

# Electrical Model-Free Voltage Calculations Using Neural Networks and Smart Meter Data

Vincenzo Bassi<sup>ID</sup>, Graduate Student Member, IEEE, Luis F. Ochoa<sup>ID</sup>, Senior Member, IEEE,  
Tansu Alpcan<sup>ID</sup>, Senior Member, IEEE, and Christopher Leckie<sup>ID</sup>

**Abstract**—The proliferation of residential technologies such as photovoltaic (PV) systems and electric vehicles can cause voltage issues in low voltage (LV) networks. During operation, voltage calculations can help determining settings, such as PV curtailment, that ensure compliance with statutory limits. In planning, voltage calculations can help assessing the effects of new connection requests. However, the main challenge for distribution companies is that voltage calculations require power flow analyses (or similar) that need accurate electrical models which, for LV networks, are not readily available for most companies. This paper proposes a scalable electrical model-free voltage calculation methodology that uses Neural Networks to capture the underlying relationships among historical smart meter data ( $P$ ,  $Q$ , and  $V$ ) and the corresponding LV network. The methodology is demonstrated using half-hourly data from one realistic LV network (146 customers) from Victoria, Australia, over three weeks and with 20% PV penetration. To account for upstream voltage fluctuations, an integrated MV-LV network with 3,400+ customers is considered. Results considering different weeks with the same and higher PV penetrations demonstrate an average error of less than 2 Volts; showing that, in the absence of LV electrical models, the methodology could be used by distribution companies for different applications.

**Index Terms**—Distribution networks, electric vehicles, low voltage networks, neural networks, photovoltaic systems, smart meters, voltage calculations.

## I. INTRODUCTION

ENVIRONMENTAL concerns along with declining technology costs have triggered an increasing adoption of residential photovoltaic (PV) systems and electric vehicles (EVs) in low voltage (LV) networks [1], [2]. However, the increasing proliferation of such technologies can cause voltage issues in existing networks. Reverse power flows from residential PV systems can lead to excessive voltage rise [3] whereas the larger demand due to EVs can exacerbate voltage drops [4].

Manuscript received 24 April 2022; revised 19 August 2022; accepted 3 December 2022. Date of publication 8 December 2022; date of current version 21 June 2023. This work was supported by The University of Melbourne. Paper no. TSG-00567-2022. (Corresponding author: Vincenzo Bassi.)

Vincenzo Bassi, Luis F. Ochoa, and Tansu Alpcan are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: vbassiz@ieee.org; luis\_ochoa@ieee.org; tansu.alpcan@unimelb.edu.au).

Christopher Leckie is with the School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: caleckie@unimelb.edu.au).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSG.2022.3227602>.

Digital Object Identifier 10.1109/TSG.2022.3227602

Therefore, to adequately assess the effects on voltages, distribution companies carry out calculations which can be based on simplified approaches (e.g.,  $\Delta V$  per kW) or more accurate ones (e.g., using power flows).

Voltage calculations are at the core of many applications in distribution networks. During operation, voltage calculations can help determining specific settings for these new technologies (e.g., PV curtailment, reactive power absorption or export limits) to ensure compliance with statutory voltage limits. In planning, voltage calculations can help assessing the effects of new connection requests and determine if further actions are needed. However, the main challenge for distribution companies is that accurate voltage calculations require power flow analyses that need detailed electrical models which, in the context of LV networks, are not readily available in most parts of the world. The associated data (e.g., topology, connectivity, phase grouping, impedances, etc.) is often erroneous or incomplete [5], [6] which makes the production of electrical models a very time-consuming and costly task.

On the other hand, data-driven and machine learning techniques have been widely used in power systems, including but not limited to security, stability, reliability, and resilience applications [7], [8], [9], [10], fault detection [7], [11], [12], state estimation [7], [13], as well as, wind, solar, and load forecasting [7], [14]. In the context of distribution networks, the increasing deployment of smart meters around the world [15] brings up interesting opportunities. Smart meters provide historical measurements at customer level, i.e., instantaneous, or average voltage magnitudes ( $V$ ) as well as current magnitude and power factor/angle measurements, from which active and reactive power ( $P$  and  $Q$ ) can be derived. These measurements can have resolutions that range from a few seconds to minutes, depending on how distribution companies collect, handle, and store the data. With all this newly available data, several data analytics applications are being explored around the world [16]. However, while some of these applications are leading to great advances in topology identification, phase grouping and impedance estimation [17], [18], [19], [20], [21], the combination of those approaches to build LV electrical models can be computationally time-consuming and might suffer from error accumulation [22] that can affect the calculation of voltages.

An alternative to calculate voltages, without building electrical models, is to exploit the historical data (i.e.,  $P$ ,  $Q$  and  $V$ ) through regression methods so as to capture the physics of the corresponding LV network. Once these relationships are

captured, electrical model-free (i.e., without electrical models) voltage calculations can be carried out for any what-if scenario by simply specifying customers' active and reactive power (i.e.,  $P$  and  $Q$ ) at a given instant. Such electrical model-free voltage calculations can make possible a wide range of operation and planning studies.

The electrical model-free calculation of voltages is a topic that has only started to be explored in the last few years [22], [23], [24], [25], [26], [27], [28], [29], different regression methods have been tested showing promising results, including linear [22], [23], [28], Gaussian process [24] and neural networks [25], [26], [27], [28]. However, most of these approaches consider only medium voltage (MV) networks [22], [23], [24], [25], [26], [27] and, crucially, use measurements that are not common (such as at the distribution transformer [22], [23], [24], [25], [26], [27] or do not exist (such as voltage angles [22], [27]) in most distribution companies. Even though those measurements are not typically available, they do not capture the unbalanced nature of three-phase LV networks and the resulting diversity in single, two and/or three-phase customer voltages. Thus, those approaches might not work with residential smart meter data. Full three-phase LV networks were investigated in [28] with small PV penetrations. It concluded that linear regressions and neural networks achieved the most accurate voltage calculations. Nonetheless, the interactions with the upstream MV network were not considered. This, in practice, can significantly affect the voltages seen by residential customers and, hence, the performance of the methods investigated. Additionally, in [29], smart meter data that captured the interactions with the MV network was used to calculate voltages to assess the effects of high PV penetrations. Nonetheless, the adopted regression model was not disclosed, and the effects of reactive power were not considered. Furthermore, due to the absence of the electrical model, the accuracy of the approach was not validated for higher PV penetration levels than those observed in the historical smart meter data. Finally, the electrical model-free calculation of voltage sensitivities has been explored in [30], [31], [32]. However, results are presented in terms of sensitivity coefficients and, consequently, the overall calculation of voltages is not assessed.

This paper proposes a scalable methodology based on non-linear regressions, specifically using Neural Networks (NNs), to calculate voltages without electrical models. A tailored NN is trained to capture the underlying relationships among the historical per-phase smart meter data of customers (i.e.,  $P$ ,  $Q$ , and  $V$ ) and the studied LV network. The main contributions of this paper are as follows:

- 1) It proposes a scalable methodology that defines an architecture and hyperparameters for the NN, making possible its implementation by distribution companies without the time-consuming process of tailoring a NN per each LV network;
- 2) Expanding and improving the authors' prior work [33] and [34], the effectiveness of the proposed methodology is demonstrated using multiple LV feeders simultaneously (all connected to the same MV/LV transformer)

and, crucially, accounts for the effects of the upstream MV network on LV network voltages; and,

- 3) It proposes two architectures of NNs that could be used for operation (short-term) and planning (mid-, or long-term) applications: a conventional NN and a novel tailored multi-input multi-output NN (to enhance the extrapolation capabilities).

The methodology is demonstrated using realistic half-hourly data from a LV network (146 customers) from Victoria, Australia, over three weeks and with 20% PV penetration. To account for upstream voltage fluctuations, an integrated MV-LV network with 3,400+ customers is considered. The accuracy of the model-free voltage calculations is assessed considering different what-if scenarios, i.e., the same and larger PV penetration levels than those in the historical data.

## II. MODEL-FREE VOLTAGE CALCULATIONS

This section presents the formulation of the proposed model-free voltage calculations methodology. The historical per-phase smart meter data that will be used to determine the most suitable Neural Network (NN) corresponds to the data sets in (1)–(3).  $P_t$ ,  $Q_t$  and  $V_t$  represent instances containing samples of active power, reactive power, and voltage magnitudes, respectively, for all customers in  $C$  (set of customers) at time  $t \in T$  (set of instances). Therefore, the data sets  $\mathbf{P}$ ,  $\mathbf{Q}$ , and  $\mathbf{V}$  are built with instances  $P_t$ ,  $Q_t$  and  $V_t$ , respectively, collected at regular time intervals (e.g., 5 minutes, 30 minutes) throughout the considered period (e.g., a week, a month). Note that  $|T|$  and  $|C|$  corresponds to the total number of instances in set  $T$  and customers in set  $C$ , respectively.

$$\mathbf{P} = \begin{bmatrix} P_1 \\ \vdots \\ P_{|T|} \end{bmatrix} = [\mathbf{P}_t] \in \mathbb{R}^{|T| \times |C|} \quad (1)$$

$$\mathbf{Q} = \begin{bmatrix} Q_1 \\ \vdots \\ Q_{|T|} \end{bmatrix} = [\mathbf{Q}_t] \in \mathbb{R}^{|T| \times |C|} \quad (2)$$

$$\mathbf{V} = \begin{bmatrix} V_1 \\ \vdots \\ V_{|T|} \end{bmatrix} = [\mathbf{V}_t] \in \mathbb{R}^{|T| \times |C|} \quad (3)$$

