Neural Network Design for Impedance Modeling of Power Electronic Systems Based on Latent Features

Yicheng Liao, *Member, IEEE*, Yufei Li, *Member, IEEE*, Minjie Chen, *Senior Member, IEEE*, Lars Nordström, *Senior Member, IEEE*, Xiongfei Wang, *Senior Member, IEEE*, Prateek Mittal, *Senior Member, IEEE*, and H. Vincent Poor, *Fellow, IEEE*

***Abstract*—Frequency-domain analysis is important for small-signal dynamic studies of power electronics-based power systems. However, the frequency-domain model of power electronic system needs to be linearized around a specific operating condition. Conventional analysis hence requires measurement or identification of these frequency-domain models repeatedly at many operating points due to the wide operation range of the power systems, which brings significant computation and data burden. This paper addresses this challenge by developing a deep learning approach using multilayer feedforward neural networks (FNNs) for frequency-domain modeling of power electronic systems, and particularly, focusing on the impedance modeling. It can train converter impedance models that are continuous across certain ranges of operating points. Distinguished from the prior neural network designs relying on trials and errors, this article proposes to design the FNN based on latent features of power electronic systems, i.e., the number of system poles and zeros. To further investigate the impacts of data quantity and quality, learning procedures from a small dataset are developed, and clustering is used to reveal insights into multivariable sensitivity, which helps to improve the data quality. The proposed approaches for the FNN design and learning are finally validated by case studies on a power electronic converter.**

***Index Terms*—Multilayer perceptron, deep learning, power electronics, frequency-domain model, latent features, clustering.**

I. Introduction

DYNAMIC modeling and analysis is of primary importance for stable operation of electric power systems. In recent years, with the increased penetration of renewable energy sources and the diversity of loads, more power electronic converters are integrated in modern power systems for flexible and efficient power generation, transmission, and utilization [1], [2]. Compared to legacy power systems, the control of power electronic systems is more distributed and less transparent, which poses more challenges to the dynamic modeling and control of modern power systems [3].

Frequency-domain analysis is an important approach to studying the small-signal dynamics of large-scale networked power electronic systems. Frequency-domain analysis enables black-box modeling of a subsystem at its connection terminals and provides more efficient and insightful analysis compared to time-domain simulations [4]-[6]. However, frequency-domain models are linearized around specific operating conditions [7]. To assure reliable operation of power systems, dynamic analysis under a wide range of operating conditions are needed, which requires the frequency-domain models under various operating conditions. Methods for identifying black-box frequency-domain models based on measurement have been reported [8]; however, they can only be implemented at a single operating condition each time. Whenever the system operating condition is changed, the measurement needs to be repeated. Therefore, it is desired to obtain a frequency-domain model that can cover multiple operating conditions, in order to accelerate the dynamic analysis of power systems. Although it is possible to acquire such a model through analytical modeling [9], [10], it requires detailed and accurate control information, which is practically impossible because control algorithms of power electronic systems are usually kept confidential by their manufacturers.

Data-driven approaches provide a suitable solution to this type of problems. It is worth noting that although parametric identification [11] is also a data-driven approach that can derive an analytical transfer function model for frequency-domain analysis, the identified model is only an *s*-domain function, where *s* is the Laplace variable, which does not characterize the dependence of system operating conditions. The system operating conditions are determined by multiple variables; thus, the desired model is essentially a multivariable nonlinear function. Since the model expression in relation to operating conditions is often unknown, there is no deterministic method to select a specific parametric nonlinear model for regression analysis. In contrast, with sufficiently large network size and dataset, neural networks can approximate any continuous function [12], which thus provide a promising approach to solving this problem.

There have been recent attempts to apply neural networks for frequency-domain modeling of power electronic systems. Some studies have developed neural networks to train time-domain surrogate models and then transform them into the frequency domain for further studies; examples include

This work was supported by the C3.ai Digital Transformation Institute as a collaboration between KTH Royal Institute of Technology and Princeton University.

Y. Liao was with Division of Electric Power and Energy Systems, KTH Royal Institute of Technology, Stockholm 10044, Sweden. She is now with Grid Analysis, Energinet (Danish Transmission System Operator), Fredericia 7000, Denmark (e-mail: YLI@energinet.dk).

L. Nordström and X. Wang are with Division of Electric Power and Energy Systems, KTH Royal Institute of Technology, Stockholm 10044, Sweden. (e-mail: {larsno, xiongfei}@kth.se)

Y. Li, M. Chen, P. Mittal, and H. V. Poor are with Department of Electrical and Computer Engineering, Princeton University, New Jersey 08544, USA. (e-mail: {yl5385, minjie, pmittal, poor}@princeton.edu).

recurrent neural networks [13] and nonlinear autoregressive models with exogenous inputs (NARX) [14], [15]. Such models show good training performance for time series data, yet the time-domain data still needs to be designed carefully to capture sufficient frequency-domain dynamics, which makes the model training and parameter design more difficult. Another work directly considers training the model in the frequency domain using feedforward neural networks (FNNs) [16]. However, this study oversimplifies the power electronic systems by considering the operating-point dependence only related to a single variable. The operating points of actual systems are generally dependent on multiple variables. Using a single-layer FNN requires significant amount of data and the optimal design of FNN needs to be found based on trials and errors, which also brings much design and training effort. Thus, it is very challenging to apply the single-layer FNNs in practical applications.

