with smart meters and public concerns on privacy of electricity consumption data [17].

With feeder-level data being cheaper and more easily accessible for DNOs [7], there has been more extensive research of accurately disaggregating PV generation from feeder-level net-load data. Proposed feeder-level PV disaggregation methods include using multiple linear regression [18], an adaptive framework using a combination of ML techniques (linear regression, decision tree, random forest, and multilayer perceptron) [19], multi-layer perception deep neural network [20] and a probabilistic model based on multi-quantile recurrent neural network models [21]. Although there are considerable relevant benefits of separating PV outputs from combined load data, only knowledge on PV capacity allows DNOs to prepare for the maximum potential PV generation which might happen during unprecedented weather scenarios, which is expected to become more frequent due to climate change [22].

Despite the discussed benefits, there have been limited research efforts in feeder-level DPVSCE, with the only recent research being a novel quantile analysis approach which takes the uncertainty in historical feeder load and irradiance data into account [7]. This method requires three sets of input data: historical solar irradiance data, electricity consumption data of individual households without PV or with known PV capacity and feeder net-load data. Unfortunately, these requirements on input data can be difficult to fulfil. For instance, the need for solar irradiation data greatly restricts the method's usability to locations equipped with weather monitoring equipment and household electricity consumption data can be difficult to obtain for aforementioned reasons.

As such, this thesis will propose a method which aims to overcome these limitations by having the simple requirement of just feeder net-load data, making it simple to use and applicable for many data-constrained substations. Ultimately, this thesis aims to further expand upon the field of data-based feeder-level DPVSCE.

2.2 Artificial neural networks

Similar to many of the methods outlined above, the proposed method will utilise machine learning, which describes a class of algorithms with the ability to acquire knowledge, by learning patterns from raw data [23]. ML has surged in popularity in the digital age, as the increased amount of data empowers the effectiveness of ML algorithms [23]. The proposed method in this thesis will employ a common ML algorithm: the artificial neural network, and this subsection explains what it is, how it works and its advantages and disadvantages.

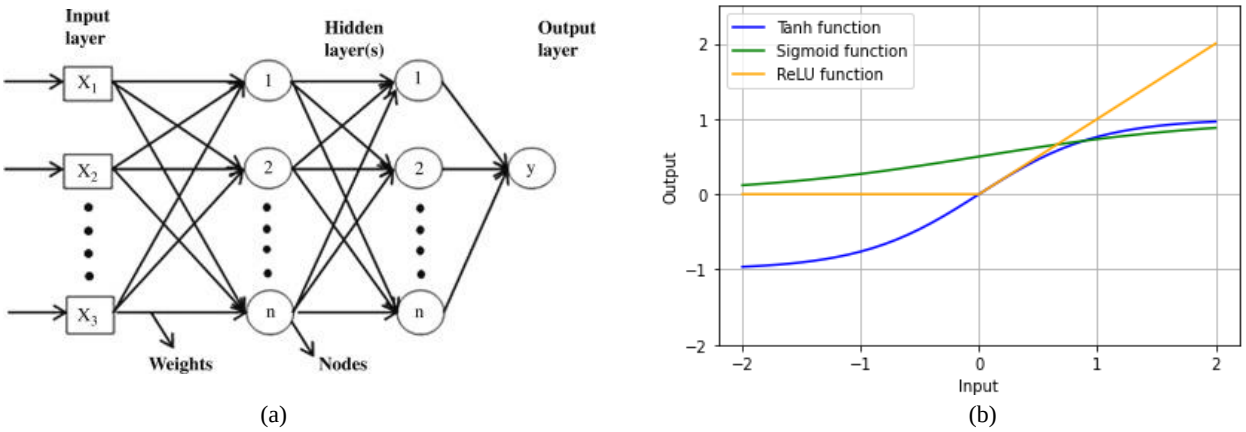
The ANN is an ML algorithm which aims to learn a non-linear function which maps a set of input values to an output value:

$$f(\cdot): X^m \rightarrow Y \tag{1}$$

where X is the set of m input features and Y is the output variable. This function is obtained using a dataset

$$D: \{X_i^m, Y\}, \quad i = 1, 2, \dots, N \tag{2}$$

with N samples, and each sample containing a set of input features and its corresponding output value. ML algorithms which require a set of data samples with known output value, or known as "labels", are also called "supervised learning" algorithms. The idea is that the labelled dataset can supervise or "train" the algorithm to predict Y accurately when given a new sample X^m.



(a) (b)

Figure 2.2: (a) Structure of an ANN with two hidden layers [24] (b) Common activation functions used in ANNs – the ReLU, logistic sigmoid and tanh functions

The basic structure of an ANN has three main components: input layer, hidden layer(s) and output layer (see Fig. 2.2a). The input layer contains a set of neurons, each representing an input feature value. Each hidden layer contains a set of neurons which converts the values in the previous layer using the following function:

$$\sigma(w^1 x^1 + w^2 x^2 + \cdots + w^n x^n) \tag{3}$$

where σ is a non-linear activation function, w is individual weight values, x is values in the previous layer and n is the number of neurons in the previous layer. The number of hidden layers and the number of neurons in

each hidden layer are hyperparameters which are to be set manually. An ANN with 2 or more hidden layers are also known as deep neural networks (DNNs). The activation function σ is also a hyperparameter and examples include the Rectified Linear Unit (ReLU), logistic sigmoid and hyperbolic tan (tanh) functions [23], as depicted in Fig. 2.2b. Finally, the output layer has a single neuron which also performs the operation shown in Eq. (3), with the special condition that the activation function is fixed as the identity function, i.e. no activation function is used. Note that this condition only applies for ANNs used for regression (which is the case for DPVSCE), rather than classification. In an ANN, the flow of information from the input layer to the output layer is also known as "forward propagation".

In an ANN, the key parameter to be designed is the individual weight values w and these are obtained via a process called backpropagation [25]. The main idea behind backpropagation is as follows: given a labelled training dataset, a loss function measures the error between the predicted value generated at the output layer and the true value denoted by the label. This loss is minimised by propagating backwards from the output layer to the input layer and updating weight parameters based on gradient descent [26]. Over iterations of this process, the ANN begins to "learn" the relationship between the input features and the output value, by having the appropriate weight parameters. The specific backpropagation method used will be introduced and explained in Section 5.

There are three main advantages of an ANN. First, ANNs have the ability to learn complex non-linear models [27]. With just two hidden layers, ANNs can represent functions with any forms of shape [28]. Secondly, ANNs can constantly learn in real-time with updated data [27]. Finally, ANNs, especially those with more input features, are less susceptible to the issue of local minimums compared to other ML algorithms, due to the low probability of having all input features be at their optimal values at a single point in the cost function space [29]. However, as mentioned in [27], ANNs also have disadvantages: firstly, the randomness of weight initialisation leads to inconsistent prediction performance. ANNs also require manual tuning of hyperparameters such as number of hidden neurons, number of hidden layers and iterations. Finally, ANNs are sensitive to the scaling of input feature values. These issues will be addressed in Section 5.

For this thesis, an ANN will accept input feature values extracted from feeder-level net-load curves and output the total installed DPVSC value associated with the substation feeder. Instead of simply inputting time-series data into the ANN, a fixed set of unique features will be extracted from time-series data to be used as input features, ensuring a fixed input layer size independent of the resolution (time interval between data points) of the available time-series data. These unique features will be introduced and explained in the next subsection.

2.3 IFEEL package

First introduced in a paper by Hu *et al*. [30], the IFEEL package was first developed for the purpose of extracting features from daily net-load curves to classify them as either PV equipped, or non-PV equipped properties. Given the positive results shown in the paper, this thesis recognises the potential of using this feature extraction package to go one step further from PV detection into PV capacity estimation.

The IFEEL package takes in a 24-hour time-series dataset which comprise single-value data points which represent the amount of electrical power flow in the feeder at each time step, from 00:00 to 23:59, and outputs 14 feature values, as listed in Table 2.1. Note that within the dataset, positive values represent power consumed by properties and negative values represent power generated by DPVS. These features serve to encapsulate principal characteristics of an electricity load curve, to substitute raw time-series data as input.

Table 2.1: Physical meaning of IFEEL features

No.	Input Feature (units/range)	No.	Input Feature (units/range)
1	Mean value of 24-hour load curve (kW)	**8**	Sum of net loads during non-business hours (kW)
2	Standard deviation of 24-hour load curve (kW)	**9**	Skewness of 24-hour load curve (no units)
3	Maximum power of 24-hour load curve (kW)	**10**	Kurtosis of 24-hour load curve (no units)
4	Minimum power of 24-hour load curve (kW)	**11**	Mode of the five-bin histogram for a 24-hour load curve (kW)
5	Range of power, i.e. maximum – minimum (kW)	**12**	Longest period of successive increase above mean value (hours)
6	Percentage fraction of values above mean (0 to 1)	**13**	Longest period of successive increase (hours)
7	Sum of net loads during business hours, 9:00 – 17:00 (kW)	**14**	Number of peaks (no units)

The first five features are basic statistical metrics common in energy system analysis. The percentage fraction of values above mean (no. 6) is in fractional form and has a range of 0 to 1. The sum of net loads during business (no. 7) and non-business hours (no. 8) are total sums of all data points within the specified time periods and represents the amount of feeder-level power consumption or generation within and outside 09:00 to 17:00 respectively.

While mean and standard deviation measure the centre and spread of a dataset respectively, skewness (no.9) and kurtosis (no. 10) measure the shape of the data point distribution, in comparison to a normal distribution curve. Skewness is the third standardised moment, where the *k*-th standardised moment is the ratio of *k*-th moment about the mean to the *k*-th power of the standard deviation. Skewness measures the asymmetry

of the distribution about its mean value. A positive skewness value means that the data is right-tailed, and the mean value is larger than the median value, and vice versa. Kurtosis is the fourth standardised moment, and it measures the "sharpness" of the peak of data distribution. Kurtosis is usually used to indicate the prevalence of extreme values, with a positive kurtosis value ("heavy-tailed" distribution) representing a more "rounded" peak and the presence of extreme data points, while a negative kurtosis value represents a "sharper" peak and fewer extreme data points. These can be visualised using Fig. 2.3 below.

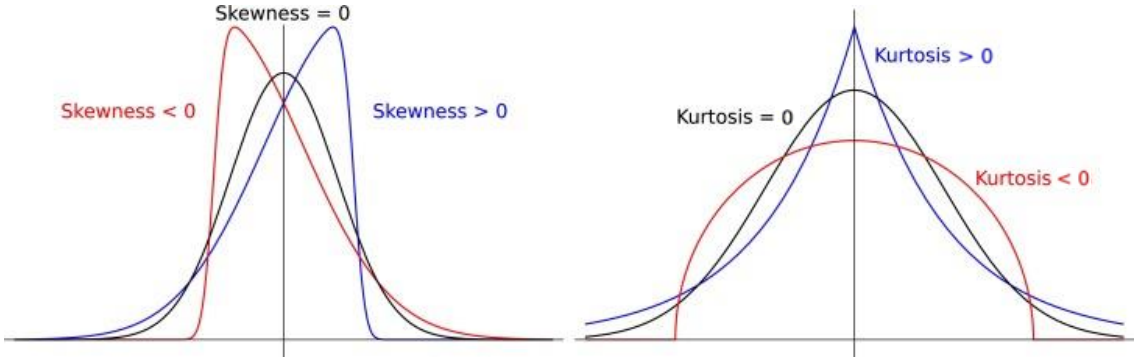


Figure 2.3: (a) Examples of a right-skewed (blue) and left-skewed (red) curves compared to a normal distribution curve. (b) Examples of a positive kurtosis (blue) and negative kurtosis (red) curves compared to a normal distribution curve (kurtosis = 0) [31]

The mode of five-bin histogram for a 24-hour load curve (no. 11) is obtained by first dividing the range of data point values into five bins of equal sizes and counting the number of data points in each bin. The average value of the bin with the most data points, i.e., the mode, will be taken as the input feature. For example, for the load curve shown in Fig. 2.4a, the five-bin histogram is shown in Fig. 2.4b. The power load value range of [0.2, 1.3] is divided into five bins of equal range and since the greatest number of data points lie in the range of [0.2, 0.42], the mode of the histogram is 0.31, i.e., the average of 0.2 and 0.42. Features no. 12 and 13 are self-explanatory, with the only clarification being that feature no. 12 requires all data point values in the sequence to be above the mean value.

To understand the process of obtaining the number of peaks (no. 14) in a 24-hour load curve, the Symbolic Aggregate approximation (SAX) representation technique [32] will first have to be introduced. The SAX representation technique converts a time-series dataset into a string of letters, where each letter represents the data point's distance from the mean. The SAX technique has three main procedures: Z-score normalisation, piecewise aggregate approximation [33] and discrete representation. The first step involves standardising the time-series dataset using Z-score normalisation by converting each data point x_t into a normalised value

$$Z_t = \frac{x_t - \bar{x}}{\sigma} \tag{4}$$

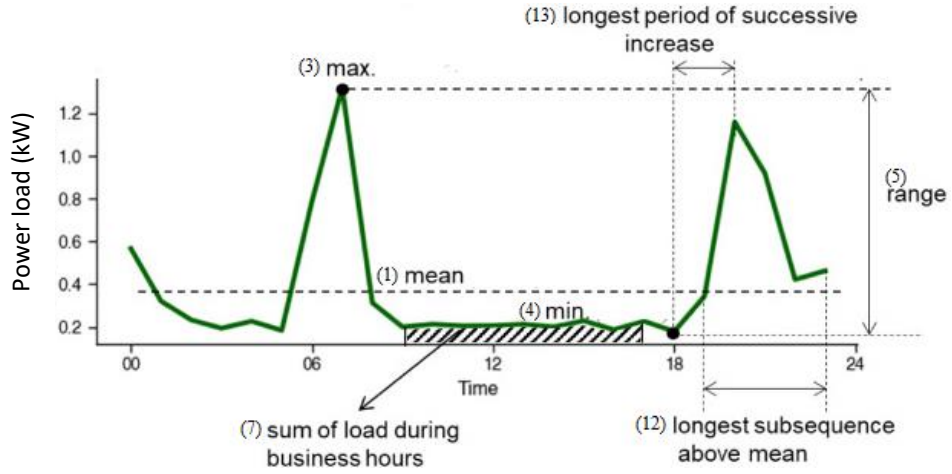
where $\bar{x}$ is the mean and σ is standard deviation of all data points. In the second step, the normalised time-series Z will be shortened into a time-series of N values, with $N = 24$ for the experiments in this thesis. To

obtain each value in the shortened time-series Z_{short}, Z is first divided into N sub-sequences of equal lengths and the mean value of each sub-sequence is extracted to form Z_{short}. Finally, Z_{short} is converted into a string of N letters, with each letter representing a range with equal probability in the normal distribution. For the experiments in this thesis, N is set at 7 and the letters available are "a" to "g", with the range of normalised values represented shown in Table 2.2 below.