Using (1)–(3), multiple NNs are trained to fit the underlying relationships among its inputs (i.e., customers' active and reactive power,  $\mathbf{P}$  and  $\mathbf{Q}$ ) and its outputs (i.e., customers' voltages,  $\mathbf{V}$ ), as presented in (4).  $\mathbf{W}_{NN_i}$  represents all parameters of the NN  $i$ , whose dimensions will vary depending on the structure of each NN (number of hidden layers and number of neurons; further details in Section II-A).

$$\mathbf{V} = f_{NN_i}(\mathbf{P}, \mathbf{Q}, \mathbf{W}_{NN_i}) + E_{NN_i} \quad (4)$$

The relationship  $f_{NN_i}$  presented in (4) is captured by defining suitable parameters  $\mathbf{W}_{NN_i}$  that attempt to reduce the resulting approximation error for each instance ( $E_{NN_i,t}$ , where  $E_{NN_i} = [E_{NN_i,t}] \in \mathbb{R}^{|T|}$ ). With the most suitable NN ( $NN_*$ ), voltage calculations for any kind of what-if scenario (customers' demand/generation) can be carried out. A what-if scenario is

defined by  $\bar{P}$  in (5) and  $\bar{Q}$  in (6), which contain one instance of active and reactive power values for all customers, respectively. All customers voltages,  $\bar{V}^{calc}$ , can be calculated using the relationship  $f_{NN_*}$  as in (7).

$$\bar{P} = [\bar{p}_1 \ \cdots \ \bar{p}_{|C|}] = [\bar{p}_c] \in \mathbb{R}^{|C|} \quad (5)$$

$$\bar{Q} = [\bar{q}_1 \ \cdots \ \bar{q}_{|C|}] = [\bar{q}_c] \in \mathbb{R}^{|C|} \quad (6)$$

$$\bar{V}^{calc} = f_{NN_*}(\bar{P}, \bar{Q}, W_{NN_*}) \quad (7)$$

### A. Neural Networks

This paper uses Neural Networks (NNs) as regression method due to its ability to learn and model complex multi-dimensional nonlinear relationships and its successful application in previous works in the area [25], [26], [27], [28]. NNs correspond to mathematical models composed of interconnected fundamental units (or neurons). A NN is comprised of several neurons stacked in three different types of layers: input, output, and hidden layers. The output of all neurons stacked in each layer corresponds to the inputs of the subsequent layer. The structure and training process of a NN is defined by external configuration variables called hyperparameters (e.g., number of neurons, learning rate, epochs, etc.). Once a given NN is trained,  $NN_i$ , specific weights and biases ( $W_{NN_i}$ ), called parameters, are produced [35].

The output of a single neuron corresponds to an activation function applied over the weighted sum of its inputs shifted by a bias, as shown in (8). For  $NN_i$  with  $|H|$  hidden layers (set of hidden layers defined as  $H$  and indexed by  $h$ ), the output of the  $j - th$  neuron in the  $h - th$  hidden layer is denoted by  $Z_j^h$ .  $\varphi^h$  corresponds to the activation function whereas  $w_{ju}^h$  and  $b_{j0}^h$ , with  $j \in N_h$  (set of neurons in the  $h - th$  hidden layer) and  $u \in N_{h-1}$  (set of neurons in the  $(h-1) - th$  hidden layer), represent the weights and biases, respectively.  $Z_u^{h-1}$  with  $u \in N_{h-1}$  corresponds to the output of the  $u - th$  neuron in the  $(h-1) - th$  hidden layer [35]. Note that for the first hidden layer, i.e.,  $h = 1$ ,  $Z_u^0$  with  $u \in N_0$  (set of inputs) accounts for the  $u - th$  input of  $NN_i$ .

$$Z_j^h = \varphi^h \left( \sum_{u \in N_{h-1}} w_{ju}^h Z_u^{h-1} + b_{j0}^h \right) \quad (8)$$

For the proposed model-free voltage calculations, the inputs for  $NN_i$  are both the active and reactive power of all the customers, thus, the input layer dimension of  $NN_i$  is twice the number of customers, i.e.,  $2|C|$ . The corresponding input matrix is  $P|Q$ , shown in (9), which is the augmented matrix of data sets  $P$  and  $Q$  in (1) and (2). Thus, a column vector of this new matrix will contain all instances of active power (first half) or reactive power (second half) for a specific customer, as shown in (10). Similarly, the outputs for  $NN_i$  are the voltages of the customers, thus, the output layer dimension is  $|C|$ . The corresponding output matrix is  $V$ , shown in (3). Each of its column vectors is comprised of all instances of voltage for a specific customer as shown in (11). Consequently,  $NN_i$  inputs (i.e.,  $Z_a^0$  with  $a \in N_0$ ) are given by (10), whereas  $NN_i$  outputs (i.e.,  $Z_l^{|H|+1}$  with  $l \in N_{|H|+1}$ , set of outputs) are given

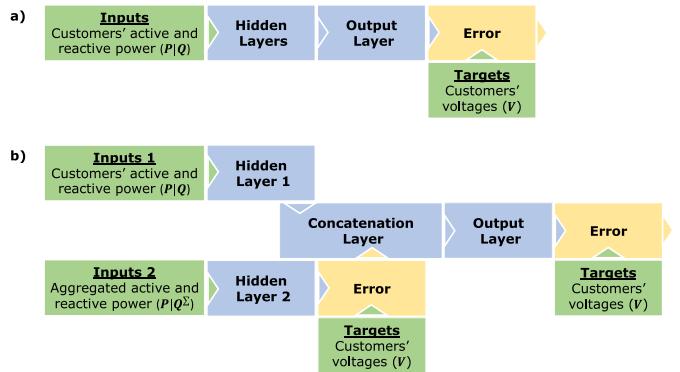


Fig. 1. Schematic of the building blocks in the training process of each NN architecture: a)  $NN_i$ , b)  $MNN_i$ .

by (11). Note that the length of all  $NN_i$  inputs and outputs correspond to the number of instances  $|T|$ .

$$P|Q = \begin{bmatrix} P_1 & Q_1 \\ \vdots & \vdots \\ P_{|T|} & Q_{|T|} \end{bmatrix} = [P_t|Q_t] \in \mathbb{R}^{|T| \times (2|C|)} \quad (9)$$

$$\begin{bmatrix} Z_1^0, \dots, Z_{2|C|}^0 \end{bmatrix} = [P|Q_{[:,1]}, \dots, P|Q_{[:,2|C|}]] = P|Q \in \mathbb{R}^{|T| \times (2|C|)} \quad (10)$$

$$\begin{bmatrix} Z_1^{|H|+1}, \dots, Z_{|C|}^{|H|+1} \end{bmatrix} = [V_{[:,1]}, \dots, V_{[:,|C|}]] = V \in \mathbb{R}^{|T| \times |C|} \quad (11)$$

Thus, the  $l - th$  column vector of the output matrix  $V$  for a  $NN_i$  with  $|H|$  hidden layers is given by (12). The superscript  $|H|$  accounts for the last hidden layer and  $N_{|H|}$  represents its corresponding set of neurons. Additionally,  $W_{NN_i}$  aggregates all weights and biases of  $NN_i$ .

$$V_{[:,l]}(P|Q, W_{NN_i}) = \varphi^{|H|+1} \left( \sum_{u \in N_{|H|}} w_{lu}^{|H|+1} Z_u^{|H|} + b_{l0}^{|H|+1} \right) \quad (12)$$

For regression problems, the activation function for the output layer,  $\varphi^{|H|+1}$ , is typically a linear function [35]. Nonlinearities, however, will be introduced by the activation functions in the hidden layers,  $\varphi^h$ . Therefore, it is possible to ensure that  $NN_i$  corresponds to a nonlinear regression function from the set of input variables  $P|Q$  to the set of output variables  $V$  as in (13). A schematic of  $NN_i$  is shown in Fig. 1 (a).

$$V = f_{NN_i}(P|Q, W_{NN_i}) + E_{NN_i} \quad (13)$$

To find the weights and biases in  $W_{NN_i}$ ,  $NN_i$  is trained with the historical data sets  $P|Q$  and  $V$ . The first step is to define an error function such as the Mean Squared Error (MSE); commonly used in regression problems [35]. Then, the weight and bias parameters in  $W_{NN_i}$  are obtained by minimizing the MSE among actual voltage values,  $V_t$ , and calculated voltages,  $\hat{V}_{NN_i,t}(P|Q_t, W_{NN_i})$ , as in (14), where  $P|Q_t = P_t|Q_t$  and  $V_t - \hat{V}_{NN_i,t}(P|Q_t, W_{NN_i}) = E_{NN_i,t}$ .

$$\min_{W_{NN_i}} \frac{1}{|T|} \sum_{t \in T} (V_t - \hat{V}_{NN_i,t}(P|Q_t, W_{NN_i}))^2 \quad (14)$$

Due to the nonlinearities introduced by the activation functions, (14) corresponds to a nonconvex optimization problem. To solve this, gradient based algorithms are often used. For the present study, mini batch gradient descent is considered. Thus, the training data set is shuffled and split in different disjoint slices of equal length (mini batches) and, for each complete pass through the entire data set (known as epoch) interim values calculations and parameters updates are carried out using each mini batch separately [36].

Once the most suitable NN for the studied LV network is found (denoted as  $NN_*$ ), voltage calculations for any kind of what-if scenario can be carried out as in (7).

A conventional NN will be one that has one set of inputs ( $\mathbf{P}|\mathbf{Q}$ ) and one set of outputs ( $\mathbf{V}$ ) as shown in Fig. 1 (a). However, there can be opportunities to improve the NN if additional information is considered.

### B. Multi-Input Multi-Output Neural Network

Given that the voltages of all the customers in a given LV network are electrically coupled to the voltages at the secondary side of the corresponding MV/LV transformer, such information could be used to better capture the physics and, hence, enhance the extrapolation capabilities of the proposed model-free voltage calculations. Since such measurements are not common for most distribution companies, an alternative is to use the aggregated active and reactive power of the customers as such values are related, to some extent, to the voltage fluctuations seen at the secondary of the transformer.