To deal with the challenges, this work focuses on the frequency-domain model training of power electronic systems using FNNs. Differing from [16], this work utilizes the multilayer FNNs and develops a deep learning approach that can be generally applied to any converter system. The unique contributions of this work include

- Designing a multi-layer FNN to train converter impedance models based on the system's latent features. The number of layers is determined by the transfer function calculation properties, and the number of neurons can be designed according to pole and zero numbers, which is much more easier than designing a single-layer FNN.
- Developing learning procedures that can be implemented with small datasets, which is also a significant advantage compared with using a single-layer FNN.
- Investigating the benefit of using clustering to improve the data quality.

Section II formulates the learning problem. A method for designing the multi-layer FNN based on latent features of the converter system is proposed in Section III. Section IV details the application of the multi-layer FNN, by developing learning procedures based on small datasets and proposing to use clustering approach for data quality improvement. Section V presents the case studies by training an example converter system, to verify the proposed methods. Section VI finally concludes this paper.

II. Problem Formulation

A. Impedance-Based Analysis

Fig. 1 shows a typical diagram of a modern power system, where power electronic converters are pervasive as interfaces for various types of energy conversion and energy utilization. Power electronic converters can be controlled flexibly, yet their control algorithms are in general confidential. To analyze the small-signal stability of large-scale power electronics-based power systems, frequency-domain modeling by using impedance or admittance models, i.e., Z(s) or Y(s), has become a trend [17]. Since these models can be rapidly obtained by frequency-scan measurement at the points of connection (PoCs) of converter systems, they are suitable to analyze such black-box systems. The admittance model can be calculated by the inverse of the impedance model; and thus, the following model elaboration will use the impedance model for clarity.

B. Learning Problem and Assumptions

The impedance model of a converter system is a linearized description of the system under a certain operating condition. A power converter can operate under a wide range of operating conditions. If the impedance model that covers the range of operating conditions can be rapidly estimated with sufficient accuracy, the efficiency of system-level stability analysis can be improved significantly.

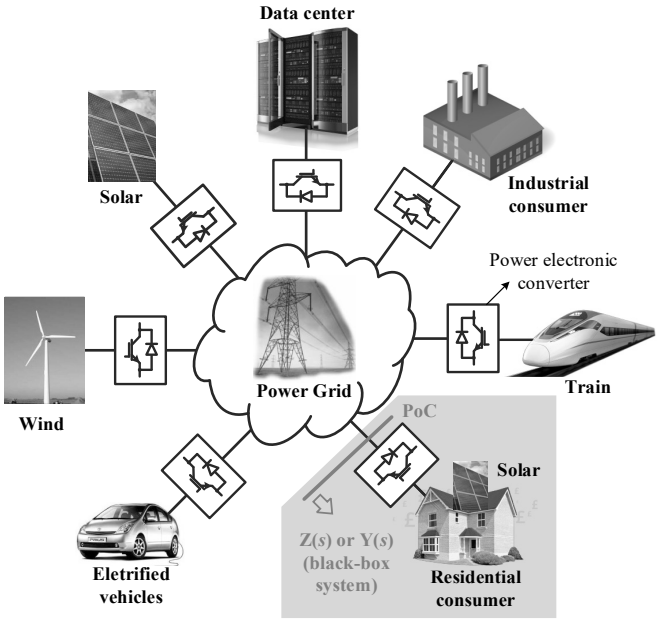


Fig. 1. Power electronics-based power systems.

The impedance model of a converter system can take different forms depending on modeling assumptions and modeling reference frames [9]. In this work, it is assumed that the impedance model is represented at the AC interface of three-phase converter systems under balanced grid conditions. For AC converter systems, the operating condition involves the voltage magnitude (V), the active and reactive power flows (P and Q) at the PoC. Thus, a general impedance model can be represented in the synchronous reference (dq) frame using a multi-input multi-output model [7], i.e.,

$$\mathbf{Z}(s,V,P,Q)=\begin{bmatrix} Z_{dd}(s,V,P,Q) & Z_{dq}(s,V,P,Q) \\ Z_{qd}(s,V,P,Q) & Z_{qq}(s,V,P,Q) \end{bmatrix}$$
$$\Downarrow s=j\omega \qquad (1)$$
$$\mathbf{Z}(j\omega,V,P,Q)=\begin{bmatrix} Z_{dd}(j\omega,V,P,Q) & Z_{dq}(j\omega,V,P,Q) \\ Z_{qd}(j\omega,V,P,Q) & Z_{qq}(j\omega,V,P,Q) \end{bmatrix}$$

where s is the Laplace variable, and V, P, Q are operating points under a certain operating condition. s represents the frequency dependence because it can be substituted by $s=j\omega$ to derive the frequency-domain model [18].

It is worth noting that although (1) only gives an example using AC impedance representation, the model order and its

dependence of operating points can be easily modified to characterize DC systems [19]. Moreover, the frequency domain model can also be represented by other transfer functions if other input-output relationships are of interest.