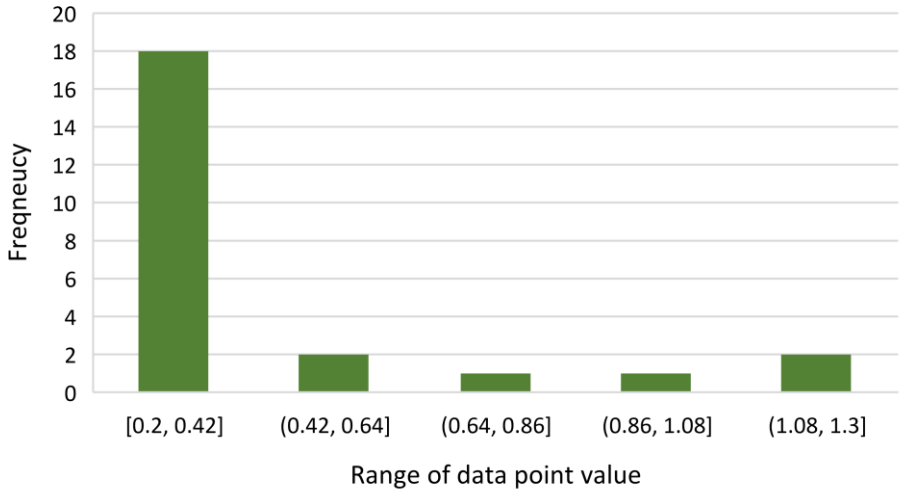
Table 2.2: SAX letters and their corresponding range of normalised values. Each letter represents a range of equal probability in a normal distribution. This table is obtained from Table 2 of [34].

SAX Letter	**a**	**b**	**c**	**d**	**e**	**f**	**g**
Range	$(-\infty, -1.07)$	$(-1.07, -0.57)$	$(-0.57, -0.18)$	$(-0.18, 0.18)$	$(0.18, 0.57)$	$(0.57, 1.07)$	$(1.07, \infty)$

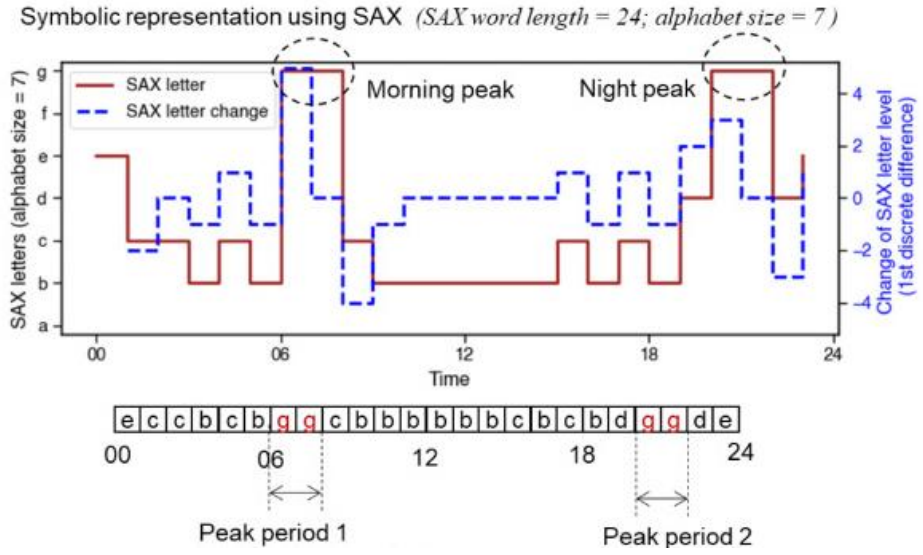
After the 24-hour load curve has been converted into a string of 24 letters using the SAX representation technique, groups of (one or more) letter "g" represents peaks and the number of peaks (feature no. 14) can then be obtained. For example, Fig. 2.4c depicts the SAX representation of the example 24-hour load curve shown in Fig. 2.4a, and two peaks are detected.



(a)



(b)



(c)

Figure 2.4: Graphical depiction of specific input features, obtained from (a) an example 24-hour power load curve [30]. (b) shows the 5-bin histogram of the power load curve for the extraction of feature no. 11. (c) shows the peak detection process for feature no. 14 where SAX letter "g" represents peaks [30].

3 Dataset Generation

Section 2.2 has explained that an effective ANN must be trained by a dataset which, in this case, will comprise data samples of input features introduced in Section 2.3 and their corresponding installed DPVSC. This training data was extracted from net-load time-series datasets based on simulated substation feeders. The feeder-level net-load curves used in this thesis, was derived from two main components: household load consumption data and PV generation data. This section will focus on explaining the source of each component.

3.1 Household load consumption data

The household load consumption data was collected as part of the Energy Demand Research Project and was kindly provided by Dr. Bruce Stephen from the University of Strathclyde [35]. The household load consumption dataset contained the half-hourly power load consumption of 162 individual household properties based in Lanarkshire, Scotland, for year 2013. Specifically, for each property, there was one data value for power load in kW every half-hour from 00:00 01-Jan-2013 to 23:30 31-Dec-2013, resulting in a total of 17520 data points for each property. Since PV generation data will be added manually to simulate installed DPVS, it is crucial that the provided load consumption data is not affected by any electricity generation. It has been confirmed [35] that the provided dataset was first collected to represent typical load consumption behaviour and PV-equipped households would be atypical and disqualified from the dataset.

3.2 PV generation data

The PV generation data was obtained from the Photovoltaic Geographical Information System (PVGIS) [36] offered by the European Commission. By keying in specific parameters, PVGIS generates a simulated PV power generation time-series dataset, with a time resolution of one hour and in units of kW. The specific parameters used to obtain the dataset for this thesis are shown in Table 3.1 below. The rationale behind the choice of each parameter will now be explained. All the following information provided will be based on the PVGIS User Manual [37], unless otherwise specified.

Table 3.1: Input parameters in PVGIS used for PV generation dataset

Parameter name	Value	Parameter name	Value
Location coordinates	$55.677°\text{N}, -3.794°\text{E}$	**Azimuth**	$45°/0°/-45°$
Year	2013	**Installed peak PV Power**	124.2 kWp for each azimuth
Solar radiation database	PVGIS-SARAH	**PV Technology**	Crystalline Silicon
Mounting Type	Fixed	**Roof slope**	30°

The chosen location coordinates for the PV generation data were that of a residential area in Lanark, Lanarkshire, Scotland, which was identical to the location of the provided load consumption data. This choice was meant to ensure location consistency between the provided household consumption data and simulated PV power generation data. A satellite image of the chosen location is shown in Fig. 3.1 below.

(a) (b)

Figure 3.1: Satellite images with a red marker denoting the location of simulated PV systems in the PV generation dataset, with (b) being a zoomed in image of (a) [38]

Similarly, the year 2013 for the PV generation dataset was chosen to maintain temporal consistency with the provided household load consumption data. The PV generation dataset had one data value for power generated in W every hour from 00:00 01-Jan-2013 to 23:00 31-Dec-2013. Given a set of specific location coordinates, PVGIS then simulated PV power generation based on weather data. The weather database selected w athe PVGIS-SARAH database, which is a solar radiation database, which was calculated using satellite images and had a spatial resolution of about 5 km × 5 km. It was chosen because it was the only database which covers the UK (at the time of data retrieval[3]). Next, the mounting type refers to the installed PV panels' ability to shift its position based on the sun's azimuth throughout the day. "Fixed" PV panels lack this ability and are the most common type of DPVS, especially for residential PV systems. Thus, the "fixed" mounting type was selected for the simulated PV generation dataset.

The PV technology type chosen was crystalline silicon since this technology type overwhelmingly dominates the PV market at a market share of 73.30% in 2020 [39]. The roof slope was chosen to be 30° as this angle was deemed to be a realistic general assumption based on "experience of housing in Central Scotland", as communicated by Dr. Bruce Stephen [35]. This meant that the simulated PV panels would be installed at an angle of 30° from the horizontal ground. The azimuth refers to the angle of the PV panels relative

[3] Note that at the time of data retrieval, the latest version of PVGIS available was version 5.1. On 1 March 2022, PVGIS version 5.2 was released and is the latest version at the time of writing. With this new release, an updated version of the SARAH database, SARAH-2, was released and contains data up till 2020. This compares to SARAH, which only contains data up till 2016.

to the direction due South, with 0° representing panel surface directly pointing South, -90° for pointing eastward and 90° for pointing westward. For optimal performance, PV panels installed in the Northern hemisphere should be facing South to capture the most amount of solar radiation, particularly in months outside of summer. However, due to the positionings of properties, this might not be possible. Thus, 3 separate datasets of azimuth angles 45°, 0° and -45°, corresponding to the directions of south-west, south and south-east respectively, were created to account for the varying positions of properties. This basically assumed that the installed PV panels would be split equally among the three directions.

To explain the choice of the input value of installed peak PV power, it must be noted that the PV generation dataset would assume that all household properties served by a substation feeder has a grid-connected PV system installed. Based on personal communication with Steven Adams, a senior project manager at Scottish and Southern Electricity Networks (SSEN) [40], internal planning guidance for grid management sets a maximum of 75 households on a single feeder. As such, for this thesis, the absolute maximum number of households per feeder will be set at 81, so that all 162 household load data can be used to represent two separate sets of substation feeder load. To determine the installed peak PV power of 81 DPVS, the following formula is used:

$$P = A \times Pp \times 81 \tag{5}$$

where P is the total installed peak PV power associated with the feeder, A is the area of PV array installed on a single property and Pp is the peak output power per unit area of the PV array. Google Earth Pro was used to measure the roof area of a PV-equipped property based in Lanark, as shown in Fig. 3.2 below, to obtain a typical value of A.

Figure 3.2: Satellite image of 38 Lockhart Drive, a PV-equipped property based in Lanark, taken from Google Earth Pro. The yellow square indicates the installed PV panel [41].

With a top-down measured flat area of 22.18 m^2, this translates to 25.61 m^2 for A when the roof slope of 30° is considered. Pp was set at 0.18 kWp per m^2 of PV panel, a typical value for a silicon PV panel [42].

With these values put in Eq. (5), the total installed peak PV power associated with a feeder connected to 81 properties came to 372.6 kWp, with each property having installed a 4.6 kWp DPVS. To consider the varying azimuth angles mentioned previously, three datasets of azimuth angles 45°, 0° and -45° respectively, each representing a third of the total PV peak power at 124.2 kWp, was generated.

3.3 Feeder net-load data

To combine the household load consumption data with the PV power generation data to create a net-load dataset, the time resolution of the PV data was converted to half-hourly by duplicating each hourly data value. For example, an hourly data value at a time of 09:00 will be converted to the same value at 09:00 and 09:30[4]. As such, a 2013 annual time-series dataset, representing the net-load of a feeder serving 81 PV-equipped households, could be generated by summing the power load consumption of 81 household properties and subtracting the simulated PV generation data. This annual feeder-level net-load dataset has a total of 17520 data points.

To simulate substation feeders of varying DPVSC, 82 annual time-series datasets of 17520 data points are generated to represent individual feeder net-loads where 0 to 81 household properties are equipped with a 4.6 kWp solar panel. These datasets are obtained by first dividing the three PV generation datasets by 27 each to obtain PV generation data of a single PV panel facing each azimuthal direction. Then, the generation data of a single solar panel is subtracted from the household consumption data in the repeating pattern of 45°,-45° and 0° (azimuth). At each subtraction, a feeder net-load dataset with a unique DPVSC is created. This means that there will be an annual time-series dataset labelled with a DPVSC of 0 kWp, 4.6 kWp, 9.2 kWp, …, 372.6 kWp. With the provided 162-household load consumption data, the first 81 households were used to generate 82 annual time-series datasets, followed by the second set of 81 households, to create a total of 164 annual feeder net-load datasets. From these datasets, IFEEL features were then extracted to be used as training samples for the proposed DPVSCE algorithm, which will be introduced and explained in detail in Section 4.

3.4 Challenge of real-life data collection

At this stage, it is important to acknowledge that the PV generation component of the dataset was ultimately generated using a simulation model and might not be representative of real-life generation. To overcome this limitation, the process of obtaining a set of real-life ground truth data began in November 2021. A partnership with the electric utility solutions company Eneida was formed, such that they would provide

[4] When performing the time resolution transformation, each hourly data value was multiplied by 0.5 when converted to half-hourly due to misinterpreting power as energy. This was not rectified due to time constraints. This results in inaccurate magnitude of PV generation but should not affect overall results, since crucial net-load curve characteristics are preserved, as will be seen in Section 4.

anonymous real feeder-level net-load data, with known installed PV capacity tagged to it, to be used as a ground truth dataset for testing the proposed method's performance. This dataset was expected to be obtained by March 2022, in time to be utilised for the report submission deadline. However, this was delayed due to crucial legal processes, including the signing of non-disclosure agreements by the University of Oxford to ensure the legal usage of the provided data. This delay meant that this data could not be received in time to be used for this thesis. This reflects the challenge of data accessibility which stakeholders must consider before adopting deep-learning based methods. In anticipation of this delay, an alternative strategy was created and ultimately enacted: to split a portion of the simulated data to be used as ground truth data, which is a common practice in developing machine learning algorithms. As demonstrated, forward planning was key in ensuring the successful completion of this thesis, despite having faced roadblocks. In addition to non-technical challenges, such as obtaining permissions and communication delays, technical challenges, such as data collection, cleaning, and storage, must be also anticipated when developing ML-based tools.