To handle the use of additional information, a multi-input multi-output NN, hereafter referred to as MNN, is proposed. The MNN incorporates the aggregated active and reactive power in the form of  $P^\Sigma$  in (15) and  $Q^\Sigma$  in (16).  $P_t^\Sigma$  and  $Q_t^\Sigma$  with  $t \in T$  contains the sum of all samples of active power in  $P_t$  and reactive power in  $Q_t$ , respectively. Thus,  $P^\Sigma$  and  $Q^\Sigma$  contain all the corresponding instances  $P_t^\Sigma$  and  $Q_t^\Sigma$ . Note that  $P^\Sigma$  in (15) and  $Q^\Sigma$  in (16) are directly constructed based on the historical data sets  $\mathbf{P}$  in (1) and  $\mathbf{Q}$  in (2) and, thus, no additional measurements are required. Like  $\mathbf{P}|\mathbf{Q}$  in (9),  $\mathbf{P}|\mathbf{Q}^\Sigma$  in (17) denotes the augmented matrix of data sets  $P^\Sigma$  and  $Q^\Sigma$  in (15) and (16), respectively.

$$\mathbf{P}^\Sigma = \begin{bmatrix} P_1^\Sigma \\ \vdots \\ P_{|T|}^\Sigma \end{bmatrix} = \begin{bmatrix} P_t^\Sigma \end{bmatrix} \in \mathbb{R}^{|T|} \quad (15)$$

$$\mathbf{Q}^\Sigma = \begin{bmatrix} Q_1^\Sigma \\ \vdots \\ Q_{|T|}^\Sigma \end{bmatrix} = \begin{bmatrix} Q_t^\Sigma \end{bmatrix} \in \mathbb{R}^{|T|} \quad (16)$$

$$\mathbf{P}|\mathbf{Q}^\Sigma = \begin{bmatrix} P_1^\Sigma & Q_1^\Sigma \\ \vdots & \vdots \\ P_{|T|}^\Sigma & Q_{|T|}^\Sigma \end{bmatrix} = \begin{bmatrix} P_t^\Sigma & Q_t^\Sigma \end{bmatrix} \in \mathbb{R}^{|T| \times 2} \quad (17)$$

The process to find the most suitable MNN for the studied LV network, i.e.,  $MNN_*$ , involves the training of multiple MNNS. Customers' active and reactive power,  $\mathbf{P}|\mathbf{Q}$ , are used along with its aggregated values,  $\mathbf{P}|\mathbf{Q}^\Sigma$ , as inputs of the  $i$ -th MNN,  $MNN_i$ , as shown in Fig. 1 (b). For one side, inputs

1 correspond to customers' active and reactive power ( $\mathbf{P}|\mathbf{Q}$ ), same as before. These inputs are processed through a single hidden layer (hidden layer 1), whose number of neurons is to be determined. On the other hand, inputs 2 correspond to aggregated active and reactive power of customers, which are processed through a hidden layer (hidden layer 2) with fixed dimensions, i.e., two inputs ( $P^\Sigma$  and  $Q^\Sigma$  in  $\mathbf{P}|\mathbf{Q}^\Sigma$ ) and  $|C|$  neurons. Hence, hidden layer 2 aims to capture the underlying relationships among  $\mathbf{P}|\mathbf{Q}^\Sigma$  (related, to some extent, to voltages at the secondary of the transformer) and  $\mathbf{V}$ . This is done by incorporating an additional error function at the end of hidden layer 2 which minimize the error between  $\hat{V}_{MNN_i,t}^{HL2}$  (voltages calculated at the output of hidden layer 2 at instance  $t$ ) and customers' actual voltage values  $V_t$ . The outputs of hidden layer 1 and hidden layer 2 are then concatenated and an activation function is applied. The outputs of the concatenation layer correspond to the inputs of the output layer, where an error function is calculated among calculated voltage values,  $\hat{V}_{MNN_i,t}$ , and actual values  $V_t$ . Finally, the weight and bias parameters of  $MNN_i$  are calculated as in (18), where  $E_{MNN_i,t}^{HL2}$  and  $E_{MNN_i,t}$  are shown in (19) and (20), respectively.  $\mathbf{W}_{MNN_i}^{HL2}$  contains the weight and biases of hidden layer 2 only, whereas  $\mathbf{W}_{MNN_i}$  is comprised of all the weight and biases, i.e.,  $\mathbf{W}_{MNN_i}^{HL2}$  is a subset of  $\mathbf{W}_{MNN_i}$ .

$$\min_{\mathbf{W}_{MNN_i}^{HL2}, \mathbf{W}_{MNN_i}} \frac{1}{|T|} \sum_{t \in T} \left( E_{MNN_i,t}^{HL2} \right)^2 + \left( E_{MNN_i,t} \right)^2 \quad (18)$$

$$E_{MNN_i,t}^{HL2} = V_t - \hat{V}_{MNN_i,t}^{HL2} \left( \mathbf{P}|\mathbf{Q}_t^\Sigma, \mathbf{W}_{MNN_i}^{HL2} \right) \quad (19)$$

$$E_{MNN_i,t} = V_t - \hat{V}_{MNN_i,t} \left( \mathbf{P}|\mathbf{Q}_t^\Sigma, \mathbf{P}|\mathbf{Q}_t^\Sigma, \mathbf{W}_{MNN_i} \right) \quad (20)$$

Mini batch gradient descent is also used to solve the non-convex optimization problem formulated in (18). Note that two error functions are minimized simultaneously in (18), i.e.,  $E_{MNN_i,t}^{HL2}$  in (19) and  $E_{MNN_i,t}$  in (20). Consequently,  $MNN_i$  delivers two outputs,  $\hat{V}_{MNN_i,t}^{HL2}$  and  $\hat{V}_{MNN_i,t}$ . The voltages considered for model-free voltage calculations correspond to those voltages that are produced at the end of the output layer, i.e.,  $\hat{V}_{MNN_i,t}$ . Voltages produced at the end of hidden layer 2, i.e.,  $\hat{V}_{MNN_i,t}^{HL2}$ , are less accurate as its calculation is based on aggregated active and reactive power only and are used to include additional information into the formulation, i.e., relationships among  $\mathbf{P}|\mathbf{Q}^\Sigma$  and  $\mathbf{V}$ . In this case,  $\hat{V}_{MNN_i,t}^{HL2}$  can be considered as an additional or auxiliary task which has been demonstrated to improve the generalization capabilities of NNs [37]. The equation for  $MNN_i$  can be formulated as in (21), where  $E_{MNN_i}^{HL2} = [E_{MNN_i,t}^{HL2}] \in \mathbb{R}^{|T|}$  and  $E_{MNN_i} = [E_{MNN_i,t}] \in \mathbb{R}^{|T|}$ .

$$\mathbf{V} = f_{MNN_i} \left( \mathbf{P}|\mathbf{Q}, \mathbf{P}|\mathbf{Q}^\Sigma, \mathbf{W}_{MNN_i} \right) + E_{MNN_i}^{HL2} + E_{MNN_i} \quad (21)$$

Once the most suitable MNN for the studied LV network is found (noted as  $MNN_*$ ), voltage calculations for any kind of what-if scenarios can be carried out as presented in (22), where  $\overline{\mathbf{P}|\mathbf{Q}}$  and  $\overline{\mathbf{P}|\mathbf{Q}^\Sigma}$  are calculated as in (9) and (17), respectively, using  $\overline{\mathbf{P}}$  and  $\overline{\mathbf{Q}}$  values of the corresponding what-if scenario.

$$\overline{\mathbf{V}}^{calc} = f_{MNN_*} \left( \overline{\mathbf{P}|\mathbf{Q}}, \overline{\mathbf{P}|\mathbf{Q}^\Sigma}, \mathbf{W}_{MNN_*} \right) \quad (22)$$

### C. Neural Network Selection

The accuracy of a NN is sensitive to its hyperparameters which defines its structure and learning process. Furthermore, the physical relationships of each LV network will depend on its specific characteristics (number of customers, topology, impedances, etc.). Therefore, the hyperparameters that are found to be the most suitable for a particular LV network will not work when using another LV network with different characteristics. Consequently, a scalable methodology to find the most suitable NNs (i.e.,  $NN_*$  and  $MNN_*$ ) is crucial so the same *recipe* can be applied by distribution companies on any type or size of LV network.

## III. METHODOLOGY

The proposed methodology is comprised of two main stages: training data assembly and NN selection. In the first stage, data is collected and organized using the historical per-phase smart meter data of customers. In the second stage, thousands of NNs are trained and compared to find the most suitable hyperparameters (through a k-fold cross validation process). Then, 10 NNs are trained from scratch using those hyperparameters to define the best one ( $NN_*$  or  $MNN_*$ ) ready to carry out model-free voltage calculations.

### A. Training Data Assembly

Using the historical data sets  $\mathbf{P}$ ,  $\mathbf{Q}$ , and  $\mathbf{V}$  presented in (1)–(3), the training data set  $\{(P_t, Q_t, V_t), t \in T^t \subseteq T\}$  is assembled considering  $T^t$  as the set of training instances. Note that  $\{(P_t, Q_t)\}$  are used as  $NN_i$  (or  $MNN_i$ ) inputs in the form  $\{P|Q_t = (P_t|Q_t), t \in T^t \subseteq T\}$  as in (10). From this data set, training values of aggregated active and reactive power of all customers are obtained  $\{(P_t^\Sigma, Q_t^\Sigma), t \in T^t \subseteq T\}$ . This data is used as  $MNN_i$  inputs in the form  $\{P|Q_t^\Sigma = (P_t^\Sigma|Q_t^\Sigma), t \in T^t \subseteq T\}$ .