Eq. (1) indicates that the impedance model is essentially a multivariable nonlinear model, and thus, neural networks can be used for learning approximations to this model.

III. Neural Network Design Based on Latent Features

A neural network design guideline for the converter impedance modeling is now proposed based on latent features of the converter system.

A. Input and Output Definition

To realize a learning task, it is important to define first the input and output data. It is known from (1) that the impedance model is related to ω, V, P, Q. Thus, these variables can be chosen as the input variables, which are real numbers. The output is the impedance value, which is a complex number. We can define the output to have two dimensions, using real and imaginary parts of the impedance value, i.e., the resistance and reactance of the electric system, or the magnitude and phase of the impedance through a polar transformation. We use the real and imaginary parts of the impedance as the data output for the following reasons:

- The frequency-domain analysis is conducted based on complex numbers. A learned neural network that can predict the real and imaginary parts of impedance models is adequate for system stability analysis.
- The polar transformation from real and imaginary parts into magnitude and phase is only for easier visualization of the model on a Bode diagram. This transformation involves additional nonlinear calculations, which can increase the complexity of neural network design and the effort of training.
- The real and imaginary parts of the complex impedance values represent the resistance and reactance of the power converter system, which shed clear physical insight in electromagnetic simulations.

B. Neural Network Selection and Design

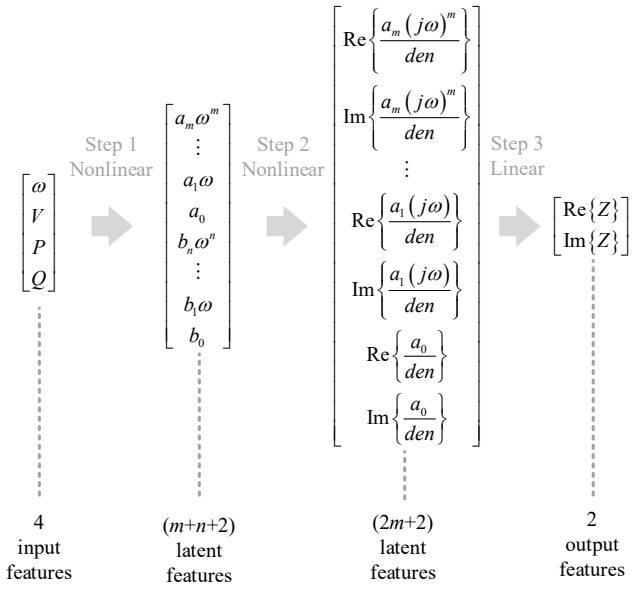
The converter's impedance can be uniquely determined by a set of input variables, which follows a forward calculation process. Further, the input variables are independent of each other. Hence, FNN is chosen in this work.

In principle, a neural network with one hidden layer can sufficiently represent any continuous function [12]. However, it requires the use of a sufficiently large number of neurons in the hidden layer, potentially as large as the number of training samples [20], which incurs substantial computation burden in training. In contrast, neural networks consisting of more hidden layers but smaller number of neurons can achieve better training performance with much smaller amount of data, but the structure of deep neural networks needs to be properly designed.

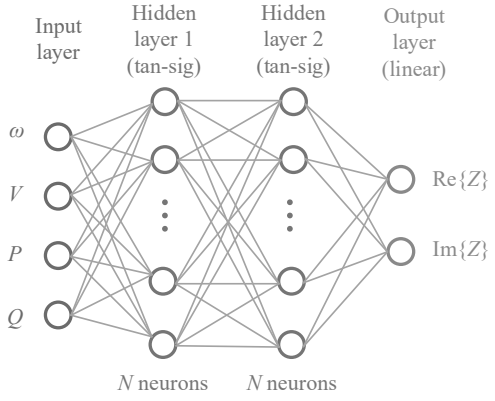
Designing optimal FNNs is a non-trivial task and it requires complicated sensitivity analysis [21]. It has been realized in recent years that if the domain knowledge of the physical system can be utilized in the neural network design, we may end up with a better designed network, which requires less data requirement yet achieves better training performance [22], [23]. Since physical systems can usually be decomposed into sequential steps, neural networks can achieve good training performance if they can be designed following such sequential steps [24]. Therefore, the FNN is designed by decomposing the learning problem into multiple sequential steps.

The impedance model is essentially a frequency-domain model. Its transfer function order is related to the differential equation order of the time-domain model; therefore, the frequency-domain model also includes information of the physical system. The sequential decomposition is directly considered based on transfer function calculation process in the frequency domain.