4 Proposed Method

Thus far, Section 2 has outlined the proposed method's design objectives and key components, while Section 3 has explained the available dataset to be used. This section will introduce the proposed DPSVCE method and explain how it utilises its key components to overcome limitations in current research.

4.1 Overview of proposed DPVSCE method

The proposed prediction algorithm works as follows: it takes a 24-hour feeder net-load curve as input, extracts the relevant input features, and outputs the estimated installed DPVS capacity associated with the substation feeder. This pipeline takes advantage of the fact that the installed DPVS capacity directly determine PV power generated, which then affects the net load. These two relationships can be summarised using the following equations respectively:

$$P_{pv}(t) = C \times \eta(t) \tag{6}$$

$$P_{net}(t) = P_{consumption}(t) - P_{pv}(t) \tag{7}$$

where P_{net} is the net power load, $P_{consumption}$ is household power consumption load, P_{pv} is the PV power generated, η is the generation efficiency[5] of the installed PV system at time t, and C is the total installed DPVS capacity. As such, installed DPVS capacity can theoretically be inferred from the characteristics (or features) of the feeder net-load curve, forming the basis of DPVSCE using feeder net-load curve. The proposed pipeline is split into three main steps: averaging intra-week net-load curves, feature extraction and ML-based estimation.

From Eqs. (6) and (7), installed DPVS capacity is obtainable from net-load data if household power consumption load[6] and generation efficiency is known, or constant. Thus, the first step aims to extract known characteristics of these two variables via two procedures: sorting input net-load curves into distinct intra-week groups and averaging data within each group. These procedures are illustrated in Fig. 4.1.

[5] The generation efficiency η is a catch-all term which includes the effects of varying irradiation levels and conversion losses of PV panels.

[6] Realistically, substation power loads will also include unmetered loads, such as streetlamps or traffic lights. It will be assumed that these unmetered loads are either constant (e.g. traffic light) or have predictable seasonal patterns (e.g. light sensor-activated street lamps).

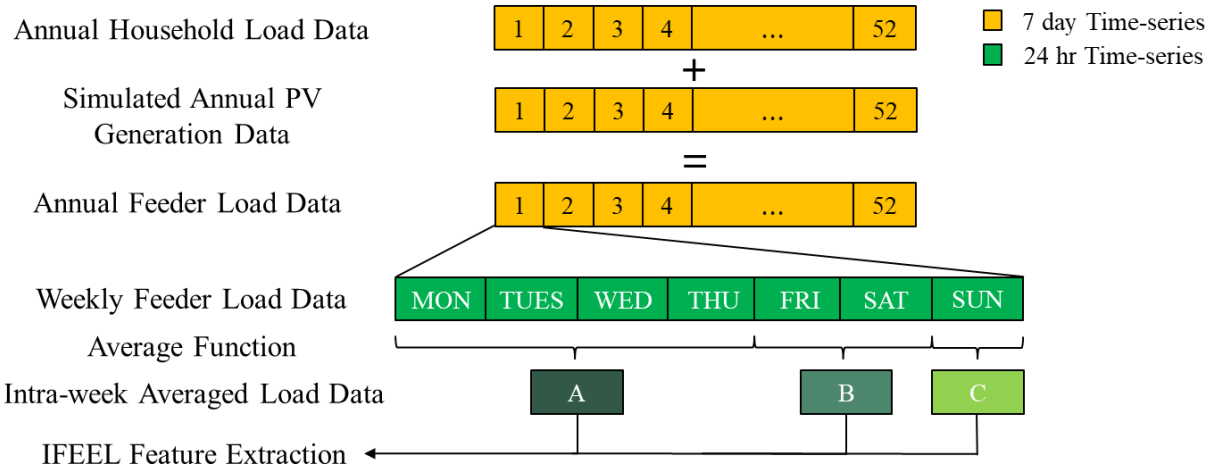


Figure 4.1: Overview of data preparation process for subsequent feature extraction and algorithm training

The first procedure serves to extract information on the household power consumption variable by making use of the distinct differences between weekdays and weekends due to differences in consumption patterns during work and non-workdays [43]. According to Wallom (2022) [16], such distinct intra-week consumption patterns can be split more precisely into 3 groups: a) Monday to Thursday, b) Friday to Saturday and c) Sunday. In this classification, Mondays to Thursdays are grouped together as these are typical workdays which exhibit similar household consumption patterns. Fridays and Saturdays are grouped together due to them being "out nights", characterised by the general behaviour of consumers returning home late on these days, reducing the evening peak in electricity consumption. Sundays are then put in a distinct group by itself. The proposed intra-week groups act as features which provide information on household power consumption $P_{consumption}$.

The second procedure is targeted towards making PV generation efficiency η known. For this variable, the main source of significant unknown uncertainty comes from the weather conditions, specifically the level of solar irradiation. By using the average of net-load curves within each intra-week group, instead of individual daily curves, the uncertainty of daily weather is reduced to a known typical weather pattern which is relatively constant within each intra-week group of each season.

From the input averaged net-load curve, a total of 14 IFEEL features, as listed in Table 2.1, will then be extracted, as the second step. In addition to the 14 IFEEL features, 2 additional features are created based on the timestamp of the input 24-hour net-load curve: intra-week group and season. These features will be input into the ANN as specific values to represent specific time periods. Intra-week group (feature no. 15) is input as 1 for Monday to Thursday, 2 for Friday to Saturday and 3 for Sunday. Season (feature no. 16) is input as 1 for summer, 2 for winter and 3 for the transitional seasons of spring and autumn. The timeframes for each season are based on the Met Office's definition [44]: the summer months are June, July, August; winter months

are December, January, February; spring months are March, April, May and autumn months are September, October and November. The purpose of the season feature is similar to that of the intra-week group feature: it serves to distinguish between season-specific household load consumption behaviour. For example, electricity demand is generally higher in winter than in summer and a steeper evening surge is usually present during winter, as compared to summer [45]. Thus, 16 feature values are input into an ANN which outputs the estimated installed DPVSC, as seen in Fig. 4.2 below.

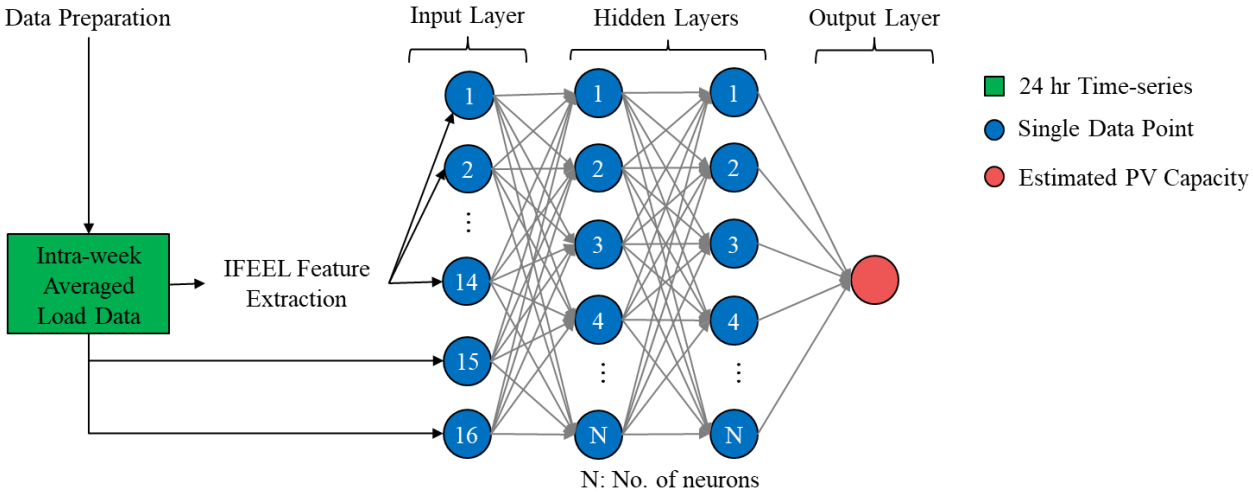


Figure 4.2: Overview of feature extraction and ML-based estimation process, with input features extracted from 24-hour averaged net-load data

This feature extraction process achieves two objectives: first, it is a method of dimensionality reduction, by removing the dimension of time and converting a time-series load curve into interpretable features. This helps to mitigate the curse of dimensionality [23, 46] and reduces computational time complexity. Second, this feature extraction process removes the requirement for the time-series input data to be of a specific time resolution. The proposed method will thus be usable regardless of the time duration between net-load data points.

4.2 IFEEL feature dataset

As explained in Section 2.2, an ANN will require an extensive dataset to train, so that the ANN can learn the relationship between 16 input features (representing feeder net-load data) and the feeder's associated DPVSC. This dataset consisting of samples with 16 input values and 1 output value was extracted from the 164 annual time-series load datasets, as explained in Section 3.3. From Fig. 4.1, one annual time-series load dataset contains an estimated total of 156 samples (52 weeks × 3 intra-week groups) of averaged 24-hour net-load curves, which means 164 annual time-series data will amount to a total of 25,584 samples. This set of data samples will be referred to as the IFEEL dataset for the rest of this thesis.

4.3 Comparison between seasons and intra-week groups

The distinct household load consumption patterns during different intra-week groups and seasons mentioned in Section 4.1 have only been theoretical thus far. To confirm these patterns via data visualisation, the averaged 24-hour net-load curves were collated from the two sets of feeder net-load curves with no PV systems installed (i.e. the 2 sets of 81 household consumption data) and classified based on season and intra-week group. Then, within each classification, all the 24-hour net-load curves were averaged and plotted in Fig. 4.3. This process was repeated for the feeders with 27, 54 and 81 PV-equipped properties, corresponding to a PV penetration rate of about 33%, 67% and 100%, to demonstrate the effects of varying amounts of installed DPVSC in a substation feeder.

From Fig. 4.3a, the household load consumption behaviours, without the influence of PV generation, can be observed. There are a few notable observations from this subplot. First of all, electricity demand generally increases from summer to transitional seasons to winter. This can be attributed to increased electricity consumption for indoor heating and other heating appliances, such as kettles and hot showers, during the winter cold. Secondly, comparing between intra-week groups, a common pattern emerges during the working hours of 09:00 to 17:00 where electricity consumption decreases in the order of Sundays, Friday/Saturdays and Monday-Thursdays. This is likely due to residents leaving their homes for work during weekdays and staying at home during the weekends. Next, the sharpest gradients during the morning and evening peaks are observed in winter, followed by transitional seasons and finally, summer. Morning and evening peaks happen when residents wake up and return home respectively to begin using electrical appliances [45]. The sharper peaks in colder and darker seasons can be attributed to increased usage of electrical appliances, especially kettles and lighting. Finally, it is worth noting the different behaviours of evening peaks between the intra-week groups. This subplot confirms the reduced evening peaks of “out nights”, as mentioned in Section 4.1, leading to less sharp evening peaks on Fridays and Saturdays. Also, evening peaks appear later on the working days of Monday-Thursday, as evening household electricity consumption only begin when residents return home from work.

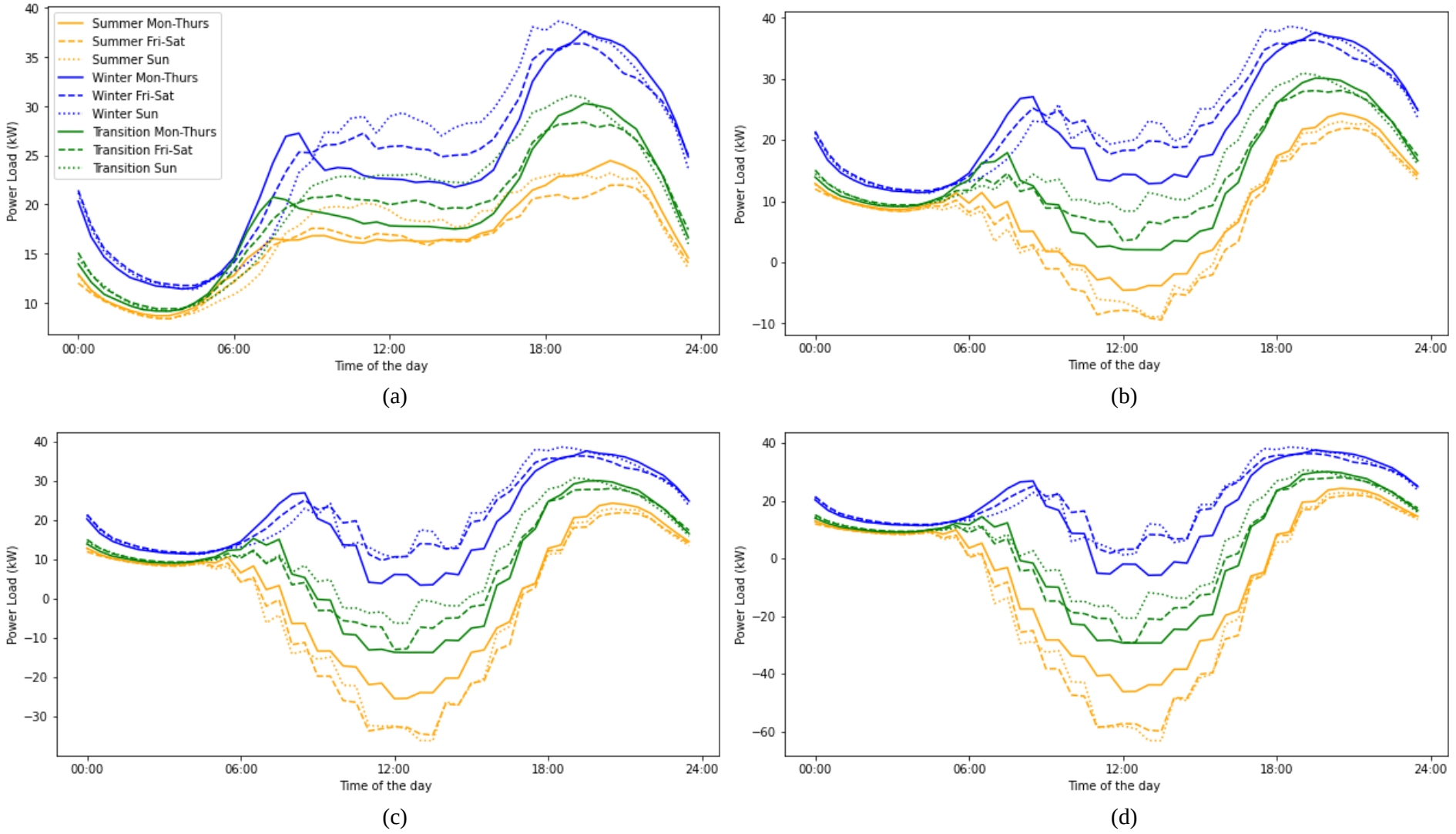


Figure 4.3: Averaged feeder-level net-load curve of 81 properties based on season and intra-week group, with the percentage of properties with (a) 0, (b) 27, (c) 54, (d) 81 PV-equipped properties, corresponding to PV penetration rates of 0%, 33%, 67% and 100% respectively.