### B. Neural Network Selection

The NN selection process is comprised of three steps. First, based on the characteristics of the problem, the number of inputs and outputs can be defined. Thus,  $NN_i$  inputs correspond to the per-phase active and reactive power of customers (i.e., number of  $NN_i$  inputs is  $2|C|$ ), whereas  $NN_i$  outputs correspond to the per-phase voltages, (i.e., number of  $NN_i$  outputs is  $|C|$ ). Similarly,  $MNN_i$  inputs correspond to the per-phase active and reactive power of customers as well as its aggregated values (i.e., number of  $MNN_i$  inputs is  $2|C| + 2$ ), while  $MNN_i$  outputs are also given by per-phase voltages, in both main and auxiliary tasks (i.e., number of  $MNN_i$  outputs is  $2|C|$ ). Besides, a linear activation function is considered for the output layer and all NNs are trained considering Mean Squared Error (MSE) as error function, commonly used in regression problems. Additionally, inputs and outputs are scaled to values within the range [0, 1] to speed up learning and convergence processes, and the ADAM optimizer [38] is used throughout all training processes due to its suitability to a wide spectrum of machine learning problems. Finally, all NNs consider L2 regularization to prevent overfitting on the historical

TABLE I  
RESIDENTIAL PV SYSTEMS CAPACITY DISTRIBUTION

PV Size kW	2.5	3.5	5.5	8.0
Distribution %	8	24	52	16

data and to enhance its extrapolation capabilities. This adds a penalty factor over the squared norm of the weight parameters of  $NN_i$  in (14) and of  $MNN_i$  in (18). Such penalty factor is known as regularization factor and is to be determined.

Then, to define the remaining hyperparameters (i.e., number of hidden layers, number of neurons and activation function in each hidden layer, learning rate, batch size, regularization factor, and epochs) a k-fold cross validation process is implemented. To this end, the search spaces for each of the remaining hyperparameters need to be defined. Then, the training data is split into  $k$  continuous disjoint folds of the same length and, for each plausible hyperparameter combination in the defined search spaces,  $k$  independent  $NN_i$  (or  $MNN_i$ ) are trained. Each of the  $k$  runs use  $k - 1$  folds for training and the remaining fold as validation fold so that each fold is used as validation fold once. Each hyperparameter combination is assessed through the average of the validation RMSE (root mean squared error) obtained by its  $k$  runs, denoted as  $RMSE_{kfold}$ . The hyperparameter combination that achieves the lowest  $RMSE_{kfold}$  define all the remaining hyperparameters.

Once all hyperparameters are defined, 10 NNs are trained independently from scratch using the entire training data set. The NNs that achieve the lowest RMSE within the training data corresponds to  $NN_*$  (or  $MNN_*$ ).

A key feature of the proposed methodology is that the search spaces defined for the number of neurons in each hidden layer contains numbers that are proportional to the number of customers. Hence, as the number of NN inputs, NN outputs, and number of neurons within the corresponding search spaces are always proportional to the number of customers, the proposed methodology can be scaled to any type or size of LV network, depending only on the number of customers and their historical smart meter data.

## IV. CASE STUDY

This section presents the case study used to demonstrate the effectiveness of the proposed methodology and assess its accuracy considering the same and larger PV penetration levels than those observed in the historical data.

### A. Realistic, Integrated MV-LV Network

The studied LV network (shown in Fig. 2) is comprised of 4 LV feeders (400 V line-to-line) with a total of 146 (31, 40, 36, and 39) single-phase customers of which, initially, 20% have PV systems installed. The distribution of PV system sizes is based on Australian statistics from 2016 – 2018 [39] and is presented in Table I. Additionally, a fixed maximum export limit of 5 kVA is considered as in Australian LV networks [40]. The transformation ratio of the MV/LV transformer is 22 kV/ 433 V, i.e., providing a natural boost of 8.25% which is commonly used in Australia.

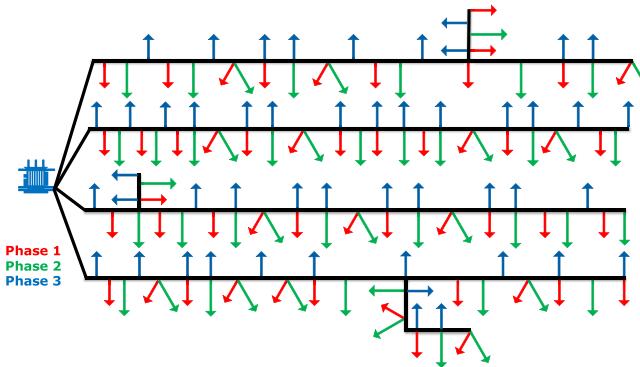


Fig. 2. Realistic LV Network.

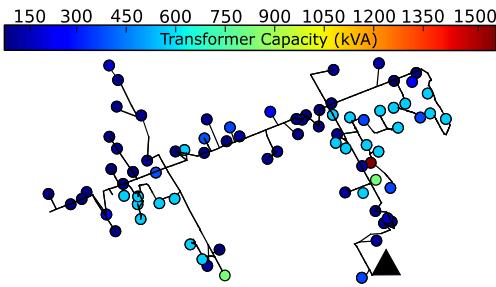


Fig. 3. Real MV Feeder.

To cater for the voltage fluctuations due to the upstream MV network, the LV network is located at a remote end of a realistic, integrated MV-LV network. The latter is comprised of a real MV feeder (22 kV), from Victoria, Australia, owned and operated by AusNet Services (Victorian distribution company), and realistically modeled LV networks. The MV feeder is supplied by a transformer of 66 kV/22 kV and is comprised of 79 MV/LV transformers as shown in Fig. 3. This MV feeder supplies a total of 142 realistically modeled LV feeders (i.e., designed based on LV networks design principles [41]). In total, the realistic MV-LV distribution network supplies more than 3,400 single-phase customers and 9 three-phase industrial customers. A fixed 20% of PV penetration is also considered for the customers outside the studied LV network.

### B. Data and Considerations

To produce  $\mathbf{P}$ ,  $\mathbf{Q}$  and  $\mathbf{V}$  data sets that correspond to the studied LV network (connected to the MV-LV integrated network) and a given PV penetration while capturing consumption patterns and physical constraints of residential customers, real smart meter data from Victoria, Australia (anonymized half-hourly active power provided by AusNet Services) is used to run power flow simulations in OpenDSS [42]. The residential demand of each customer at each day is randomly chosen from a data set of real active power demand for 342 residential customers throughout 2016. At each half-hourly interval, a random inductive power factor in the range [0.90, 0.99] is considered for each customer. Finally, to account for natural resource fluctuations, PV generation uses normalized profiles from Victoria for 2014. Using this approach, a training set with 20% PV penetration within the studied LV network is

produced. For testing purposes, three test sets with 20, 40 and 100% PV penetrations within the studied LV network are also produced.

1) *Training (Historical) Data:* The training data set is comprised of 3 consecutive summer weeks with half-hourly resolution. The training data considers 29 of the 146 single-phase customers with PV systems. This data set is used to find the most suitable hyperparameters for the studied LV network and to train the final NNs ready to carry out voltage calculations.

2) *Test Sets:* To assess the accuracy of the proposed methodology with the same and different PV penetrations than those observed during training, 3 test sets are considered. Test 1, Test 2, and Test 3 are comprised of three consecutive summer weeks (immediately after those used for training which capture different demand and PV irradiance) and consider, 29 (same as during training), 59, and all 146 customers with PV systems, respectively.

The test data sets are carefully chosen to assess the performance of the proposed methodology for operation and planning applications. For operation, voltage calculations are required in near real-time or in advance (e.g., hours, days ahead). In this context, no changes or small changes in PV penetration levels are expected with respect to the historical data. Hence, this is assessed using Test 1 (20%) and Test 2 (40%). Conversely, for planning, voltage calculations are carried out to represent what-if situations (e.g., months or years ahead). In this case, voltages need to be calculated for significantly higher PV penetration levels than those observed in the training. This is assessed using Test 3 (100%).

### C. Neural Network Selection

All NN algorithms are implemented in the Keras deep learning API [43], which works in Python [44], on top of the open-source machine learning platform Tensorflow [45].

The first step is to set those hyperparameters that can be directly obtained from the characteristics of the problem. Table II presents all the corresponding hyperparameters and settings for both the  $NN_i$  and the  $MNN_i$  (which considers two additional inputs given by customers aggregated active and reactive power). The next step is to carry out the k-fold cross validation to find the remaining hyperparameters for each NN architecture. This process considers  $k = 3$ , i.e., each fold of 1-week length, and the search space characteristics presented in Table II. Crucially, to maintain a manageable search space, the number of hidden layers for  $NN_i$  is set to 1. For completeness, deeper NNs are investigated in the Appendix.

The hyperparameter combination that achieves the lowest  $RMSE_{fold}$  for  $NN_i$  and  $MNN_i$  (0.2011 V and 0.1978 V, respectively) appear in bold in Table II.

Finally, with all the hyperparameters being defined, 10 NNs are trained from scratch using the entire training data (i.e., all  $k = 3$  folds). The NNs with the lowest training RMSE, 0.0793 V and 0.0789 V, respectively, correspond to  $NN_*$  and  $MNN_*$  which are ready to carry out model-free voltage calculations.