It is known that any element in the impedance model of (1) can be represented by the following general form:



(a)



(b)

Fig. 2. FNN design based on latent features. (a) impedance calculation process decomposed into sequential steps; (b) FNN design based on decomposed steps and numbers of latent features.

$$Z(s,V,P,Q)=\frac{a_m s^m+\cdots+a_1 s+a_0}{b_n s^n+\cdots+b_1 s+b_0}$$
$$\Downarrow s=j\omega \quad , \qquad (2)$$
$$Z(j\omega,V,P,Q)=\frac{a_m (j\omega)^m+\cdots+a_1 (j\omega)+a_0}{b_n (j\omega)^n+\cdots+b_1 (j\omega)+b_0}$$

where the polynomial coefficients $\{a_k\}$, $\{b_k\}$ in the numerator and denominator are determined by the operating points through unknown nonlinear relationships. By substituting $s=j\omega$, the impedance value can be calculated by several sequential steps, as decomposed in Fig. 2(a). Step 1 calculates all the terms in the numerator and denominator regarding different power of angular frequency, which is a nonlinear process. Step 2 further divides the numerator terms by the denominator (*den*) and then calculates the real and imaginary parts, which is another nonlinear process. Step 3 finally calculates the real and imaginary parts of the impedance, which is only a linear calculation based on the output of Step 2. By decomposing the entire process into these sequential steps, the number of latent features in the middle of calculations can be determined by the number of poles (n) and the number of zeros (m) of the system. Step 1 yields (m+n+2) latent features, and Step 2 yields (2m+2) latent features.

Based on the decomposed simple steps and the number of latent features, a multilayer FNN can be design as shown in Fig. 2(b). Although the FNN does not calculate in the same way as the original model, as it calculates through weighted sums, biases and activation functions, the number of latent features can be used as an indicator for designing the number of neurons in hidden layers [25]. There are two decomposed nonlinear steps, thus, two hidden layers are designed accordingly. In each nonlinear step, the number of latent features is no more than

$$N=2\max\{m,n\}+2. \qquad (3)$$

Thus, two hidden layers can be simply designed with N neurons in each layer. Here, designing both hidden layers with N neurons results in more hyper parameters in the FNN, but it in turn simplifies the structure design. In actual applications, the exact numbers of poles and zeros are unknown because the converters are usually black-box systems, which needs a "guess". Considering only one parameter N in the FNN design makes it easier to apply this approach for black-box systems. Since the two hidden layers represent two nonlinear steps, and this is a regression problem where the calculated result in each step can be both positive and negative, the hyperbolic tangent sigmoid (tan-sig) function is used for activation in hidden layers. Step 3 is a linear step, thus the output layer in the neural network is designed as a linear layer.

It is noted that although the number of poles and zeros is usually unknown for a black-box converter system, the possible range of the pole and zero numbers is limited. A typical converter system usually has at most three cascaded control loops: outer loop – voltage loop – current loop, and each control loop has at most one integrator. The plant of the converter system usually has the differential order no more than three. Although there can be different combinations of control loops and converter plant, the numbers of system poles and zeros are still within a certain range. Then, by comparing training performance with different values of N within a certain range based on trial and error, an optimal N per layer that achieves the best training performance can be found for learning a specific system.

C. Optimization Algorithm

Training a neural network is essentially an optimization problem that minimizes the model error, and this in turn can be solved by different optimization algorithms. The commonly used optimization algorithms for training neural networks can generally be classified into two categories which are based on gradient descent and Newton's method, respectively [26], [27]. The former is a first-order iterative algorithm, which can find the local optimum in the steepest-descent direction. However, it requires a careful tuning of learning rate and momentum constant throughout the training process for certain problems in order to find the global optimum. Moreover, training with the gradient descent-based algorithm converges relatively slowly. Newton's method considers both first- and second-order derivatives to find the optimum using a quadratic approximation, which can better find the global optimum for nonlinear problems, but its computational burden is much higher.

The Levenberg-Marquardt algorithm is a method that combines features of gradient descent and Newton's method. It acts more like a gradient-descent method when the parameters are far from optima and acts more like the Newton's method when the parameters are close to optima [27]. Thus, this algorithm can achieve faster convergence than gradient descent, but incurs a lower computational burden than Newton's method. It is suitable for learning small-scale systems whose neural networks only have a few hundred weights. Therefore, the Levenberg-Marquardt algorithm is selected as a candidate for optimization in our problem, since the studied system (a single converter system) is a small-scale system.

D. Loss Function

The mean squared error is used as loss function in this work to train the neural network. If the mean square error bound is defined as MSE, its relation to the impedance magnitude error can be interpreted as follows:

$$\begin{aligned}&\mathrm{Error}_{|\mathrm{Re}\{Z\}|}\le\sqrt{\mathrm{MSE}},\ \mathrm{Error}_{|\mathrm{Im}\{Z\}|}\le\sqrt{\mathrm{MSE}}\\&\Rightarrow \mathrm{Error}_{|Z|}\le\sqrt{2\mathrm{MSE}}\end{aligned}. \qquad (4)$$

For the learning of the impedance model, MSE can be defined based on the minimum required magnitude of the impedance model for a certain case, to ensure that the error of the trained neural network is smaller than an acceptable threshold.

IV. Learning Procedures with Small Dataset

Effective learning relies on having sufficient data. Learning from a sufficiently large dataset is always desirable; however, obtaining such a dataset is often difficult in practice. This section discusses the data size problem for learning the

impedance model of converter systems, by considering as small dataset as possible.