As such, this data visualisation plot has proven the existence of distinct household load consumption patterns which are unique to each season and each intra-week group. Thus, when given the input features of season and intra-week group, the ANN is able to learn the shape of typical household load consumption curves of the specific season and intra-week group, in terms of their IFEEL features. For example, the range of power load will be lower for Fridays and Saturdays, compared to other intra-week groups, within the same season, due to the smaller evening peak. Note, however, that this data was from before the COVID-19 pandemic, and the transition to remote working will result in different electricity consumption patterns today. Thus, the use of more recent data, which highlights such changes, is a potential area for future exploration.

The four subplots in Fig. 4.3 collectively show the effects of PV power generation on 24-hour feeder net-load curves, in terms of creating the "Duck Curve" [47]. The Duck Curve represents a net-load curve under the effects of daytime electricity generation: during midday, electricity generation leads to negative net-load, causing a "belly" appearance. This is followed by a rapid increase in electricity consumption in the evening, causing an "arch" which looks like a duck's neck. In this case, as DPVSC increases in a feeder service area, the "belly" becomes more pronounced (take note that the ranges of the y-axis are different for each of the subplots), suggesting a strong correlation between DPVSC and the range of power load (input feature no. 5 in Table 2.1), making it a useful feature for DPVSCE.

5 Sensitivity Analysis Methodology

The next step was to test the performance of the proposed method and thus, determine its suitability for practical application. In this thesis, there was also an additional focus on the proposed method's sensitivity to the input features of season and intra-week group, and to the total number of households associated with the substation feeder. This section will provide detailed explanations on the two main stages of this analysis: ANN hyperparameter optimisation and sensitivity analysis. All scripts were written in the Python programming language [48] and all specified programming functions used are from the scikit-learn ML package [49], unless otherwise stated.

5.1 ANN hyperparameter optimisation

Before beginning sensitivity analysis, the hyperparameters of the ANN must first be manually tuned and optimised for best performance, as mentioned in Section 2.2. Thus, this subsection explains how specific hyperparameter values were chosen. In this thesis, the ANN was implemented using the *MLPRegressor* function [49], which allowed the manual tuning of hyperparameters. The hyperparameters of activation function, backpropagation method, and the number of hidden layers were selected based on qualitative reasons described in Section 5.1.1, whereas the number of hidden neurons, initial learning rate, L2 regularisation parameter, and maximum iterations were manually tuned based on quantitative performance, which will be explained in Section 5.1.3.

5.1.1 Fixed hyperparameters

For the activation function, the ReLU function (see Fig. 2.2) is recognised as the standard option due to its preservation of linear properties which generalise well and make gradient-based optimisation simplified [23]. It was used as it has been proven to enable more efficient and faster training of ANNs, specifically DNNs, which makes it the most popular activation function for DNNs [50].

Next, as discussed in Section 2.2, ANNs are trained via a backpropagation process, which can be implemented differently depending on the chosen algorithm. In this thesis, the Adam solver [51] was used to optimise the weight values of the ANN. Developed by Kingma and Ba [51], the Adam solver is a method for stochastic gradient-based optimisation that only requires first-order gradients with little memory requirement. Other advantages include its compatibility with non-stationary and sparse gradients and that it naturally

decreases the gradient descent step size over time. The Adam solver can be summarised using the pseudocode shown in Fig. 5.1.

Inputs: η: Initial learning rate
$\beta_1, \beta_2 \in [0,1)$: Exponential decay rates for the moment estimates
$f(w)$: Stochastic loss function with weight parameters w
ϵ: Value for numerical stability
w_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialise 1st moment vector)
$v_0 \leftarrow 0$ (Initialise 2nd moment vector)
$t \leftarrow 0$ (Initialise timestep)
while w_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_w f_t(w_{t-1})$ (Get gradients w.r.t. stochastic loss at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased 1st moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased 2nd raw moment estimate)
$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected 1st moment estimate)
$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected 2nd raw moment estimate)
$w_t \leftarrow w_{t-1} - \eta \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
end
return w_t (Resulting parameters)

Figure 5.1: Pseudocode of Adam optimisation process [51]

To minimise prediction error while preventing w from becoming too large due to overfitting [23], the loss function $f(w)$ used was square error, with ridge regression, as shown in Eq. (8) below,

$$f(w) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{n} w_j^2 \tag{8}$$

where y_i is the true output value, $\hat{y}_i$ is the predicted output value for the ith sample and α is the L2 regularisation parameter. The values for β_1, β_2 and ϵ will be fixed at the recommended default values [50] of 0.9, 0.999 and 10^{-8} respectively. η and α was optimised and the process will be explained in Section 5.1.3.

The key feature of the Adam solver is that it uses the gradient's exponential moving average m_t and the squared gradient v_t, instead of the gradient itself, to update the weight parameters. This has been proven to quicken the convergence process and smoothen the gradient descent process [51]. It is also important to note that at each timestep, the Adam solver calculates the gradient-related values using random subsamples of the provided dataset, instead of the entire dataset, thus further speeding up the computation process. This backpropagation process is also known as "training" the ANN and will be referred to as such in the rest of this thesis.

Finally, the effects of number of hidden layers are as follows [23]: increasing the number of hidden layers allows for an ANN to represent functions of increasing complexity, but also increases the risk of overfitting: a phenomenon when the ANN is fitted too closely to the training dataset and performs poorly on other validation datasets, reflecting its poor generalisation performance. Thus, the number of hidden layers was fixed at two since this generates a deep neural network which can represent a function of any shape and

there is no theoretical reason to increase this beyond two [28]. Overfitting was prevented via optimising the maximum number of iterations, which will be discussed in Section 5.1.3.

5.1.2 Background to hyperparameter optimisation

Before explaining the hyperparameter optimisation process, this subsection will first introduce key background information. To prepare for the subsequent experiments, the IFEEL dataset was split into two datasets: one for training and validation, and one for testing. The first was used to train the ANN with specific hyperparameters, following which the trained ANN was assessed for its performance using the validation set. Based on the prediction performance on the validation set, the optimal hyperparameters were selected. With the selected hyperparameters, the optimised ANN then went through sensitivity analysis, based on its predictions using the testing set. Unless otherwise stated, the main performance metrics used for validation and testing in this thesis was root mean squared error (RMSE), as shown in Eq. (9) below.

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{9}$$

Key features of RMSE are its stronger penalisation on extreme errors, since the square function amplifies errors, which highlights differences in model performance [52]. This feature is particularly useful since large errors are particularly undesirable when estimating installed PV capacity for grid operations.

In this thesis, the training/validation - testing split was performed at a ratio of 90:10 meaning 90% of the dataset was sorted into a training/validation dataset and the final 10% for testing. This split was performed using the *train_test_split* function [49]. The *random_state* parameter was set at 1 when first splitting the dataset into training/validation and testing at a ratio of 90:10 respectively. This setting allows for a reproducible dataset split. Note that this split was also performed in a stratified manner, meaning that the proportion of data samples with specific output in each dataset were equal. This ensured that the training/validation datasets did not have a disproportionate number of samples with certain output values, which might cause the trained ANN to be biased towards these values and have poor generalisation performance.

After obtaining the training/validation dataset, the input features were standardised using *StandardScaler* function [49]. This aimed to address the ANN's sensitivity to feature scaling, as mentioned in Section 2.2. The standardisation process replaced each input feature with a standardised value

$$z = \frac{x - \mu}{s} \tag{10}$$

where x is the input feature, μ is the mean and s is standard deviation. The result was a zero mean and unit variance distribution within each input feature. This scaling was performed within each of the 16 input features.

Each hyperparameter tuning process was conducted using the grid search technique, executed using the *GridSearchCV* function [49]. The grid search technique is an exhaustive search for the best performing combination of hyperparameters, from a given set of hyperparameters. The *GridSearchCV* function does this via a process called 5-fold cross-validation. This involves first randomly splitting the provided dataset into 5 subgroups (*not* in a stratified manner). Among the subgroups, 4 were used to train the ANN algorithm, which is then assessed for its RMSE performance using the final subgroup. This was then repeated such that the subgroup combination was unique at each trial, thus there were a total of 5 trials. The average RMSE value of each hyperparameter combination was recorded and the combination with the lowest RMSE value was selected.

5.1.3 Hyperparameter optimisation method

The optimised hyperparameters were, in order, number of hidden neurons in each hidden layer, initial learning rate, L2 regularisation parameter (or alpha, for short) and maximum iterations. The number of hidden neurons was optimised based on a compromise between training time duration and prediction accuracy. Increasing the number of hidden neurons allows higher representational capacity of the ANN, but this comes at the cost of exponentially increasing training time due to increased amount of computation [23], as will be visualised in Section 6.1. With the training/validation dataset as input, the mean and standard deviation of RMSE and training times were recorded using *GridSearchCV* for ANNs with number of hidden neurons from 10 to 500, in multiples of 10. Note that for this process, the other hyperparameters were fixed at the default values, i.e. initial learning rate at 0.001, alpha at 0.0001, and maximum iterations at 200. An optimal balance between low RMSE and low training time were obtained based on visual inspection of the plots of RMSE and training times against number of hidden neurons.

With the number of hidden neurons now fixed, the initial learning rate (see Fig. 5.1) and alpha value (see Eq. (8)) were optimised simultaneously using the grid search technique. This optimisation process was performed with the optimal number of hidden neurons as determined previously, with maximum iterations fixed at 200. Combinations of 3 different values of initial learning rate and alpha were first tested at each iteration. For each hyperparameter, these 3 values were the following: 1. its default value as mentioned above 2. the value which was an order of magnitude higher and 3. the value which was an order of magnitude lower below the default value. For example, the first tested learning rate values were 0.001 (default), 0.01 and 0.0001. After the first grid search, the selected hyperparameter values went through a second grid search with values higher and lower than itself within a smaller range. This process continued until the best-performing hyperparameter value remains constant for two consecutive grid searches.

Finally, the maximum number of iterations was optimised. For the Adam solver, this hyperparameter referred to the number of epochs (i.e. the number of times each data points is used). This hyperparameter was optimised based on preventing model overfitting. Overfitting can be characterised by a widening gap between training and validation error [23], as this indicates an ANN which generalises poorly and has become too specific to the training set. The optimisation process was performed as follows: the *validation_curve* function [49] was used which performs a grid search on ANNs (via 5-fold cross validation) with the hyperparameters determined previously and the maximum number of iterations tested at values from 10 to 400, in multiples of 10. At each value of maximum iterations, the function then plots average RMSE when the trained ANN makes predictions on the training data samples (training score) and on the validation data samples (validation score). The maximum iteration value was selected based on minimum RMSE value when making predictions on the validation set.

5.2 Sensitivity analysis

Once the optimal ANN hyperparameters had been determined, a sensitivity analysis of the proposed DPVSCE algorithm was conducted using ANN with optimal hyperparameters. Specifically, the prediction performance was compared between using only data samples of each season and only data samples of each intra-week group. Additionally, a sensitivity analysis to the number of household properties served by the target feeder substation was conducted. For each sensitivity analysis, a complementary analysis on PV penetration rate was also conducted. Note that for this analysis, the training/validation dataset was used simply for training and thus will be referred to as the training dataset, from here on. The performed sensitivity analysis will be explained in detail below.

5.2.1 Season

To test the DPVSCE algorithm's sensitivity to the season of the dataset, the experiment process proceeded as follows: first, data samples from the IFEEL dataset were filtered based on their season. For example, for summer, only the data samples from summer (i.e. season input feature value = 1) were used. After obtaining the filtered datasets, the training dataset was standardised (see Eq. (10)) and the scaling (mean and standard deviation of each feature) was applied to the testing dataset. This was to ensure that there was no information leak from testing dataset to the training dataset, which might have resulted in an ANN which would be biased towards the testing dataset, and thus produce unrepresentative results. Finally, using these

standardised datasets, an ANN, with the optimal hyperparameters obtained earlier, was trained, and tested for its prediction performance.

Considering possible inconsistencies due to random weight initialisation (as discussed in Section 2.2), this process was repeated five times and the performance was measured using the 3 averaged metric values across the five trials. In addition to the RMSE, two other performance metrics were used: mean absolute error (MAE), which is represented by Eq. (11) below, and standard deviation of absolute error (AE).

$$\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{11}$$

MAE serves as an additional performance metric, where high error values are not penalised as much as RMSE, while the standard deviation of AE measures the spread of the predictions' absolute error. Both metrics reflects the preciseness of the predictions: a large difference between MAE and RMSE indicates the presence of substantial error values, while a high standard deviation of AE indicates a large range of absolute error values. Note that an inaccurate (high MAE) but precise (low standard deviation of AE) algorithm would still be useful for DPVSCE as the true PV capacity value can be reliably inferred from the estimated value via the known consistent error value.