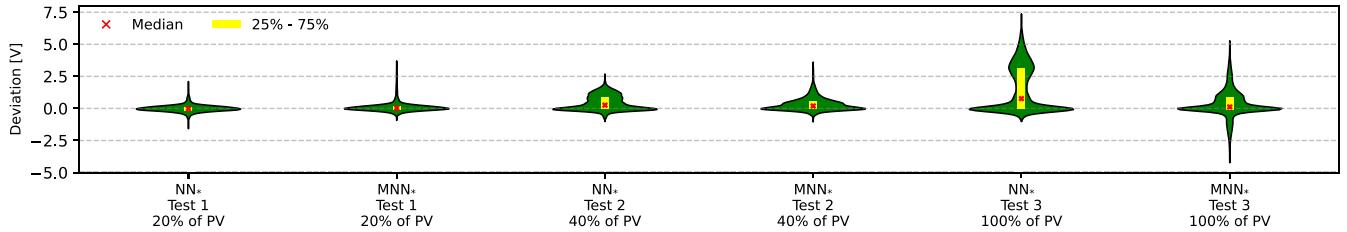


Fig. 4. Model-free voltage calculations deviation (global).

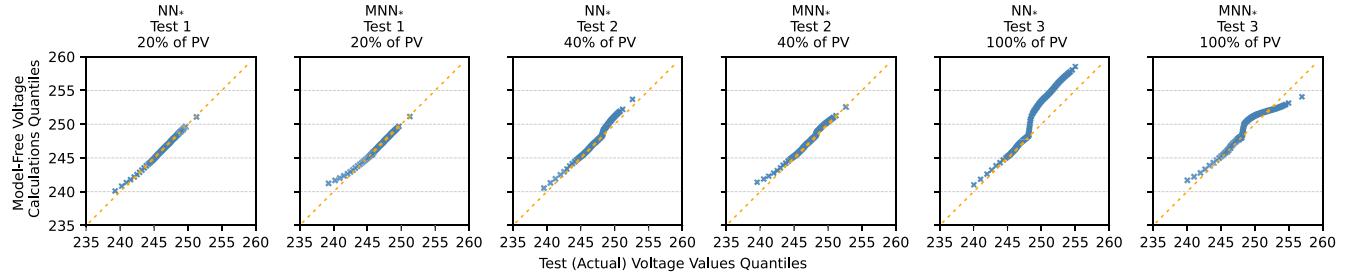


Fig. 5. Quantile-Quantile plots (global).

TABLE II  
HYPERPARAMETERS, SETTINGS AND SEARCH SPACES  
FOR  $NN_i$  AND  $MNN_i$

Hyperparameters and settings	$NN_i$	$MNN_i$
Inputs	$2 \times  C  = 292$	$2 \times  C  + 2 = 294$
Outputs	$ C  = 146$	$2 \times  C  = 292$
Output layer activation function	Linear	
Error function	MSE	
Scaler	[0,1]	
Optimizer	ADAM	
Regularization	L2	
Number of neurons	Hidden layer 1: [0.5, 1, 2, 3, 4, <b>5</b> , 6, 7, 8, 9, 10] $\times  C $	Hidden layer 1: [0.5, 1, 2, <b>3</b> , 4, 5, 6, 7, 8, 9, 10] $\times  C $
Activation function	Hidden layer 1: [ <b>Tanh</b> , ReLu, Swish]	Hidden layer 1: [ <b>Tanh</b> , Swish] Hidden Layer 2: [Tanh, Swish] Concatenation layer: [ <b>Tanh</b> , Swish]
Learning rate	$[1 \times 10^{-3}, 1 \times 10^{-4},$ $1 \times 10^{-5}]$	$[1 \times 10^{-3}, 1 \times 10^{-4},$ $1 \times 10^{-5}]$
Regularization factor	$[1 \times 10^{-3}, 1 \times 10^{-4},$ <b><math>1 \times 10^{-5}</math></b>	$[1 \times 10^{-3}, 1 \times 10^{-4},$ <b><math>1 \times 10^{-5}</math></b>
Batch size	<b>[12 (eq. 6 hrs), 24 (eq. 12 hrs), 48 (eq. 24 hrs)]</b>	[12, 24, <b>48</b> ]
Epochs	[1,000, <b>2,000</b> , 5,000]	[1,000, 2,000, <b>5,000</b> ]
Total combinations	2,673	7,128

#### D. Results: Model-Free Voltage Calculations

The performance of the model-free voltage calculations is assessed by comparing the results with *actual* voltage values which are obtained running power flow simulations as in Section IV-B. The obtained results are presented for all test sets in Table III in terms of RMSE, absolute average deviation (Av. Dev.), and absolute maximum deviation (Max. Dev.). Results are presented separately for

solar hours (06:00 to 20:00 hrs) and nonsolar hours (remaining hours).

The deviations of model-free voltage calculations (using  $NN_*$  and  $MNN_*$ ) with respect to actual voltages are presented as violin plots in Fig. 4 for all customers and for all instances, for each test set. The corresponding model-free voltage calculations and actual voltage quantiles are presented in the quantile-quantile plots in Fig. 5. The same plots considering solar hours only are shown Fig. 6 and Fig. 7, respectively. Time-series plots for model-free voltage calculations along with actual voltages are presented in Fig. 8. Finally, to illustrate the extent of deviations, the time-series model-free voltage calculations and actual voltages for the customer with the largest deviation are presented in Fig. 9. Overall, both,  $NN_*$  and  $MNN_*$ , can capture the physics of the studied LV network, obtaining accurate voltage calculations in all test sets. The worst performance is given by  $NN_*$  in Test 3 (planning) achieving an average deviation of just 1.6149 V (noting that the nominal voltage is 230 V).

In Test 1 (operation),  $NN_*$  outperforms  $MNN_*$  in all metrics achieving a maximum deviation of 2.0882 V, an average deviation of 0.1892 V, and a RMSE of 0.2681 V. In Test 2 (operation),  $NN_*$  continues to outperform  $MNN_*$  in almost 1 V in terms of maximum deviation, achieving a value of 2.6815 V. However, in terms of average deviation and RMSE,  $NN_*$  is slightly worse. In Test 3 (planning),  $MNN_*$  outperforms  $NN_*$  in all metrics, obtaining a maximum deviation of 5.2559 V, an average deviation of 0.7220 V, and a RMSE of 1.0883 V.

From Fig. 4 to Fig. 9 it can be observed that for Test 1 both,  $NN_*$  and  $MNN_*$ , have model-free voltage calculations close to the actual values. Nonetheless, as more PV installations with respect to the historical data are investigated (Test 2 and Test 3), the  $NN_*$  tends to deviate during solar hours more than  $MNN_*$ .

From Table III, as expected, it can be noted that accuracy decreases during solar hours (compared to non-solar hours) in

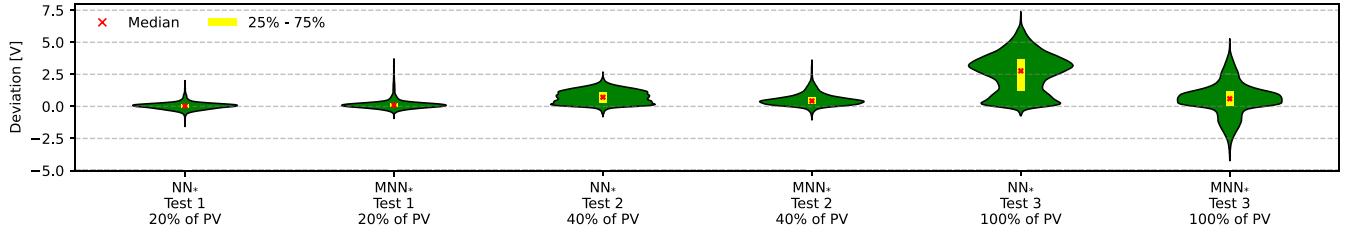


Fig. 6. Model-free voltage calculations deviation (solar hours only).

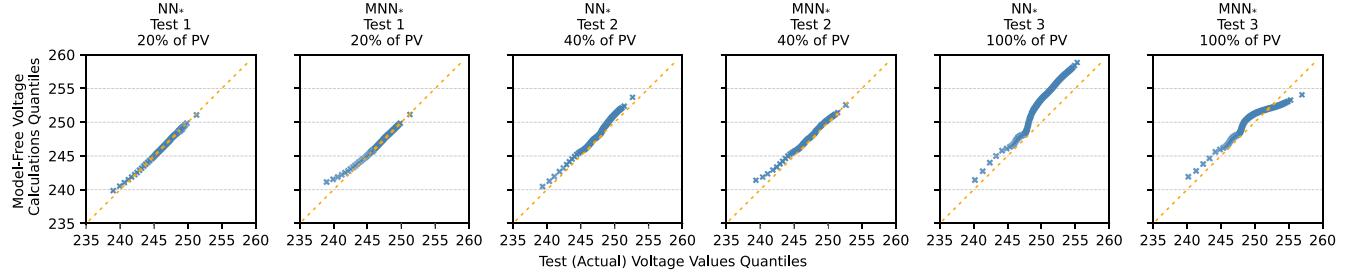


Fig. 7. Quantile-Quantile plots (solar hours only).