A. Learning Procedure

The challenge in frequency-domain modeling of converter impedances mainly lies in the dependence of operating points, thus, impedance data under multiple operating points needs to be obtained for learning. However, the amount of data that is sufficient for learning a problem is usually unknown before learning. Therefore, a learning procedure starting from a dataset under a small number of operating points (OPs) is proposed in Fig. 3.

To ensure a good training performance, the total dataset needs to be split into a training set, a validation set and a test set [28]. The training set is used to train the neural network based on the training loss, which is calculated as the MSE of the model. The validation set is used to calculate the validation loss, which is monitored during the learning process and compared with the training loss, to assess whether the training is successful and whether the dataset is sufficient. If the validation loss is significantly increased during the learning process, it is an indication that the training is unstable and the data is insufficient. If the training finally ends when reaching the goal of MSE, yet the validation loss is much higher than the training loss, it is also an indication that the dataset is insufficient and needs to be enlarged. Therefore, a threshold ε can be set for the ratio of validation loss to training loss, to check whether the dataset is sufficient or not, as shown in Fig. 3.

It is worth noting that although the flowchart in Fig. 3 mainly focuses on the dataset regarding change of OPs, it works also for the dataset regarding change of frequencies. Here, the frequency resolution is not focused upon, since the impedance measurement usually scans sufficient frequency points to ensure that the frequency dependence is characterized sufficiently [8], [16].

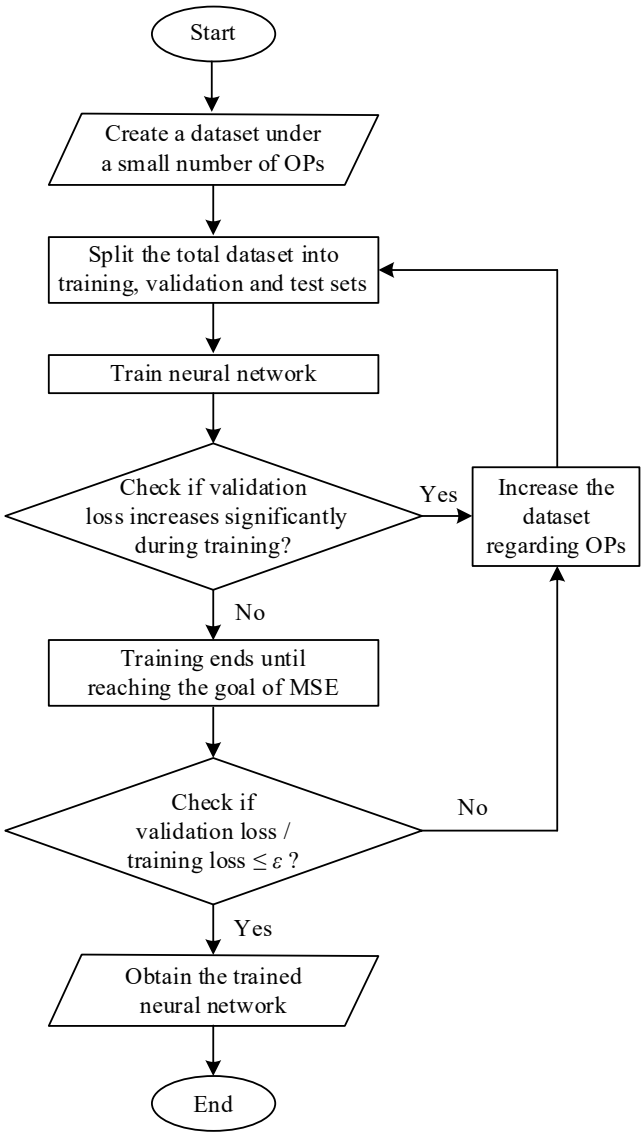


Fig. 3 Flowchart for learning from a small dataset of operating points.

B. Clustering for Improving Data Quality

Since the operating points are determined by multiple variables, how to establish a dataset with higher quality considering different samples of multiple variables is unclear. Thus, the clustering of impedance models is used to identify the sensitivity of impedance model to different OP variables.

Since the impedance model is a frequency-domain model represented by data points in a frequency series, which is analogous to a time series, the dynamic time wrapping approach [29] is used to estimate the distance between two frequency series for clustering.

Moreover, the impedance model is represented by multiple elements in a matrix, which include both real and imaginary parts. The data used for clustering can be reformulated by the frequency series in the following form:

$$Z_{\text{cluster}}(\omega)=\left[\cdots\ \mathrm{Re}\{Z_{ij}\}(\omega)\ \mathrm{Im}\{Z_{ij}\}(\omega)\ \cdots\right]\ \text{for}\ \forall i,j, \quad (5)$$

where $\mathrm{Re}\{Z\}$ denotes the real part and $\mathrm{Im}\{Z\}$ denotes the imaginary part. The real and imaginary data of each impedance

element need to be normalized before being put into $Z_{cluster}$, to ensure they are equally considered.

Then, clustered labels of the impedance data can be shown in the multi-dimensional OP space, to compare the sensitivity of impedance models to different OP variables. If the impedance data is clustered into more clusters in one dimension, it is indicated that the impedance is more sensitive to this variable. Finally, the data quality can be improved by selecting more samples in more sensitive variables and less samples in less sensitive variables.