This experiment process was repeated for each seasonal group (summer, winter and transitional seasons), and for the entire dataset (all seasons). Note that for the season-specific experiments, the season input feature was not used, since it was constant throughout the filtered dataset and offered no distinguishing value between samples. This means there were only 15 input features in each sample for season-specific experiments. In real terms, these experiments tested the model's performance if feeder net-load data had only been collected during specific seasons and offered an insight into the value of having a separate algorithm for each season.

5.2.2 Intra-week group

The entire experimentation process described in Section 5.2.1 was then repeated, except instead of filtering the dataset based on season, samples from specific intra-week groups were used. Similarly, the intra-week group input feature was removed for the tests with filtered IFEEL datasets. This intra-week group analysis served to test the effectiveness of the proposed intra-week groups in extracting knowledge on the household load consumption behaviour.

5.2.3 PV penetration rate

In addition to recording the average RMSE, MAE and standard deviation of AE for sensitivity analysis to season and intra-week group, the average and standard deviation of prediction RMSE at each true DPVSC value is recorded and plotted for each season and intra-week group. This creates a residual plot which depicts the variation of algorithm performance in response to varying PV penetration rates. In real terms, this tests the proposed algorithm's prediction performance for substation feeders which serve residential neighbourhoods with different amount of DPVS installed.

5.2.4 Number of households

The final sensitivity analysis of the proposed DPVSCE algorithm will be performed based on varying number of household properties served by the target substation feeder. With different neighbourhood sizes in the UK, this sensitivity analysis aims to assess the model's effectiveness for substation feeders which serve a different number of households. The pseudocode for this sensitivity analysis is shown in Fig. 5.2 below.

```
1  Input: 81 individual household load consumption datasets (annual)
2  Input: PV generation dataset (annual)

3  For no. of household properties (20 to 80, in multiples of 10):
4      For number of DPVS installed (0 to no. of household properties):
5          Randomly select the specified number of individual household load consumption dataset
6          Add PV generation of specific number of installed DPVS to load consumption
7          Obtain annual feeder-level net-load dataset
8          Extract averaged 24-hour net-load datasets
9          Extract IFEEL input features
10     Obtain dataset of samples with various output PV capacity value, and same number of households
11     (Fixed) split dataset into training and testing datasets in a stratified manner
12     Select data samples based on season or intra-week group (if necessary)
13     Standardise training and testing datasets
14     Train ANN using training dataset
15     Test ANN using testing dataset and record RMSE of each true output value
16 Return m × n array of RMSE values, where m is number of households and n is PV penetration rate
```

Figure 5.2: Pseudocode for sensitivity analysis to number of household properties, with lines in italics being implicit checkpoints.

For this thesis, the number of household properties tested will be from 20 to 80, in multiples of 10, and the tested number of DPVS installed will be from zero to the total number of household properties. The code shown in Fig. 5.2 will be repeated five times and the final product will be a heatmap with the axes of number of households against PV penetration rate, with depth as the average RMSE value. This experiment process will also be repeated using only data samples of specific seasons and of specific intra-week groups respectively. Ultimately, this provides a comprehensive overview of the proposed algorithm's performance in terms of 4 variables: PV penetration rate, number of household properties, using season-specific data samples and using intra-week group specific data samples.

6 Results and Discussion

Based on the methodology detailed in Sections 5.1, this section will first present the results of the hyperparameter optimisation process and explain the rationale behind each optimal hyperparameter value. Then, based on the methods explained in Section 5.2, the results of sensitivity analysis will be presented and their implications on real world application of the proposed DPVSCE method will subsequently be discussed.

6.1 Hyperparameter optimisation results

The results of the each hyperparameter optimisation process detailed in Section 5.1.3 will now be presented and discussed, followed by a summary of optimal hyperparameters.

6.1.1 Number of hidden neurons

The first hyperparameter optimised was the number of neurons in each hidden layer. Fig. 6.1 shows the plot of average training time (seconds) and RMSE (kW) against the number of neurons in each hidden layer. The standard deviations of training time and RMSE are also plotted as a range around each point on the curve. From the figure, both training times and RMSE are within a tight range for each number of hidden neurons, and the anticipated trend of increasing accuracy with the cost of increasing training time, as the number of hidden neurons increase, is reflected clearly.

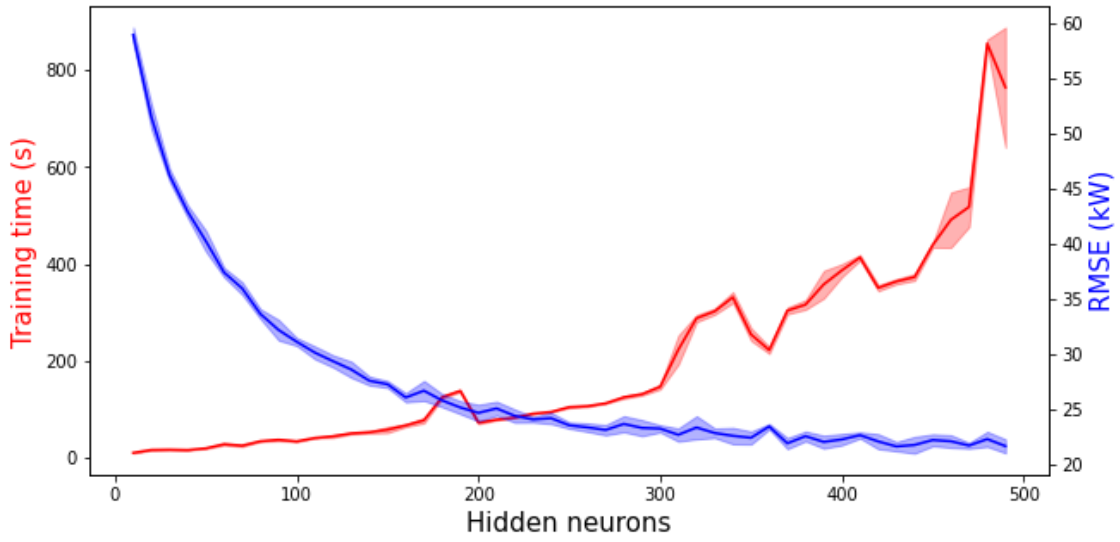


Figure 6.1: Plots of average training time and RMSE against number of hidden neurons, with standard deviation depicted as a range around the average values

While the objective was to minimise RMSE, the cost of training time must be taken into consideration for practical application since the ANN would be constantly updated with new data, as discussed in Section 2.2, and it would be impractical for the ANN to take longer to update itself than it receives new data.

Additionally, Fig. 6.1 shows that as the number of hidden neurons increase, training time begins to increase exponentially whereas the rate of decrease of RMSE reduces. This reflects an increasingly disadvantageous trade-off of training time for prediction accuracy, as such, the number of hidden neurons will be selected at a point where RMSE value begins to flatten, and training time begins to rise sharply.

From Fig. 6.1, a clear spike in training time is observed at more than 300 neurons per hidden layer, while the RMSE value has stayed unchanged from 270 neurons per hidden layer. Thus, the optimal number of neurons in each hidden layer was set at 270, where the average RMSE is 23.2 kW and average training time is 113 seconds.

6.1.2 Initial learning rate/Alpha

The results of the optimisation process for the initial learning rate and alpha values are presented in Table 6.1 below, where the tested hyperparameter values are listed and the best performing values are underlined.

Table 6.1: Progression of the iterative process of optimising learning rate and alpha values, with the underlined values being the best-performing

Iteration Number	Initial learning rate	Alpha (in 10^{-5})
1	**0.01**, 0.001, 0.0001	**100**, 10, 1
2	0.1, **0.01**, 0.005	1000, 100, **50**
3	0.05, **0.01**, 0.0075	75, **50**, 25

Note that the initial learning rate has already been optimised by the second iteration, since the best performing value has remained constant for two iterations. Despite this, a third iteration was performed as the alpha value was still being optimised. This additional iteration further corroborates the optimality of 0.01 for the initial learning rate and thus, was selected as the optimal value, with 50×10^{-5} as the optimal alpha value.

6.1.3 Maximum iterations

The final hyperparameter to be optimised was the maximum number of iterations during the ANN training process. Fig. 6.2 shows the average training and validation RMSE against the maximum number of iterations. As expected, the ANN model performed better on the training samples than the validation samples, given that it was trained using the latter and so, it has "learnt" the relationship between the input features and the output variable well. Prediction accuracy is also observed to decrease in a logarithmic fashion as maximum

iterations increase, although it stagnates at a RMSE value of around 15 kW at more than 150 max iterations. This makes any iterations more than 150 times unnecessary, given its miniscule amount of error reduction.

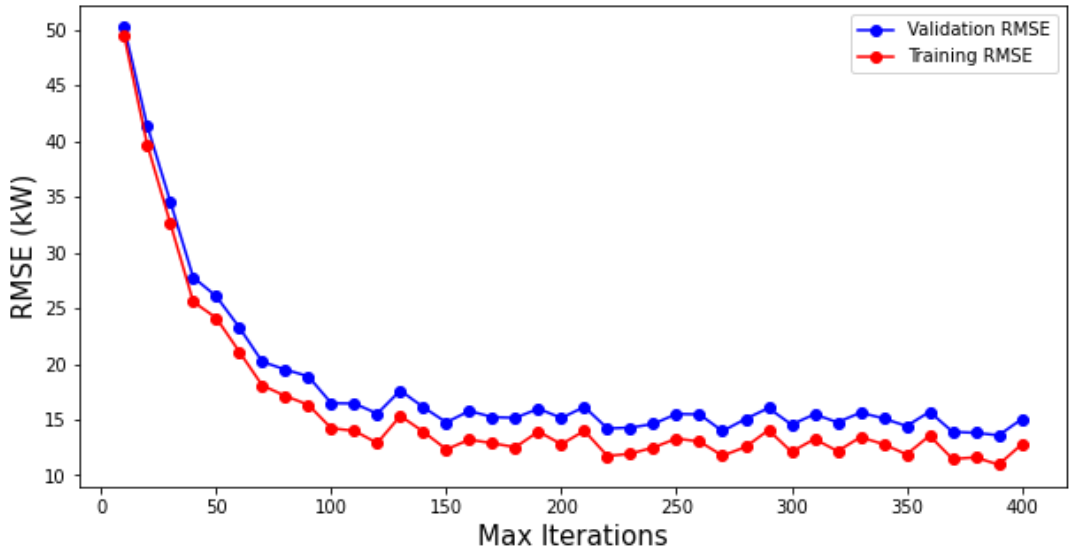


Figure 6.2: Plots of training and validation RMSE against max training iterations

To make a more informed decision on this value, the training time will be taken into consideration. To do this, the mean and standard deviation of training times are plotted against the maximum number of iterations in Fig. 6.3 below.

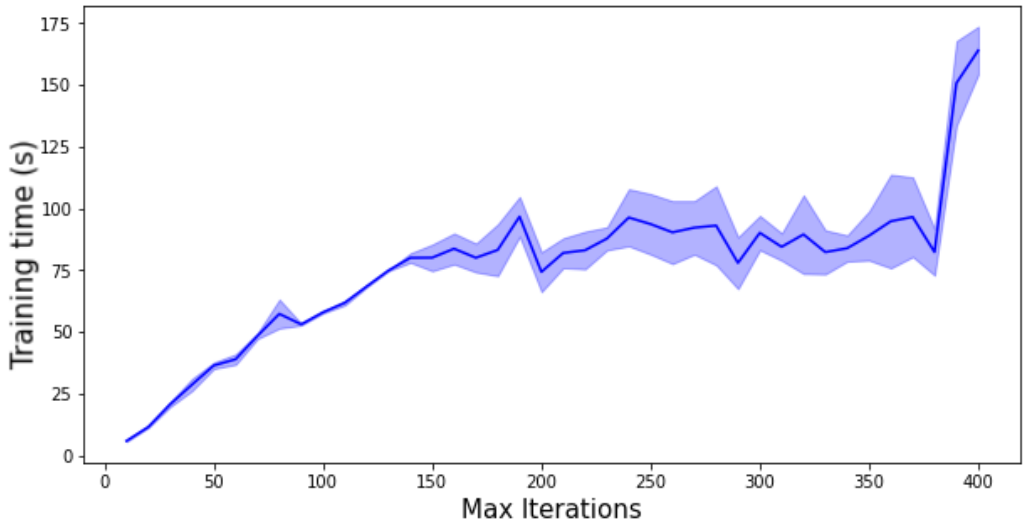


Figure 6.3: Plot of average training time against maximum number of iterations, with standard deviation depicted as a range around the average values

The aim was to minimise both RMSE and training time, with priority placed on minimising the latter. From Fig. 6.3, it can be observed that training time is within the range of 75 to 100 seconds when the maximum number of iterations go beyond 150 up to 370. At 380 maximum iterations, the training time nearly doubles to 150 seconds, while RMSE decreases minimally. Thus, the number of iterations should be capped at a value

between 150 and 380 and chosen based on minimum validation RMSE value. As such, the maximum number of iterations was set at 370.

6.1.4 Finalised hyperparameters

The optimal ANN hyperparameters are listed in Table 6.2 below. These custom hyperparameter values were used for the subsequent experiments, with all other unmentioned hyperparameters fixed at the default values provided by the scikit-learn package [49].

Table 6.2: Finalised ANN hyperparameter type or values used for sensitivity analysis experiments

	Hyperparameter	Value
Fixed	Activation function	ReLU
	Backpropagation method	Adam Solver
	Number of hidden layers	2
Optimised	Number of hidden neurons (in each layer)	270
	Initial learning rate	0.01
	L2 regularisation parameter (Alpha)	50×10^{-5}
	Max iterations	370

6.2 Sensitivity analysis results

In this subsection, the results of the sensitivity analysis procedures, as described in Section 5.2 are presented and discussed. For each set of results, key observations will be pointed out and each will be accompanied with its reasons and its implication on real-life applications.

6.2.1 Season

The first analysis involved only using data samples from specific seasons, namely summer, winter and the transitional seasons (spring and autumn). The average RMSE, MAE and standard deviation of AE of each circumstance is shown in Table 6.3 below.