TABLE III  
MODEL-FREE VOLTAGE CALCULATIONS RESULTS

Metric	Test 1 (20% of PV penetration)			Test 2 (40% of PV penetration)			Test 3 (100% of PV penetration)		
	Solar hours	Nonsolar hours	Global	Solar hours	Nonsolar hours	Global	Solar hours	Nonsolar hours	Global
<i>NN*</i>	<b>RMSE V</b> <b>0.2807</b>	<b>0.2474</b>	<b>0.2681</b>	0.8724	<b>0.2477</b>	0.6970	2.9507	<b>0.2497</b>	2.2990
	<b>Av. Dev. V</b> <b>0.2010</b>	<b>0.1712</b>	<b>0.1892</b>	0.7454	<b>0.1714</b>	0.5181	2.5599	<b>0.1725</b>	1.6149
	<b>Max. Dev. V</b> <b>2.0106</b>	<b>2.0882</b>	<b>2.0882</b>	<b>2.6815</b>	<b>2.0974</b>	<b>2.6815</b>	7.3602	<b>2.1321</b>	7.3602
<i>MNN*</i>	0.4574	0.3354	0.4135	<b>0.7136</b>	0.3353	<b>0.5935</b>	<b>1.3737</b>	0.3352	<b>1.0883</b>
	0.2560	0.1801	0.2259	<b>0.5450</b>	0.1802	<b>0.4006</b>	<b>1.0765</b>	0.1810	<b>0.7220</b>
	3.7041	3.5673	3.7041	3.6107	3.5613	3.6107	<b>5.2559</b>	3.5479	<b>5.2559</b>

TABLE IV  
COMPUTATIONAL TIMES FOR 1,008 CONTINUOUS INSTANCES (30 MINUTES INTERVALS OVER 3 WEEKS)

Computational times in seconds	Test 1 (20% of PV)	Test 2 (40% of PV)	Test 3 (100% of PV)
Power flow simulation (OpenDSS)	20.6704	20.8970	21.0383
<i>NN*</i>	0.0776	0.0744	0.0787
<i>MNN*</i>	0.0842	0.0850	0.0847

almost all metrics as the PV penetration levels increase for both NN architectures (except for maximum deviation, Test 1, *NN\**). This is because of the unseen effects of the new PV penetrations (i.e., not part of the training). Nonetheless, from Fig. 4 it can also be noted that for *NN\** in Test 1 and Test 2 as well as for *MNN\** in all tests, 50% of global deviations (represented by the yellow bar) are maintained within a range of 1 V (e.g., within -0.1216 V and 0.8362 V for *MNN\** in Test 3). Overall, both NN architectures accomplish great accuracy among all tests, achieving, in the worst case, an average deviation of around 1.6 V, representing an alternative for distribution companies.

Based on the above and considering the maximum deviation as the critical metric for voltage calculations (given its effects on the resulting PV settings or PV hosting capacity), it can be concluded that for operation (short-term) applications (Test 1 and Test 2), i.e., when none or small changes in PV penetration with respect to the historical data are expected, *NN\** would be the most suitable NN. On the other hand, for planning (mid-, or long-term) applications (Test 3), i.e., when significant changes

in PV penetration are explored, *MNN\** would be the most suitable NN.

#### E. Computational Time: Model-Free vs Power Flows

Once trained, the final NN (either *NN\** or *MNN\**) becomes a proxy for the conventional power flow simulation required to calculate voltages. Due to the direct nature of the equations of the NN, the proposed methodology is not only accurate but also an extremely quick alternative for distribution companies. This is demonstrated by calculating voltages for all customers considering 1,008 instances (30 minutes intervals over 3 weeks) and using a standard commercial computer with an Intel i7 processor of 2.80 GHz and 16 GB of ram (DDR4). The results are shown in Table IV where power flow simulations (using OpenDSS [42]) and NN computational times are presented. As it can be observed, the power flow simulations can take more than 20 seconds, in all cases, whereas the model-free voltage calculations take less than 0.1 seconds. Consequently, the proposed approach can be used by

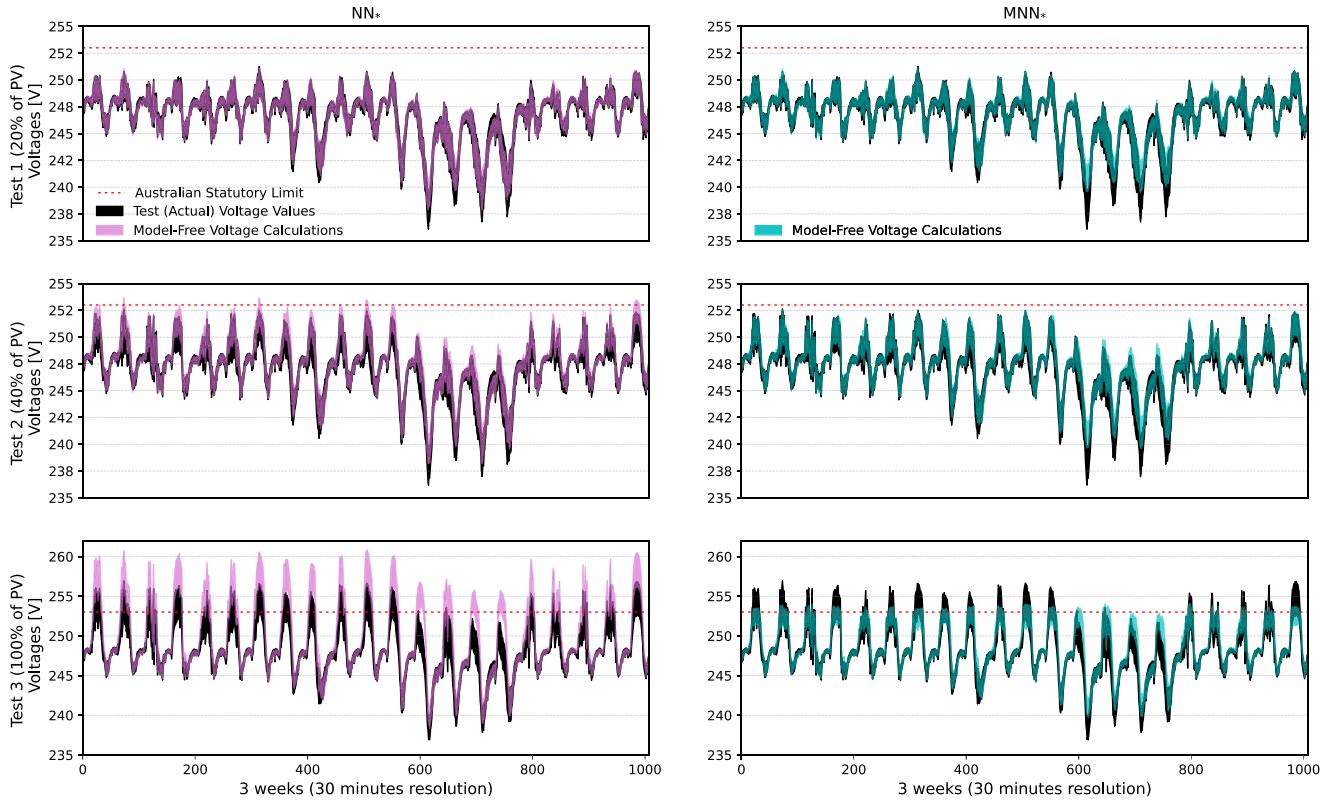


Fig. 8. Model-free voltage calculations for all customers in Test 1 (20% of PV) to Test 3 (100% of PV). (Left)  $NN_*$ , (Right)  $MNN_*$ .

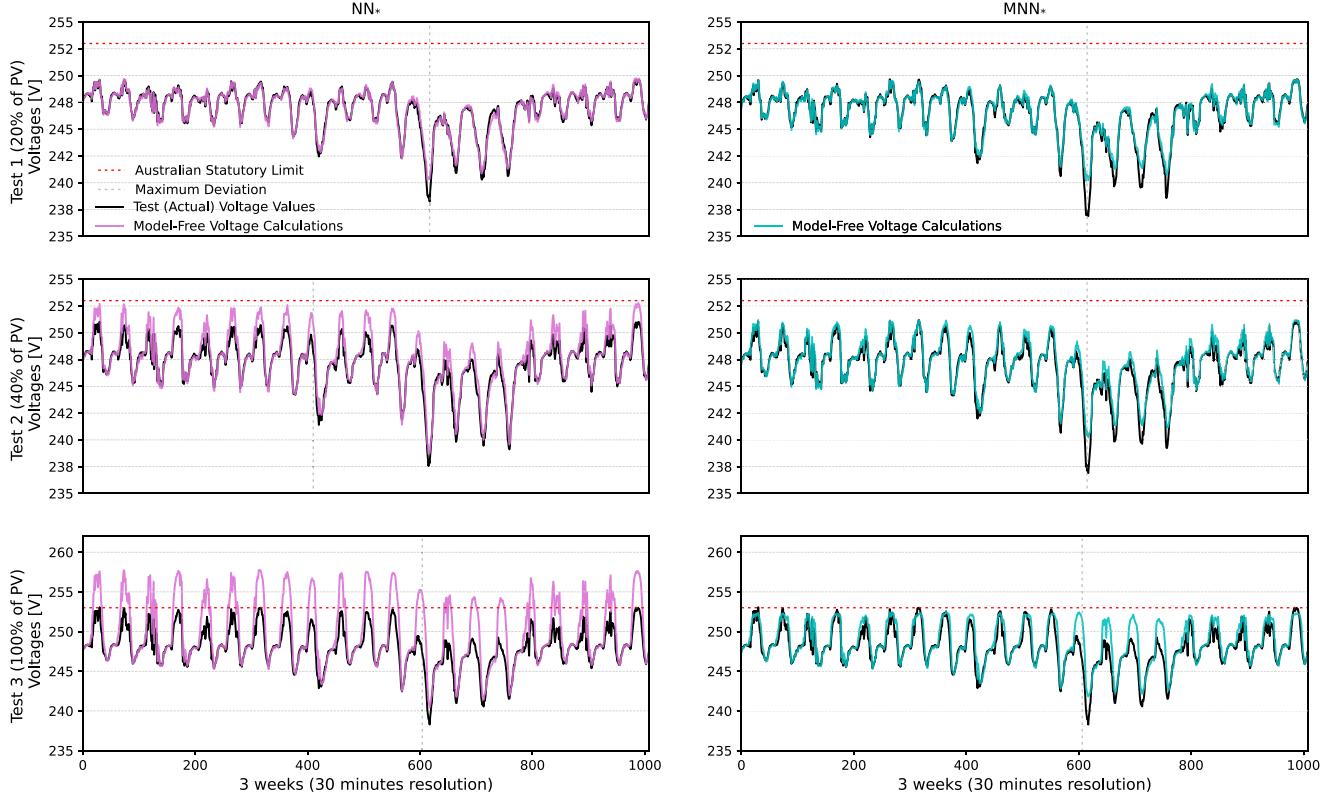


Fig. 9. Model-free voltage calculations for the worst performing customer in Test 1 (20% of PV) to Test 3 (100% of PV) (Left)  $NN_*$ , (Right)  $MNN_*$ .

distribution companies even in near real-time operation applications, being many times faster than conventional power flow analyses.