V. Case Studies

The proposed FNN design has been tested on a grid-connected converter system. The impedance data can be obtained through virtual measurement in electronic transient (EMT) simulation software [16] or calculated analytically if the analytical model is available [7]. Since actual impedance measurement under a large number of operating points in an EMT simulation environment is very time consuming, and the measured models have been verified in good agreement with the analytical models even through experiments [8], this work forms the FNN training based on data obtained from analytical calculations.

The training is conducted in Matlab running on Intel(TM) i7-12700K CPU @ 4.6 GHz, with parallel computing enabled.

A. Description of Studied System and the Data Range

The studied converter system is shown in Fig. 4 with a single-line representation for the three-phase system. It adopts the typical grid-following control, with a phase-locked loop (PLL) for voltage synchronization and a current control loop (CCL) for current regulation [3]. The electrical system and control parameters are listed in Table I.

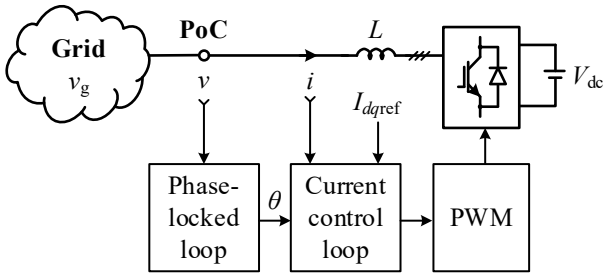


Fig. 4 Studied three-phase converter system with a single-line representation.

Table I: System circuit and control parameters

Parameter	Value	Parameter	Value
V_g	400V	V_{dc}	700V
f_0	50 Hz	f_s	10 kHz
L	5.1 mH	S_0	3 kVA
PLL bandwidth	200 Hz	CCL bandwidth	1 kHz

The frequency points are defined within [1, 5000] Hz, because the impedance model is only valid up to half of the switching frequency [9]. 100 frequency samples linearly distributed in the logarithmic coordinate are considered, because the impedance model is calculated based on the power of ω.

The operating points for the converter system are defined in the following ranges, according to the power system operation requirement [30].

$$\begin{cases} V \in [0.9,\ 1.1] \text{ p.u.} \\ P \in [-1,\ 1] \text{ p.u.} \\ Q \in [-1,\ 1] \text{ p.u.} \end{cases}, \quad \text{s.t. } 0 \text{ p.u.} < |I| < 1.1 \text{ p.u. and } |V_{\text{mod}}| < 0.5V_{\text{dc}} \tag{6}$$

where p.u. denotes *per unit* as determined by using the converter ratings, $|I|$ represents the converter current magnitude, $|V_{\text{mod}}|$ denotes the converter modulation voltage magnitude, and V_{dc} is the DC-link voltage of the converter. Due to physical limitations of the converter system, the operating current cannot exceed 1.1 p.u., and the modulation voltage cannot exceed $0.5V_{\text{dc}}$, i.e., the modulation index does not exceed one.

Since the grid-following converter is usually represented in admittance form, the following analysis uses admittance data for training. The *dq*-frame admittance has four elements, which can be trained using the same approach, thus, the following verification mainly takes one element as an example for illustration, i.e., $Y_{dq}(\omega, V, P, Q)$.

B. Verification of FNN Design

To verify the FNN design, a sufficiently large dataset is first considered. The OPs are sampled with stepsize 0.02 p.u. for V and stepsize 0.2 p.u. for both P and Q. In total, the dataset has 823 OPs. The total set is then randomly split, 70% of which is the training set, 15% of which is the validation set and the rest is the test set. Fig. 5 shows the distribution of OPs in the 3-dimension (3-D) space of V, P, and Q. The FNN design is verified in two aspects: 1) FNN structure and optimizer comparison; and 2) proposed design compared with trial-and-error design.

1) FNN structure and optimizer comparison

To verify the effectiveness of the FNN structure and the optimizer, it is first assumed that the analytical model is known, thus the latent features can be used to design N. In this case, the analytical model of $Y_{dq}(s, V, P, Q)$ indicates the highest pole/zero order of 12, if assuming a third-order Pade approximation for the time delay modeling, which is sufficiently adequate to model a converter's dynamics in the frequency range of interest [31]. Thus, according to (3), N can be selected as 26.

The training is conducted in Matlab. Several optimizers using the following algorithms are considered and compared [32].

- Variable Learning Rate Gradient Descent (*traingdx*)
- BFGS Quasi-Newton (*trainbfg*)
- Scaled Conjugate Gradient (*trainscg*)
- Levenberg-Marquardt (*trainlm*)

The Matlab train function names of these algorithms are denoted in the above brackets. These algorithms are compared by training the dataset in Fig. 5. The training goal is set to achieve the same MSE of 10^{-10}, which means that the absolute error of $\text{Re}\{Y_{dq}\}$ or $\text{Im}\{Y_{dq}\}$ is less than 10^{-5} (−100 dB).