Table 6.3: Prediction performance based on different metrics using only data samples from specific seasons

Season / Metric	**RMSE** (kW)	**MAE** (kW)	**Std of AE** (kW)
Summer	13.8	8.66	10.7
Winter	27.3	19.9	18.8
Transitional	12.8	9.07	9.05
All	13.0	8.90	9.51

Two key observations can be made from Table 6.3: first, using only data samples based in the winter months clearly produced the worst results across all 3 metrics, while the other three seasonal groups' performance were relatively close to one another. Secondly, comparing between summer, transitional seasons and all data, RMSE and standard deviation of AE increased, while MAE decreased in the order of transitional seasons, all data and summer. This means that using only data samples from summer produces the most accurate predictions, with a cost of reduced preciseness, and vice versa for the transitional seasons, whereas using data samples across all four seasons achieves a balance between the two.

The first observation can be explained by the low amounts of PV generation during winter, resulting in less obvious distinctions between net-load curves of different installed PV capacities. This is reflected in Fig. 4.3, where the net-load behaviour based in winter changed the least as PV penetration rate increased. These characteristics implied a poorer prediction performance during winter. Thus, DNOs are strongly encouraged to plan data collection in advance if DPVSCE during winter (using data samples from winter) is unavoidable, so that they have data from all seasonal groups to train a more accurate and precise ANN for winter predictions. An example of time-sensitive DPVSCE could be a sudden deployment of government-subsidised DPVS installations, coincidentally during winter, which would lead to grid malfunctions if unaccounted for.

Based on the logic of the first observations' explanation, using only data samples from summer should perform better than that of transitional seasons, since there is the largest amount of PV generation during summer. However, this is not reflected in the results shown in Table 6.3. The transitional seasons' unexpectedly desirable performance could be attributed to the fact that there were about twice as many data samples from transitional seasons (6 months of data) as that from summer or winter (3 months of data), which allowed for the ANN to better learn the relationship between the input and output variables. To test this hypothesis, the experiment process described in Section 5.2.1 was repeated using a half of the training and validation data samples (selected randomly) from transitional seasons and the performance results are presented in Table 6.4 below.

Table 6.4: Prediction performance based on different metrics using half of the available data samples from transitional seasons

RMSE (kW)	**MAE** (kW)	**Std of AE** (kW)
24.0	15.8	18.0

With equal number of training and testing samples, the performance of using samples based in transitional seasons now sat between summer and winter across all three metrics, supporting the explanation that the difference in PV generation between seasons causes the difference in prediction performance. This also suggests that increasing the number of data samples used for training can greatly improve prediction

performance, which will be especially useful for predictions in winter. As such, DNOs may wish to allocate more resources into data collection and storage for a large pool of training data or collaborate with each other to share relevant feeder level net-load data. However, if the available data is not enough to generate an accurate and precise predictor for all seasons, DNOs are advised to plan DPVSCE to happen only in summer, so that optimal performance can be achieved with less data. In fact, the results showed that using only summer-based data produce comparable performance to using data from all year round, despite the former requiring only a third of data quantity.

6.2.2 Intra-week groups

The next analysis involved only using data samples from specific intra-week groups. The average RMSE, MAE and standard deviation of AE of each circumstance are shown in Table 6.5 below.

Table 6.5: Prediction performance based on different metrics using only data samples from specific intra-week groups

Intra-week Group / Metric	**RMSE** (kW)	**MAE** (kW)	**Std of AE** (kW)
Mon-Thurs (A)	12.2	8.65	8.6
Fri-Sat (B)	14.8	10.5	10.5
Sun (C)	16.5	11.1	12.2
All	13.0	8.90	9.51

For simplicity, the intra-week groups will be referred to as the bracketed letter shown in Table 6.5. Two key observations can be made from the results: first, using only data samples from Group A produced the best results across all 3 metrics, compared to using all samples as presented in both Table 6.5 and 6.3. Secondly, comparing between the three intra-week groups, all 3 metric values increased, reflecting a reduction in both accuracy and precision, in the order of Groups A, B and C. The first observation leads to the straightforward implication that priority should be placed on collecting and storing feeder-level net-load data from Monday to Thursday, so that they can be used to develop the most accurate and precise deep learning-based DPVSCE algorithm.

There are two hypotheses which could explain the second observation. The most straightforward explanation would be that household consumption behaviour is the most consistent during the days of Group A, followed by Groups B and C, which allowed the differences in net-load curve to be solely based on differences in DPVSC and PV generation. The second hypothesis, which involves a factor which might be overlooked, is that the amount of data which constitutes each data sample affects prediction performance. Specifically, referring to Fig. 3.1, IFEEL features obtained from a Group A net-load curve were the average of 4 days' worth of data, while a Group B net-load curve was the average of only 2 days' worth of data and

just one day for Group C. Considering this factor, the results could reflect the effectiveness of averaging 24-hour net-load curves to reduce the daily variability of weather conditions, as mentioned in Section 4.1.

To further support this hypothesis, more in-depth research can be conducted by maintaining the intra-week group variable constant and comparing prediction performance when averaged net-load curves are obtained from different amount of data. For example, 3 IFEEL datasets can be created using IFEEL features extracted from Group A averaged net-load curves of 4 days, 8 days and 12 days respectively. Prediction performance will then be assessed for these 3 circumstances. This will provide more conclusive results on the effectiveness of removing noise by taking the average of more 24-hour net-load datasets.

6.2.3 PV penetration rate

After analysing the general accuracy and precision of the proposed DPVSCE algorithm across all PV penetration rates in a service area, this subsection will present and discuss the variation in prediction performance in response to different PV penetration rates (or installed DPVSC). The residual plots of the average and standard deviation of prediction RMSE against PV penetration rate, when using only data samples of specific seasonal or intra-week groups, are shown in Fig. 6.4 below. Note that the axes labels are constant throughout all plots.

Three key observations can be obtained from the plots shown in Fig. 6.4. Firstly, a common pattern emerged in all plots: RMSE values tend to be higher and more inconsistent at PV penetration levels below 50%, compared to above 50%. Secondly, huge spikes in RMSE tend to appear at 100% PV penetration rates, as seen in 5 out of 7 plots. Finally, the results presented in Tables 6.3 and 6.5 are consistent with plots in Fig. 6.4, where a lower curve on the y-axis implies more accurate predictions and the presence of sharp spikes and dips implies imprecise predictions. For example, comparing between summer and transitional seasons plots in Fig. 6.4, the RMSE values for summer were generally lower than that for transitional seasons, but have more extreme values, representing its higher accuracy but lower precision.

The first observation can be explained with the low PV generation at low PV penetration rates. With lower PV generation, it was easier for inconsistent weather patterns or load consumption behaviour to be mistaken by the algorithm as changes in installed DPVS capacity. For example, the holiday season could lead to a significant spike in household load consumption which appears similar to a reduction in DPVSC at low PV capacities, leading the algorithm to underestimate the PV capacity. At high PV capacities, however, such a consumption spike might be minimal compared to the amount of PV generated, and thus would not be detected by the DPVSCE algorithm.

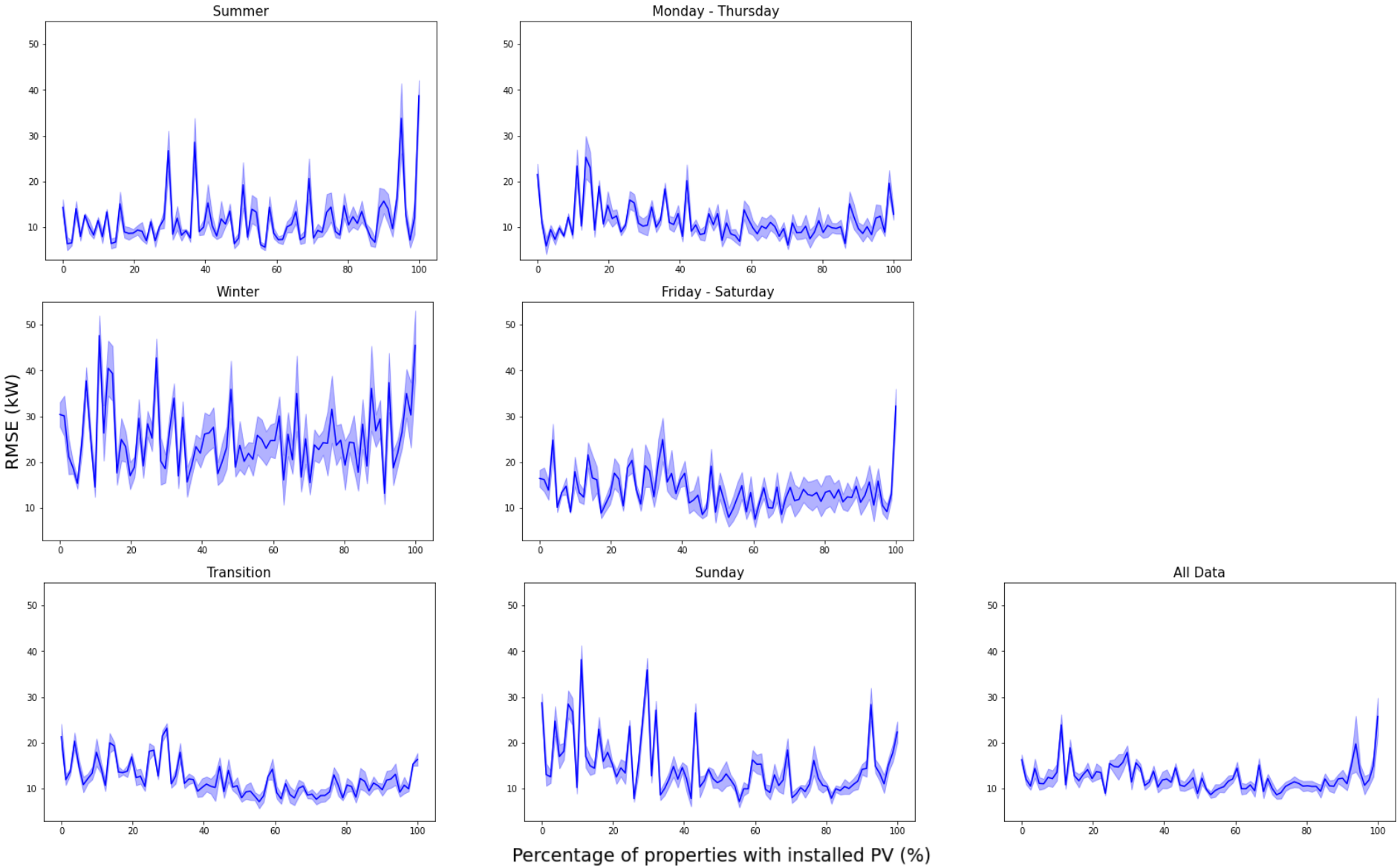


Figure 6.4: Average root mean squared errors of prediction against percentage of properties with installed PV for ANNs using all data samples and data samples from specific seasons and intra-week groups, with standard deviation depicted as a range around the average values

In real terms, with only 3.3% of UK households having installed solar panels in 2020 [53], DNOs may wish to employ the proposed DPVSCE method on newer estates with higher PV penetration rates for more accurate predictions or delay the deployment of the proposed method as PV penetration rates continue to increase throughout the UK in the near future. Otherwise, obtaining an initial estimate of the PV penetration rate may be useful in determining the expected error range to decide if the proposed method's estimation can be utilised. This initial estimate could be performed either using satellite imagery or in-person surveying, though the latter might be resource intensive.

From the second observation, the next step is to understand if the highly erroneous predictions are specific to a true PV capacity of 372.6 kW or if they are specific to the condition of a 100% PV penetration rate. This is explored in Section 6.2.4 where the prediction performance of different number of households is measured. The varying number of households results in different PV capacities at 100% PV penetration rate.

6.2.4 Number of households

The final set of experiment involved assessing the accuracy of the proposed DPVSCE method when used on substation feeders which serve different numbers of household. The average RMSE values with varying number of households and PV penetration rate are presented in Fig. 6.5, with each heatmap representing using only data samples from specific seasons or intra-week groups. Note that the colour scale of each heatmap is constant, with black corresponding to a value of 40 kW, hotter colours being more than 40 kW and cooler colours being less than 40 kW. Each heatmap also has a colour scale which indicates the range of values present in the heatmap.

Three key observations can be made from Fig. 6.5. These observations are common patterns which emerge in all heatmaps in Fig. 6.5. Firstly, following from the previous subsection, the phenomenon of highly erroneous predictions at near 100% PV penetration is observed across different number of households. Combined with the high RMSE values generally seen at near 0% PV penetration, this can be visualised as a quadratic curve when plotting RMSE against PV penetration rate (see Fig. 6.4). Secondly, there is a general trend of increasing RMSE values as number of households increases. Finally, despite the trend mentioned previously, RMSE drops sharply from 70 to 80 households served by the substation feeder.