## V. DISCUSSION

For the proposed model-free voltage calculation methodology to capture the physics of the LV network, historical smart

meter data from all customers is required. However, in practice, missing measurements might occur due to, for instance, smart meters malfunctions. Hence, the historical data needs to be preprocessed to delete instances with missing measurements. Because of this, the historical data set that can be used to produce the final NN (either  $NN_*$  or  $MNN_*$ ) can end up being smaller than the original set.

Additionally, it is important to ensure that the historical smart meter data used to produce the final NN (either  $NN_*$  or  $MNN_*$ ) is representative of the latest condition of the studied LV network, i.e., the LV network have not undergone topological or any other type of changes within the considered period. However, in practice, topological changes due to, for instance, reconfiguration, reconductoring or new customers may occur. This, in turn, means that the  $NN_*$  (or  $MNN_*$ ) needs to be updated periodically with the latest data (e.g., every month). An alternative, once the proposed methodology is operationalized, is to flag topological or conductor changes as significant inconsistencies might appear in the measurements compared to what is expected by the  $NN_*$  (or  $MNN_*$ ). Upon such a flag, new historical data needs to be collected to produce a new  $NN_*$  (or  $MNN_*$ ). Such an update would also be required if using power flows instead.

It is also worth mentioning that, for the same period (e.g., a week), higher smart meter data resolution (e.g., 5 min instead of 30) will have more instances. Therefore, with higher resolution, the proposed methodology is likely to achieve an adequate accuracy using a reduced period. Hence, no significant impacts over the production times of the final NN (either  $NN_*$  or  $MNN_*$ ) are expected with higher resolution.

When the final NN (either  $NN_*$  or  $MNN_*$ ) is used for operation applications, it would require very recent data (e.g., 5 or 15 minutes ago). If a complete, adequate set of data is not available (due to missing or erroneous data issues), a preprocessing filtering process can be implemented to replace missing inputs by a previous value or even by 0. This, however, will impact the accuracy of the obtained voltage calculations as the interactions related to these data points will not be represented. For planning applications, on the other hand, these issues are not expected given that the inputs are defined by the scenario being investigated.

Nowadays, PV systems can have inverter functions activated (e.g., fixed export limits, volt-watt, volt-var, etc.). The effects of such functions are expected to be captured by the proposed methodology as they will be reflected in the historical data.

Even though the effectiveness of the proposed methodology is illustrated considering residential PV systems, the effects of EVs will also be captured.

Finally, ongoing research carried out by the authors [46] suggests that if the studied LV network have not undergone topological or conductor changes within the considered period, the NN (either  $NN_*$  or  $MNN_*$ ) can be used without requiring (or significantly benefiting) from regular updates. Additionally, it has been observed that the accuracy of the proposed methodology can increase considerably if voltages at the head of the LV network, i.e., at the secondary side of the distribution transformer, are available. This requires minimal adaptations on  $NN_i$ . However, this can be impractical for distribution companies as such measurements are largely unavailable.

## VI. CONCLUSION

Voltage calculations are at the core of many applications in distribution networks. However, the main challenge for distribution companies is that voltage calculations require power flow analyses (or similar) that need accurate electrical models which, in the context of low voltage (LV) networks, are not readily available for most companies around the world.

Taking advantage of smart meters, this paper proposes a scalable methodology to calculate voltages without electrical models (hence, electrical model-free) based on nonlinear regressions, using Neural Networks (NNs). Thus, a tailored NN is trained to capture the relationships among the historical per-phase smart meter data of customers (i.e.,  $P$ ,  $Q$ , and  $V$ ) and the studied LV network. Crucially, this paper accounts for the effects of the upstream MV network on LV network voltages and proposes two architectures of NNs that could be used for operation and planning applications: a conventional NN and a tailored multi-input multi-output NN.

The methodology is demonstrated using half-hourly data from one realistic LV network (146 customers) from Victoria, Australia, over three weeks and with 20% PV penetration. To cater for the upstream voltage fluctuations, an integrated MV-LV network with 3,400+ customers is considered. Results for different weeks with the same and higher PV penetrations demonstrate an average accuracy, in the worst case, of just 1.6 V; showing that, in the absence of LV electrical models, the methodology can be used by distribution companies to carry out accurate voltage calculations.

Specifically, it is shown that, for operation (short-term) applications, e.g., determine specific settings (PV curtailment or EV throttling, reactive power absorption or maximum import/export limits) to ensure compliance with statutory voltage limits, the most suitable NN architecture would be a conventional single-hidden layer NN. On the other hand, for planning (mid-, or long-term) applications, e.g., determine PV/EV hosting capacity or to assess PV/EV connection requests, the most suitable architecture would be the proposed multi-input multi-output NN. Additionally, it is shown that, once trained, the final NN (either  $NN_*$  or  $MNN_*$ ) becomes a proxy for the conventional power flow analyses required to calculate voltages and that can be much faster, representing an attractive alternative for distribution companies unable to produce accurate electrical models.

Further research is needed in terms of the performance of the proposed methodology considering the inherent noise embedded in real smart meter measurements. This is being investigated by the authors within the context of an ongoing project in Victoria, Australia [46]. Additionally, ongoing research of the authors is assessing several other regression methods which may also perform well (e.g., ridge regression, Gaussian process regression, support vector regression, etc.) and must be investigated.

## APPENDIX DEEPER NEURAL NETWORKS

To maintain a manageable amount of hyperparameters combinations within the k-fold cross validation process, the number of hidden layers for  $NN_i$  is fixed at 1. Nonetheless,

TABLE V  
MODEL-FREE VOLTAGE CALCULATIONS RESULTS FOR  $NN_*$  WITH MORE HIDDEN LAYERS

	Metric	Test 1 (20% of PV)	Test 2 (40% of PV)	Test 3 (100% of PV)
$NN_*$ 2 HL	<b>RMSE V</b>	0.4766	0.7108	1.6384
	<b>Av. Dev. V</b>	0.2716	0.4806	1.1031
	<b>Max. Dev. V</b>	3.7664	4.0334	6.8662
$NN_*$ 3 HL	<b>RMSE V</b>	0.4122	0.6342	1.3441
	<b>Av. Dev. V</b>	0.2428	0.4428	0.8741
	<b>Max. Dev. V</b>	3.7365	3.6013	6.2967
$NN_*$ 4 HL	<b>RMSE V</b>	0.5875	0.7693	1.2750
	<b>Av. Dev. V</b>	0.3124	0.5055	0.8348
	<b>Max. Dev. V</b>	4.5428	4.5344	6.0250
$NN_*$ 5 HL	<b>RMSE V</b>	0.5984	0.7103	1.3259
	<b>Av. Dev. V</b>	0.3103	0.4592	0.8754
	<b>Max. Dev. V</b>	4.8361	4.7427	5.9464
$NN_*$ 6 HL	<b>RMSE V</b>	0.6485	0.7459	1.3337
	<b>Av. Dev. V</b>	0.3307	0.4732	0.8840
	<b>Max. Dev. V</b>	4.9529	4.8214	5.8670

NNs with a higher number of hidden layers could capture more complex relationships and achieve a better performance in complex problems [36]. This is assessed in Table V, where model-free voltage calculations results are presented for  $NN_*$  with 2 to 6 hidden layers, i.e., considering the same hyperparameters as  $NN_*$  with more layers. Thus, the number of neurons in the first hidden layer is repeated as appropriate, and all the remaining hyperparameters of  $NN_*$  are maintained.

From Table III and Table V, it can be noted that despite the addition of layers can help  $NN_*$  to improve its generalization capabilities,  $MNN_*$  achieves the most accurate results in Test 3 (planning). Besides, such improvements come at the expense of longer training periods and larger search spaces to be considered for the k-fold cross validation process, as well as an accuracy decrease, principally in terms of maximum deviation, in Test 1 and Test 2 (operation).