Fig. 6 shows the training performances with epoch and time using the above algorithms, where the right figures are the zoomed-in plots of the left figures. The epoch denotes the number of iterations that indicates the convergence rate. Time

reflects the efficiency of training. For easier comparison, the training performance is only displayed within 10000 epochs or 150 minutes. The *gdx* algorithm (blue line) converges slowly since it is based on the gradient descent. Even with variable learning rate and momentum, it can hardly scape from local minimum to achieve the training goal. The *bfg* algorithm (red line) considers the Newton's method, which thus converges much faster. However, it may also confront the local-minimum problem when the error becomes small. The *scg* algorithm (yellow line) converges slightly slower than the *bfg* algorithm, yet it reduces training time a lot, because it avoids the time-consuming line search used in the *bfg* algorithm by scaling the step size with a Levenberg-Marquardt based algorithm [33]. It reaches the goal after around 144 minutes. However, the *scg* algorithm is still based on the gradient, which also escapes from local minima slowly, thus the training becomes more time-consuming when the error becomes smaller. The *lm* algorithm (purple line) is the fastest, which achieves the goal only at Epoch 287 within 6 minutes. Compared with the *scg* algorithm, the *lm* algorithm can escape from local minima more easily, as shown by its stair-like performance curve. The comparison results verify the optimizer selection discussed in Section III-C.

The training, validation and test performance with the *lm* algorithm is shown in Fig. 7. It can be seen that the validation performance is very close to the training performance along epochs, indicating that the training is effective. The test performance also shows a good prediction. It is also noted that the validation check is considered during the training process, in order to identify the best fitting point in case it is present before the training goal is achieved. However, it can be seen that the performance curves are converging until the MSE goal is achieved, indicating that the designed multilayer FNN structure works successfully in learning this problem.

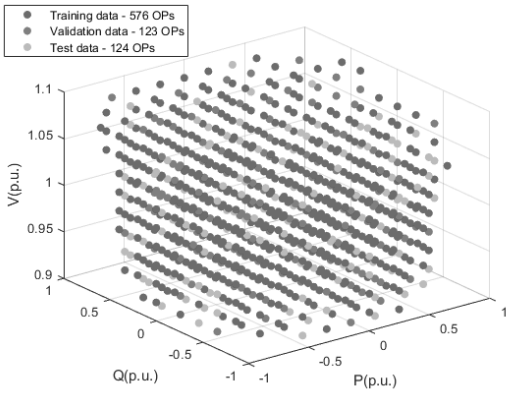


Fig. 5 Operating point distribution based on a random split of the dataset: training data – 70%, validation data – 15%, test data – 15%.

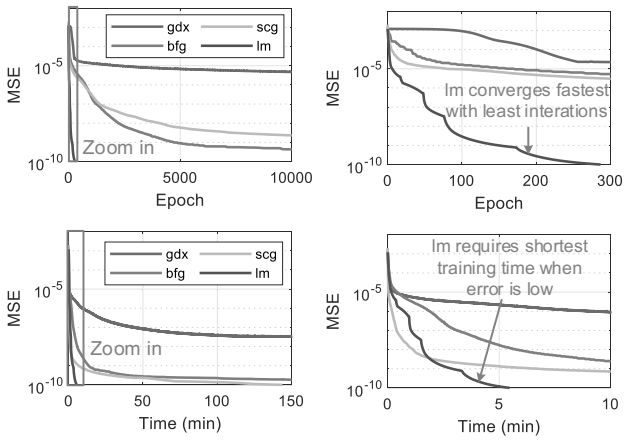


Fig. 6 Comparison of different optimization algorithms for training Y_{dq}.

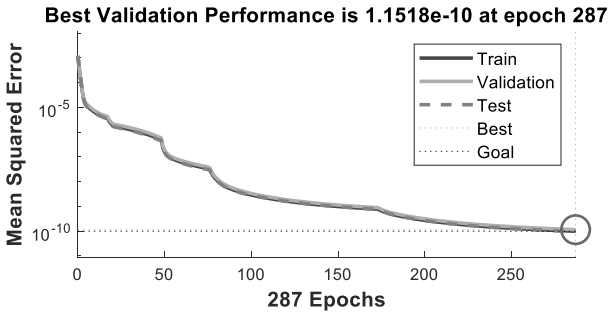


Fig. 7 Training, validation and test performance with the *lm* algorithm for training Y_{dq}.

2) Proposed design compared with trial-and-error design

Although the effectiveness of the proposed FNN design has been preliminary verified, it is still unclear if this approach really achieves a good design. Therefore, the proposed design is also compared with the trial-and-error design. In practice, the latent features of pole/zero numbers can be unknown, thus a trial-and-error approach by searching N in a certain range can also give an optimal design of FNN.

To indicate a reasonable range of N for the trial-and-error design, the pole/zero orders of analytical models are listed in the left column of Table II, considering different orders of Pade approximation ranging from 0 to 3. The designed N based on these latent features are called as N_{theory} in the table. In contrast, the trial-and-error design searches the FNN's N around the possible range of N_{theory}, i.e. [14, 28]. The training stops as MSE reaches 10^{-10}, and the training results including end epoch and end time are listed in the right column of Table II.