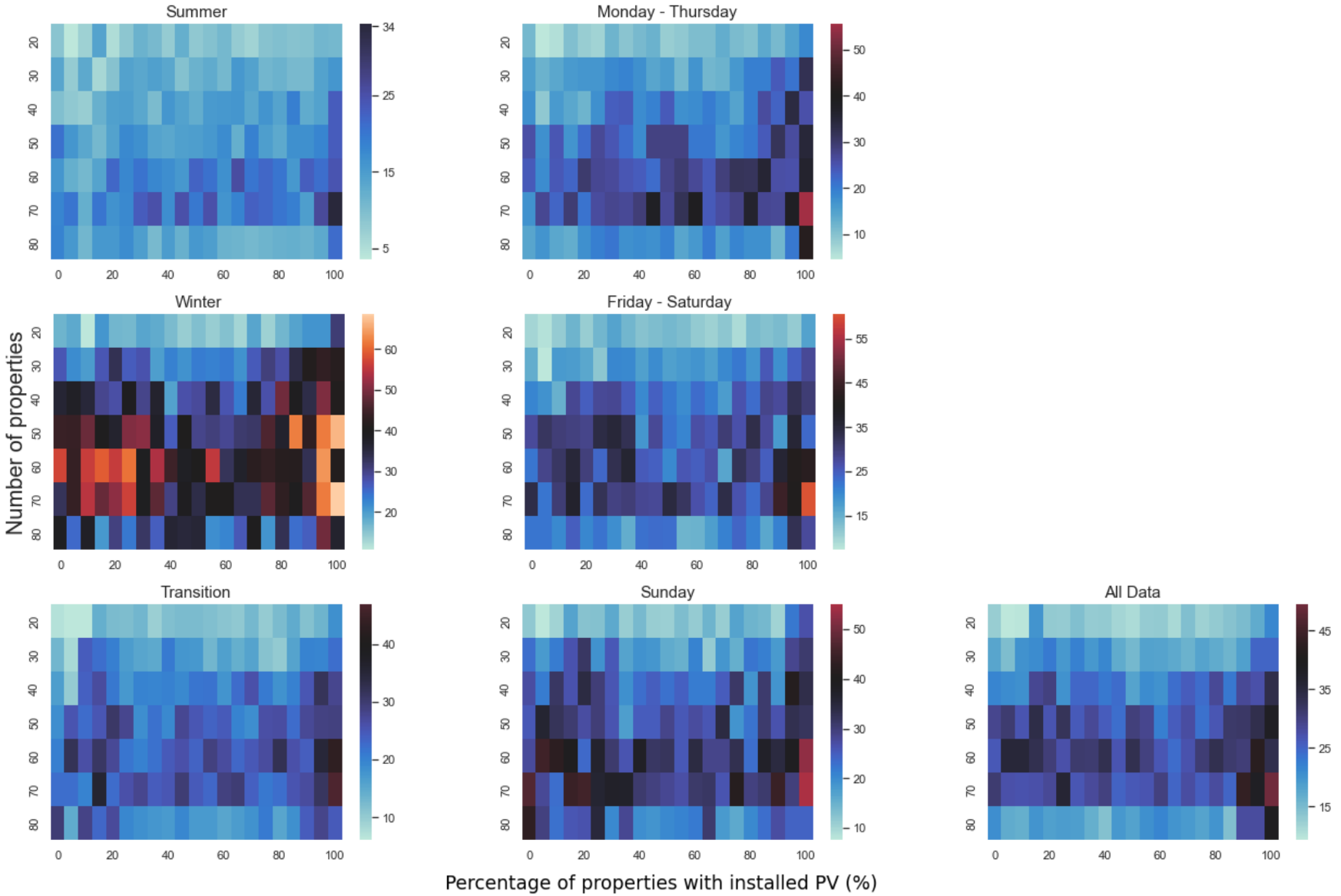


Figure 6.5: Average root mean squared error of predictions varying with number of properties and percentage of properties with installed PV (based on true value of installed PV capacity), with the title of each heatmap indicating the season or intra-week group of the data samples used

The first observation confirms that the extreme error at high PV penetration rate was not tied to the absolute value of the installed PV capacity, but rather the PV penetration rate. This experiment results can be interpreted as an issue of non-linear residual plot, which can usually be fixed either by transforming an input feature or adding a "missing" feature [54]. A possible input feature would be the number of households or the "theoretical" maximum DPVSC based on the number of households. Further research can be conducted to explore this alternative. Regardless, a high PV penetration rate at more than 80% is unrealistic as the costs of integrating that much DPVS into the grid is predicted to suppress PV penetration to roughly 60% [55], thus, this observed limitation of the proposed model will unlikely become a concern in real-life application.

The second observation could be explained by a possible cause-effect relationship between true PV capacity and the margin of error, where an increase in the former leads to an increase in the latter. To test this hypothesis, the mean absolute percentage error (MAPE), represented by Eq. 14 below, is plotted as a heatmap in Fig. 6.6 below, with MAPE in fractional form (unitless)

$$\frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|}{y_i} \tag{12}$$

Fig. 6.6 presents prediction error in proportion to the true DPVSC value, with the colour scale constant where black represents the value of 1.5, with hotter colours representing values more than 1.5 and vice versa. It can be seen that the error-to-true value ratio stayed relatively constant for all number of households at PV penetration rate of 40% and above, while at below 40% PV penetration, MAPE tends to increase with decreasing number of households and decreasing PV penetration rate. With the highest MAPE at around 0% to 10% PV penetration rate, this result continues to encourage the use of the proposed method for newly built estates with more installed PV or in the future when PV penetration rate has risen. With more homogeneous MAPE values at about 0.1 when using data from specific time (see Fig. 6.6), this provides a useful guide for expected error margins in predictions. Based on this guide and acceptable error margins, DNOs can then decide on the suitable substations to use this method on, depending on the their estimated number of households and PV penetration rate.

Finally, the third observation is likely to be the result of overfitting the hyperparameters to the specific circumstance of 80 households, causing the algorithm to perform poorly when the number of households is different. To tackle this, DNOs can employ the proposed method by either training separate predictors for feeders serving different number of households, or include data samples corresponding to various number of households in the training process, to create a predictor which is able to generalise well.

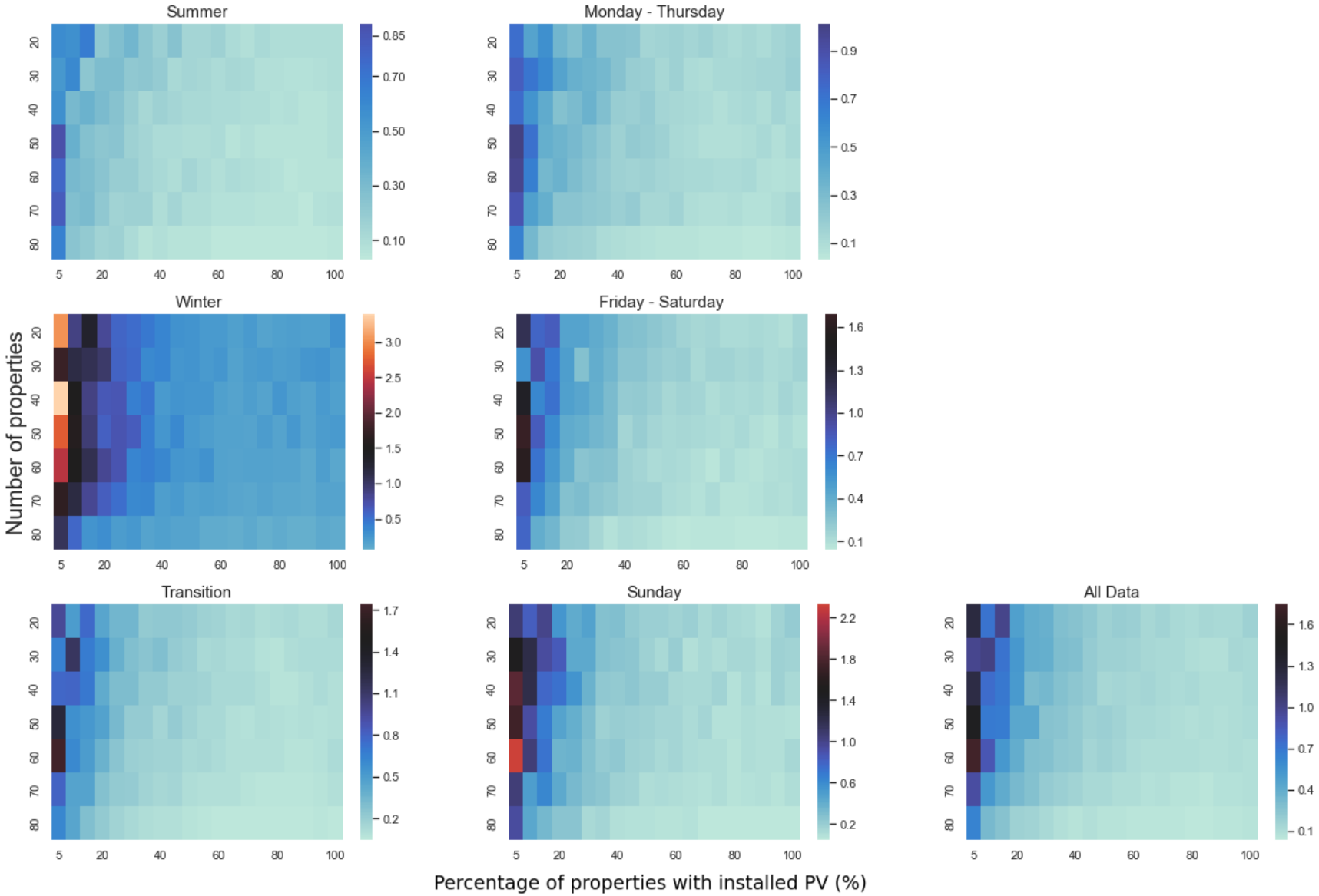


Figure 6.6: Average mean average percentage error (in fractional form) of predictions varying with number of properties and percentage of properties with installed PV (based on true value of installed PV capacity), with the title of each heatmap indicating the season or intra-week group of the data samples used

6.3 Limitations of results

To conclude the analysis results, this subsection will explain their limitations which readers must be aware of. These limitations can also be interpreted as areas of improvement if this project were to be conducted again. These limitations either arise due to the lack of suitable data, or were discovered during the course of this project, but could not be fixed due to time constraints.

The main limitation of this thesis was the lack of real-life data. As explained in Section 3.4, real feeder net-load datasets, with known levels of installed DPVS, were to be obtained as ground truth data. However, they did not arrive in time to be used for this thesis. Thus, the results are limited by the assumption that the simulated PV generation reflect real-life generation. Unfortunately, real DPVS parameters will not be as homogeneous as the simulated data parameters. For example, real household rooftops would be of different areas and slopes, or PV panels installed would be of more varied efficiencies and azimuthal positions. With more time, using real data to test the model will improve the results' reliability when considering industrial application.

Another limitation related to data was the use of outdated datasets. The use of household load consumption data from 2013 assumed that residential electricity consumption patterns had remain unchanged since then. Unfortunately, this assumption has lost credibility since the COVID-19 pandemic had led to changes in household electricity consumption [56]. For example, electricity consumption has generally increased, and load peaks increased by 15-20%. Thus, this might imply a necessary restriction to only use post-pandemic data to train the proposed model.

Referring to Section 3.3, the PV generation dataset's conversion from a time resolution of hourly to half-hourly assumed that power generation levels stayed constant within each hour. To reflect a more realistic gradual increase/decrease of generation levels, linear interpolation can be performed, where each half-hour timestamp will have the average value of half-hour before and after. For example, a 1 kW value at 09:00 and 2 kW value at 10:00 would lead to an additional 1.5 kW value at 09:30.

Finally, the trained model assessed in this thesis was trained using a dataset which assumed that the substation feeder will be connected to 81 households, resulting in the overfitting to this specific circumstance, as described in Section 6.2.4. Thus, for a more accurate assessment of the model's sensitivity to the number of households in the future, the number of households can be an additional input feature, and the model should be trained using data samples with varying values of this feature.

7 Conclusion and next steps

Increasing installations of behind-the-meter DPVS have led to the need for updated information on installed DPVSC to ensure the continued smooth operations of the power network. However, currently researched DPVSCE methods face a major limitation of having extensive data requirements, whether they are satellite images, household-level net-load datasets, or weather data. The proposed method developed in this thesis overcomes this limitation by employing the data pre-processing technique of extracting time invariant IFEEL features from averaged 24-hour net-load curves. Thus, this model is highly accessible for DNOs or other relevant stakeholders to use since it only requires feeder level net-load time-series data of any time resolution.

In this thesis, extensive analysis was conducted to assess the proposed method's sensitivity to changing circumstances, from which a few key takeaways are to be noted. First, Section 6.2.1 have shown that the amount of data will be crucial to improving prediction performance, with more training data samples leading to more accurate and precise estimations. In this case, doubling the number of data samples from about 6,396 to 12,792 saw a reduction in average RMSE from 24.0 kW to 12.8 kW for the model trained using only transitional season data. To put this into perspective, 40 PV-equipped households (~50% PV penetration in an 81-household substation) will have a total estimated DPVSC of 184 kWp, based on the assumptions explained in Section 3.2. As such, doubling data samples will reduce RMSE from being 13% to 7% of the true value. Thus, DNOs should place emphasis on data collection and storage, to obtain an extensive training dataset. However, if faced with constraints, DNOs are advised to prioritise developing a summer exclusive DPVSCE model. Despite only using about 6,396 training samples for the summer-based model, compared to 25,584 samples for the model trained with all data, the former performed comparably with the latter, with RMSE values of 13.8 kW and 13.0 kW respectively, and MAE values of 8.66 kW and 8.90 kW respectively.

Next, Section 6.2.2 revealed that for models trained and tested exclusively using specific intra-week groups, using data from Mondays to Thursdays performed the best with an average RMSE of 12.2 kW, followed by Fridays to Saturdays with 14.8 kW and Sundays with 16.5 kW. These results could be caused by either consumers exhibiting the most predictable consumption patterns from Mondays to Thursdays, or the effectiveness of taking the average daily net-load curve of a longer time-series, or both. The exact contributing factor can be determined through further research. Finally, the proposed method displayed weakness in making estimations at low PV penetration rates and at low power consumption (when there are fewer household properties), where high percentage errors are observed. Using the summer-based model as an example, MAPE values at 5% PV penetration rate were generally twice of that at 10% PV penetration rate, before stabilising at

about 0.1 to 0.2 at higher PV penetration. Also, the MAPE value only stabilised at roughly 0.1 at 60% PV penetration, for 20 households, while the MAPE value for 70 households dropped much more rapidly. With the residential PV adoption rate at just 3.3% in 2020 [53], the use of the proposed method is encouraged at bigger estates, or at a later time when there are more DPVS installed. After all, Section 6.2.4 showed that MAPE values stay relatively constant at higher number of households or higher PV penetration rates. This is useful as an error margin guide for DNOs to make a more informed decision on the utility of the proposed DPVSCE method based on the estimated number of households and PV penetration rate served by the substation feeder.