## REFERENCES

- [1] (IEA, Paris, France). *World Energy Outlook 2021*. (2021). [Online]. Available: <https://www.iea.org/reports/world-energy-outlook-2021>
- [2] (IEA, Paris, France). *Global EV Outlook 2021*. (2021). [Online]. Available: <https://www.iea.org/reports/global-ev-outlook-2021>
- [3] A. Navarro-Espinosa and L. F. Ochoa, "Probabilistic impact assessment of low carbon technologies in LV distribution systems," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2192–2203, May 2016.
- [4] J. Quirós-Tortós, L. F. Ochoa, S. W. Alnaser, and T. Butler, "Control of EV charging points for thermal and voltage management of LV networks," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3028–3039, Jul. 2016.
- [5] P. Quesada, A. Arguello, J. Quirós-Tortós, and G. Valverde, "Distribution network model builder for OpenDSS in open source GIS software," in *Proc. IEEE PES Transm. Distrib. Conf. Exposit. Latin America (PES T D-LA)*, 2016, pp. 1–6.
- [6] A. Navarro-Espinosa, L. F. Ochoa, R. Shaw, and D. Randles, "Reconstruction of low voltage distribution networks: From GIS data to power flow models," in *Proc. 23rd Int. Conf. Electricity Distrib. (CIRED)*, Lyon, France, 2015, p. 1273.
- [7] M. Khodayar, G. Liu, J. Wang, and M. E. Khodayar, "Deep learning in power systems research: A review," *CSEE J. Power Energy Syst.*, vol. 7, no. 2, pp. 209–220, Mar. 2021.
- [8] O. A. Alimi, K. Ouahada and A. M. Abu-Mahfouz, "A review of machine learning approaches to power system security and stability," *IEEE Access*, vol. 8, pp. 113512–113531, 2020.
- [9] L. Duchesne, E. Karangelos, and L. Wehenkel, "Recent developments in machine learning for energy systems reliability management," *Proc. IEEE*, vol. 108, no. 9, pp. 1656–1676, Sep. 2020.
- [10] J. Xie, I. Alvarez-Fernandez, and W. Sun, "A review of machine learning applications in power system resilience," in *Proc. IEEE Power Energy Soc. General Meeting (PESGM)*, 2020, pp. 1–5.
- [11] R.-A. Tirnovan and M. Cristea, "Advanced techniques for fault detection and classification in electrical power transmission systems: An overview," in *Proc. 8th Int. Conf. Modern Power Syst. (MPS)*, 2019, pp. 1–10.
- [12] C. M. Furse, M. Kafal, R. Razzaghi, and Y.-J. Shin, "Fault diagnosis for electrical systems and power networks: A review," *IEEE Sensors J.*, vol. 21, no. 2, pp. 888–906, Jan. 2021.
- [13] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan, and F. Bu, "A survey on state estimation techniques and challenges in smart distribution systems," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2312–2322, Mar. 2019.
- [14] N. Ahmad, Y. Ghadi, M. Adnan, and M. Ali, "Load forecasting techniques for power system: Research challenges and survey," *IEEE Access*, vol. 10, pp. 71054–71090, 2022.
- [15] B. K. Sovacool, A. Hook, S. Sareen, F. W. Geels, "Global sustainability, innovation and governance dynamics of national smart electricity meter transitions," *Global Environ. Change*, vol. 68, May 2021, Art. no. 102272.
- [16] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, May 2019.
- [17] S. J. Pappu, N. Bhatt, R. Pasumarthy, and A. Rajeswaran, "Identifying topology of low voltage distribution networks based on smart meter data," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5113–5122, Sep. 2018.
- [18] J. Peppanen, S. Grijalva, M. J. Reno, and R. J. Broderick, "Distribution system low-voltage circuit topology estimation using smart metering data," in *Proc. IEEE/PES Transm. Distrib. Conf. Exposit. (T D)*, 2016, pp. 1–5.
- [19] A. Simonovska and L. F. Ochoa, "Phase grouping in PV-rich LV feeders: Smart meter data and unconstrained k-means," in *Proc. IEEE Madrid PowerTech*, 2021, pp. 1–6.
- [20] Y. Wang, M. Z. Liu, and L. F. Ochoa, "Assessing the effects of DER on voltages using a smart meter-driven three-phase LV feeder model," *Electr. Power Syst. Res.*, vol. 189, Dec. 2020, Art. no. 106705.
- [21] V. C. Cunha, W. Freitas, F. C. L. Trindade, and S. Santoso, "Automated determination of topology and line parameters in low voltage systems using smart meters measurements," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5028–5038, Nov. 2020.
- [22] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang, "Data-driven power flow linearization: A regression approach," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2569–2580, May 2019.
- [23] A. F. Bastos, S. Santoso, V. Krishnan, and Y. Zhang, "Machine learning-based prediction of distribution network voltage and sensor allocation," in *Proc. IEEE Power Energy Soc. General Meeting (PESGM)*, 2020, pp. 1–5.
- [24] P. Pareek, W. Yu, and H. D. Nguyen, "Optimal steady-state voltage control using Gaussian process learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7017–7027, Oct. 2021.
- [25] Y. Chen, Y. Shi, and B. Zhang, "Data-driven optimal voltage regulation using input convex neural networks," *Electr. Power Syst. Res.*, vol. 189, Dec. 2020, Art. no. 106741.

- [26] D. Cao et al., "Model-free voltage control of active distribution system with PVs using surrogate model-based deep reinforcement learning," *Appl. Energy*, vol. 306, Jan. 2022, Art. no. 117982.
- [27] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2082–2092, May 2021.
- [28] S. Balduin, T. Westermann, and E. Puiutta, "Evaluating different machine learning techniques as surrogate for low voltage grids," *Energy Inform.*, vol. 3, p. 24, Oct. 2020.
- [29] R. Pellerej, T. Trouillon, C. Benoit, Q. Garnier, and A. Versyp, "Impact of flexibility on low voltage network's hosting capacity—Belgium experimentation," in *Proc. CIRED Berlin Workshop*, Berlin, Germany, 2020, pp. 209–212.
- [30] J. Zhang, C. Y. Chung, and L. Guan, "Noise effect and noise-assisted ensemble regression in power system online sensitivity identification," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2302–2310, Oct. 2017.
- [31] E. L. da Silva, A. M. N. Lima, M. B. D. R. Corrêa, M. A. Vitorino, and L. T. Barbosa, "Data-driven sensitivity coefficients estimation for cooperative control of PV inverters," *IEEE Trans. Power Del.*, vol. 35, no. 1, pp. 278–287, Feb. 2020.
- [32] C. Mugnier, K. Christakou, J. Jaton, M. De Vivo, M. Carpita, and M. Paolone, "Model-less/measurement-based computation of voltage sensitivities in unbalanced electrical distribution networks," in *Proc. Power Syst. Comput. Conf. (PSCC)*, 2016, pp. 1–7.
- [33] V. Bassi, L. Ochoa, and T. Alpcan, "Model-free voltage calculations for PV-rich LV networks: Smart meter data and deep neural networks," in *Proc. IEEE Madrid PowerTech*, 2021, pp. 1–6.
- [34] V. Bassi, L. F. Ochoa, and T. Alpcan, "Calculating voltages without electrical models: Smart meter data and neural networks," in *Proc. CIRED 26th Int. Conf. Exhibit. Electricity Distrib.*, 2021, pp. 1547–1551.
- [35] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [37] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
- [38] D. P. Kingma and J. L. Ba, "ADAM: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14.
- [39] "Australian PV market since 2001." Australian PV Institute. 2022. [Online]. Available: <https://pv-map.apvi.org.au/analyses>
- [40] *Export Limits for Embedded Generators up to 200 kVA Connected at Low Voltage*, AusNet Serv., Southbank, VIC, Australia, 2017.
- [41] A. T. Procopiou et al., "Integrated MV-LV network modelling for DER studies," in *Proc. CIRED Berlin Workshop*, Berlin, Germany, 2020, pp. 274–277.
- [42] "OpenDSS—Open distribution system simulator." EPRI. 2020. [Online]. Available: <http://smartgrid.epri.com/SimulationTool.aspx>
- [43] F. Chollet et al. "Keras." GitHub. 2015. [Online]. Available: <https://github.com/fchollet/keras>
- [44] "Python software foundation." Python. Accessed: Jan. 13, 2022. [Online]. Available: <https://www.python.org/>
- [45] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Des. Implement.*, 2016, pp. 265–283.
- [46] "Model-free operating envelopes at NMI level." The University of Melbourne. Accessed: Apr. 20, 2022. [Online]. Available: <https://electrical.eng.unimelb.edu.au/power-energy/projects/model-free-operating-envelopes>



**Luis F. Ochoa** (Senior Member, IEEE) received the B.Eng. degree from the Universidad Nacional de Ingeniería, Lima, Peru, in 2000, and the M.Sc. and Ph.D. degrees from Universidade Estadual Paulista, Ilha Solteira, Brazil, in 2003 and 2006, respectively. He is currently a Professor of Smart Grids and Power Systems with The University of Melbourne, Melbourne, VIC, Australia. Previously, from 2011 to 2021, he was full and part-time with The University of Manchester, U.K. From 2007 to 2010, he was a Research Fellow in Energy Systems with The University of Edinburgh, U.K. His expertise in network integration of distributed energy resources and smart grids as well as his extensive portfolio of industrial and academic projects have led to more than 210 research papers, more than 80 technical reports, and two patents.



**Tansu Alpcan** (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2006. He has (co)authored more than 150 journal and conference articles as well as the book titled *Network Security: A Decision and Game Theoretic Approach* (Cambridge University Press, 2011). He co-edited the book titled *Mechanisms and Games for Dynamic Spectrum Allocation* (Cambridge University Press, 2014). He has worked as a Senior Research Scientist with Deutsche Telekom Laboratories, Berlin, Germany, from 2006 to 2009, and as an Assistant Professor (Juniorprofessur) with Technical University Berlin from 2009 to 2011. He is currently with the Department of Electrical and Electronic Engineering, The University of Melbourne as a Professor and a Reader. His research interests include applications of control, optimization, game theories, and machine learning to security and resource allocation problems in communications, smart grids, and Internet of Things. He chaired or was an Associate Editor, TPC chair, or TPC member of several prestigious IEEE workshops, conferences, and journals.



**Vincenzo Bassi** (Graduate Student Member, IEEE) received the B.Sc. degree and the Professional degree in electrical engineering from the University of Chile, Chile, in 2015 and 2016, respectively. He is currently a Ph.D. candidate and a Research Associate in Smart Grids at The University of Melbourne, Australia. Previously, he was an Energy Consultant with SPEC, Chile, and a Research and Development Engineer with the Energy Centre of the Faculty of Physical and Mathematical Sciences, University of Chile. His research is focused on data-driven operation and planning applications in DER-rich distribution networks.



**Christopher Leckie** is a Professor with the School of Computing and Information Systems, University of Melbourne. He has over two decades of research experience in artificial intelligence and data mining, having led research teams with Telstra Research Laboratories, National ICT Australia, and the University of Melbourne. His research on using data mining for anomaly detection, fault diagnosis, cyber-security, and the life sciences has led to a range of operational systems used in industry, as well as over 300 articles published in leading international conferences and journals.