Moving forward from this thesis, there are several research areas which can be further explored. Firstly, further research can be conducted to assess the effectiveness of removing net-load curve noise via taking the average of data from more days. An example research process was explained in Section 6.2.2. If using more data to obtain averaged net-load curves is indeed more effective, note that this will incur the cost of reduced training samples, given a constant amount of time-series data available. This cost might, in turn, lead to worsened prediction performance, based on the result of Section 6.2.1. Thus, this leads to a second potential research area of exploring the trade-off between number of training samples and the amount of data used per sample. Finally, the rising popularity of energy storage tagged with residential PV systems will lead to significant changes in net-load curve behaviour due to the load shifting effects of the battery charging during peak PV production and battery discharging during peak electricity consumption [57]. Thus, further research in updating the proposed DPVSCE method to incorporate the effects of energy storage when analysing feeder net-load curves will be necessary.

Finally, for these results to be utilised and make real impact, the work presented in this thesis will be publicised in various forms to reach relevant stakeholders, such as DNOs like SSEN, or energy technology companies like Eneida. A research poster based on this thesis will be presented at the upcoming opening ceremony of The Energy Systems Accelerator, or TESA, a think tank workspace for key stakeholders in achieving net-zero, in May 2022, and a paper will be produced and submitted to a journal with high impact factor (e.g. Applied Energy) for publication.

Bibliography

[1] International Energy Agency. (n.d.). Solar PV – Renewables 2020 – Analysis. Renewables 2020 - Analysis and Key Findings. A Report by the International Energy Agency. Retrieved 9 February 2022, from https://www.iea.org/reports/renewables-2020/solar-pv

[2] International Renewable Energy Agency. (2019). Future of Solar Photovoltaic: Deployment, investment, technology, grid integration and socio-economic aspects. Retrieved 18 November 2021, from https://irena.org/-/media/Files/IRENA/Agency/Publication/2019/Nov/IRENA_Future_of_Solar_PV_2019.pdf

[3] European Commission. (2017). Study on "Residential Prosumers in the European Energy Union". Retrieved 9 February 2022, from https://ec.europa.eu/info/sites/default/files/study-residential-prosumers-energy-union_en.pdf

[4] Department for Business, Energy & Industrial Strategy. (2022). Solar photovoltaics deployment. Retrieved 2 March 2022, from https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fassets.publishing.service.gov.uk%2Fgovernment%2Fuploads%2Fsystem%2Fuploads%2Fattachment_data%2Ffile%2F1056130%2FSolar_photovoltaics_deployment_January_2022.xlsx&wdOrigin=BROWSELINK

[5] Planning Portal. (n.d.). Planning Permission. Retrieved 23 February 2022, from https://www.planningportal.co.uk/permission/common-projects/solar-panels/planning-permission

[6] Planning Portal. (n.d.). Planning Permission: Solar equipment mounted on a house or a block of flats or on a building within the curtilage. Retrieved 23 February 2022, from https://www.planningportal.co.uk/permission/common-projects/solar-panels/planning-permission-solar-equipment-mounted-on-a-house-or-a-block-of-flats-or-on-a-building

[7] Waswa, L., Chihota, M. J., & Bekker, B. (2021). A Probabilistic Estimation of PV Capacity in Distribution Networks From Aggregated Net-Load Data. IEEE Access, 9, 140358–140371. https://doi.org/10.1109/ACCESS.2021.3119467

[8] Zhang, X., & Grijalva, S. (2016). A Data-Driven Approach for Detection and Estimation of Residential PV Installations. IEEE Transactions on Smart Grid, 7(5), 2477–2485. https://doi.org/10.1109/TSG.2016.2555906

[9] Wang, F., Li, K., Wang, X., Jiang, L., Ren, J., Mi, Z., Shafie-khah, M., & Catalão, J. (2018). A Distributed PV System Capacity Estimation Approach Based on Support Vector Machine with Customer Net Load Curve Features. Energies, 11(7), 1750. https://doi.org/10.3390/en11071750

[10] Li, K., Wang, F., Mi, Z., Fotuhi-Firuzabad, M., Duić, N., & Wang, T. (2019). Capacity and output power estimation approach of individual behind-the-meter distributed photovoltaic system for demand response baseline estimation. Applied Energy, 253, 113595. https://doi.org/10.1016/j.apenergy.2019.113595

[11] Massachusetts Institute of Technology. (2020). Researchers find benefits of solar photovoltaics outweigh costs. 24 February 2022, from https://news.mit.edu/2020/researchers-find-solar-photovoltaics-benefits-outweigh-costs-0623

[12] Hu, M., Ge, D., Telford, R., Stephen, B., & Wallom, D. C. H. (2021). Classification and characterization of intra-day load curves of PV and non-PV households using interpretable feature extraction and feature-based clustering. Sustainable Cities and Society, 75, 103380. https://doi.org/10.1016/j.scs.2021.103380

[13] Charabi, Y., Rhouma, M. B. H., & Gastli, A. (2010). GIS-based estimation of roof-PV capacity amp; energy production for the Seeb region in Oman. 2010 IEEE International Energy Conference, 41–44. https://doi.org/10.1109/ENERGYCON.2010.5771717

[14] Malof, J. M., Bradbury, K., Collins, L. M., & Newell, R. G. (2016). Automatic detection of solar photovoltaic arrays in high resolution aerial imagery. Applied Energy, 183, 229–240. https://doi.org/10.1016/j.apenergy.2016.08.191

[15] Scottish and Southern Electricity Networks. (2020). Smart Meter Data Privacy Plan. Retrieved 02 March 2022, from https://www.ofgem.gov.uk/sites/default/files/docs/2020/05/ssen_smart_meter_data_privacy_plan_redacted_final.pdf

[16] Wallom, D. (2022). Personal Communication.

[17] Corti, L., Bishop, L. & Elam S. (n.d.). Legal and ethical challenges surrounding big data: energy data. Retrieved 10 February 2022, from ukds-case-studies-ethical.pdf (ukdataservice.ac.uk)

[18] Sossan, F., Nespoli, L., Medici, V., & Paolone, M. (2018). Unsupervised Disaggregation of Photovoltaic Production From Composite Power Flow Measurements of Heterogeneous Prosumers. IEEE Transactions on Industrial Informatics, 14(9), 3904–3913. https://doi.org/10.1109/TII.2018.2791932

[19] Saeedi, R., Sadanandan, S. K., Srivastava, A. K., Davies, K. L., & Gebremedhin, A. H. (2021). An Adaptive Machine Learning Framework for Behind-the-Meter Load/PV Disaggregation. IEEE Transactions on Industrial Informatics, 17(10), 7060–7069. https://doi.org/10.1109/TII.2021.3060898

[20] Zhang, X.-Y., Kuenzel, S., & Watkins, C. (2020). Feeder-Level Deep Learning-based Photovoltaic Penetration Estimation Scheme. 2020 12th IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), 1–5. https://doi.org/10.1109/APPEEC48164.2020.9220536

[21] Zhang, X.-Y., Watkins, C., & Kuenzel, S. (2021). Multi-Quantile Recurrent Neural Network for Feeder-Level Probabilistic Energy Disaggregation Considering Roof-Top Solar Energy. https://doi.org/10.36227/techrxiv.16569735.v1

[22] Carbon Brief. (2021). Mapped: How climate change affects extreme weather around the world. Retrieved 03 March 2022, from https://www.carbonbrief.org/mapped-how-climate-change-affects-extreme-weather-around-the-world

[23] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. http://ebookcentral.proquest.com/lib/oxford/detail.action?docID=6287197

[24] N.J. Sairamya, L. Susmitha, S. Thomas George, M.S.P. Subathra. (2019). Chapter 12 - Hybrid Approach for Classification of Electroencephalographic Signals Using Time–Frequency Images With Wavelets and Texture Features, In Intelligent Data-Centric Systems, Intelligent Data Analysis for Biomedical Applications. https://doi.org/10.1016/B978-0-12-815553-0.00013-6.

[25] Rumelhart, D., Hinton, G. & Williams, R. (1986). Learning representations by back-propagating errors. Nature 323, 533–536. https://doi.org/10.1038/323533a0

[26] J. Kiefer & J. Wolfowitz. (1952). Stochastic Estimation of the Maximum of a Regression Function. The Annals of Mathematical Statistics, 23(3), 462–466. https://doi.org/10.1214/aoms/1177729392

[27] Scikit-Learn. (n.d.). 1.17. Neural network models (supervised). Retrieved 28 March 2022, from https://scikit-learn/stable/modules/neural_networks_supervised.html

[28] Heaton, J. (2008). Introduction to Neural Networks with Java. Heaton Research, Inc.

[29] Geron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly.

[30] Hu, M., Ge, D., Telford, R., Stephen, B., & Wallom, D. C. H. (2021). Classification and characterization of intra-day load curves of PV and non-PV households using interpretable feature extraction and feature-based clustering. Sustainable Cities and Society, 75, 103380. https://doi.org/10.1016/j.scs.2021.103380

[31] David, T. W., Marshall, D. P., & Zanna, L. (2017). The statistical nature of turbulent barotropic ocean jets. Ocean Modelling, 113, 34–49. https://doi.org/10.1016/j.ocemod.2017.03.008

[32] Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: A novel symbolic representation of time series. Data Mining and Knowledge Discovery, 15(2), 107–144. https://doi.org/10.1007/s10618-007-0064-z

[33] Keogh, E., Lin, J., & Fu, A. (2005). HOT SAX: Efficiently finding the most unusual time series subsequence. Fifth IEEE International Conference on Data Mining (ICDM'05), 8 pp.-. https://doi.org/10.1109/ICDM.2005.79

[34] Patel, P., Keogh, E., Lin, J., & Lonardi, S. (2002). Mining motifs in massive time series databases. 2002 IEEE International Conference on Data Mining, 2002. Proceedings., 370–377. https://doi.org/10.1109/ICDM.2002.1183925

[35] Stephen, B. (2022). Personal Communication.

[36] European Commission. (n.d.). JRC Photovoltaic Geographical Information System (PVGIS). Retrieved 09 December 2021, from https://re.jrc.ec.europa.eu/pvg_tools/en/#api_5.1

[37] European Commission. (n.d.). PVGIS user manual. Retrieved 7 April 2022, from https://joint-research-centre.ec.europa.eu/pvgis-photovoltaic-geographical-information-system/getting-started-pvgis/pvgis-user-manual_en

[38] Google. (n.d.). [Google Maps location coordinates of simulated PV generation data]. Retrieved April 1, 2022, from https://www.google.com/maps/place/55%C2%B040'37.2%22N+3%C2%B047'38.4%22W/@55.6741611,-3.8032914,4756m/data=!3m1!1e3!4m5!3m4!1s0x0:0x40746384877d9baf!8m2!3d55.677!4d-3.794

[39] Chowdhury, M., Rahman, K. S., Chowdhury, T., Nuthammachot, N., Techato, K., Akhtaruzzaman, M., Tiong, S. K., Sopian, K., & Amin, N. (2020). An overview of solar photovoltaic panels' end-of-life material recycling. Energy Strategy Reviews, 27, 100431. https://doi.org/10.1016/j.esr.2019.100431

[40] Adams, S. (2022). Personal Communication.

[41] Google Earth Pro. (2021). [38 Lockhart Drive] Retrieved December 9, 2021.

[42] Viridian Solar. (n.d.). Performance of PV Solar Panels. Retrieved 5 December 2021, from https://www.viridiansolar.co.uk/resources-4-4-performance-of-pv-solar-panels.html

[43] Lee, S., Whaley, D., & Saman, W. (2014). Electricity Demand Profile of Australian Low Energy Houses. Energy Procedia, 62. https://doi.org/10.1016/j.egypro.2014.12.370

[44] Met Office. (n.d.). When does summer start? Retrieved 19 November 2021, from https://www.metoffice.gov.uk/weather/learn-about/weather/seasons/summer/when-does-summer-start

[45] Gavin C. (2014, March). Seasonal variations in electricity demand. Retrieved 10 April 2022, from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/295225/Seasonal_variations_in_electricity_demand.pdf

[46] Neptune.Ai. (2021). Dimensionality Reduction for Machine Learning. Retrieved 18 March 2022, from https://neptune.ai/blog/dimensionality-reduction

[47] California Independent System Operator. (2016). What the duck curve tells us about managing a green grid. Retrieved 12 May 2022, from https://www.caiso.com/Documents/FlexibleResourcesHelpRenewables_FastFacts.pdf

[48] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

[49] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

[50] Szandała, T. (2021). Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks. In A. K. Bhoi, P. K. Mallick, C.-M. Liu, & V. E. Balas (Eds.), Bio-inspired Neurocomputing (Vol. 903, pp. 203–224). Springer Singapore. https://doi.org/10.1007/978-981-15-5495-7_11

[51] Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. ArXiv:1412.6980 [Cs]. http://arxiv.org/abs/1412.6980

[52] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geoscientific Model Development, 7(3), 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

[53] The Eco Experts. (2021). How Many People Have Solar Panels in the UK? https://www.theecoexperts.co.uk/solar-panels/popularity-of-solar-power

[54] Qualtrics. (n.d.). Interpreting Residual Plots to Improve Your Regression. Retrieved 28 April 2022, from https://www.qualtrics.com/support/stats-iq/analyses/regression-guides/interpreting-residual-plots-improve-regression/

[55] Dong, C., et al. (2016). Forecasting residential solar photovoltaic deployment in California. Technological Forecasting and Social Change. http://dx.doi.org/10.1016/j.techfore.2016.11.021

[56] Abdeen, A., Kharvari, F., O'Brien, W., & Gunay, B. (2021). The impact of the COVID-19 on households' hourly electricity consumption in Canada. Energy and Buildings, 250, 111280. https://doi.org/10.1016/j.enbuild.2021.111280

[57] Bagalini, V., Zhao, B. Y., Wang, R. Z., & Desideri, U. (2019). Solar PV-Battery-Electric Grid-Based Energy System for Residential Applications: System Configuration and Viability. Research, 2019. https://doi.org/10.34133/2019/3838603