To clearly show the optimal FNN design based on the trial-and-error method, the end epoch and end time against the FNN's N changing is plotted in Fig. 8. A smaller end epoch indicates that the FNN structure works better in terms of convergence. It can be seen that when N is larger than 22, the end epoch becomes closer, but the end time becomes longer because the increased complexity of FNN also increases the computational burden. Thus, considering the tradeoff between convergence and training efficiency, an optimal value can be found around N=22 in this case. Existing analytical model-based studies also show that usually a third-order Pade approximation is adequate for modeling time delay [31].

Although the optimal N is found slightly lower than the N_{theory} indicated by the analytical model based on a third-order Pade approximation, this is reasonable, because when we design N_{theory} using (3), we use more latent features, which may lead to more conservative design and end up with a more complex FNN. But the resulting optimal N is still very close to the suggested order based on the analytical model with a third-order Pade approximation, which can verify that the proposed FNN design based on latent features almost achieves an optimal design.

Table II Training results with different N

Analytical model			Training results		
Pade approx. order	Highest pole/zero order	N_{theory}	FNN N	End epoch	End time (min)
0	6	14	14*14	814	1:31
-	-	-	16*16	474	1:15
1	8	18	18*18	958	3:31
-	-	-	20*20	606	3:02
2	10	22	22*22	256	1:57
-	-	-	24*24	363	4:27
3	12	26	26*26	287	5:29
-	-	-	28*28	298	7:26

It is worth noting that although the exact pole/zero order of the transfer function may be unknown in practice, a reasonable range of the zero/pole order can also be estimated. Then, an optimal N for the FNN design can still be searched for within this range.

C. Verification of Learning Flowchart from Small Dataset

To verify the learning flowchart in Fig. 3, three datasets are used to train Y_{dq}, by varying number of samples in operating points, as shown in Fig. 9. These datasets are listed as follows:

- Dataset A considers a stepsize of 0.05 p.u. in V and a stepsize of 0.5 p.u. in P/Q, resulting in 49 OPs.
- Dataset B considers a stepsize of 0.04 p.u. in V and a stepsize of 0.4 p.u. in P/Q, resulting in 106 OPs.
- Dataset C considers a stepsize of 0.03 p.u. in V and a stepsize of 0.3 p.u. in P/Q, resulting in 244 OPs.

The performance of Y_{dq} with the three datasets is displayed in Fig. 10. From the very large ratio of validation loss to training loss, it can be seen that Dataset A is not sufficient to train a good model. If the data is increased to 106 OPs, the validation loss becomes much closer to training loss, resulting in a ratio around 1.26 at the end of training. Even though the test loss is a bit higher, these loss performance curves show a similar trend of convergence, indicating that the training is effective. The slight error difference in the test performance can be easily compensated by setting a smaller training goal in MSE. When the dataset is further increased to 244 OPs, the validation loss becomes smaller than the training loss with a ratio of 0.88, which implies that the data is more sufficient for training. Therefore, the ratio of validation loss over training loss can be used as an indicator for dataset selection. In this case, if considering a threshold of $\varepsilon = 1.5$ for the ratio of validation loss over training loss, it can be found that using a dataset around 100 OPs is already acceptable for training Y_{dq}.

D. Verification of Clustering for Improving Data Quality

The dataset increasement in Fig. 9 considers the sample increasement in each dimension equally. However, this may not lead to the best data quality. How clustering can help improve the data quality is verified in this part.

It is worth noting that to conduct an effective clustering, a sufficient amount of data is still needed. Thus, such clustering analysis can be done when data can be more easily acquired from analytical models or simulation models. Then, the sensitivity conclusion drawn from the clustering can still be helpful for guiding data collection from actual measurements.

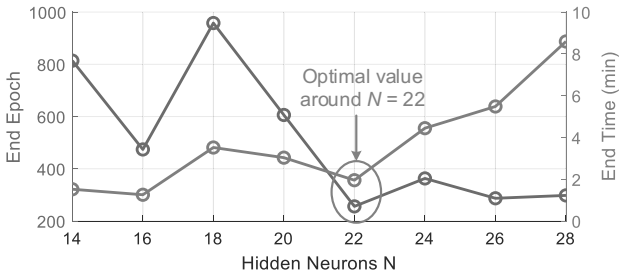
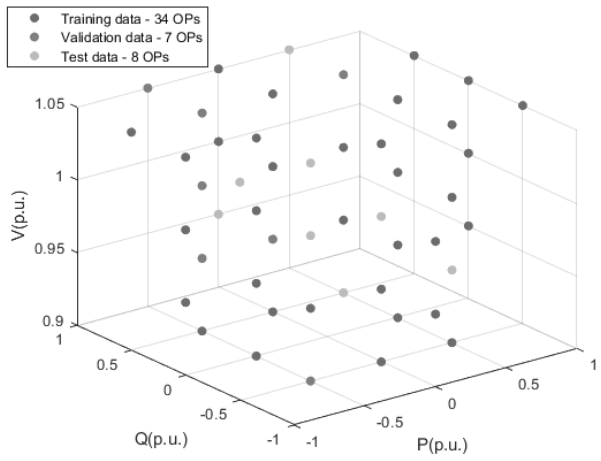
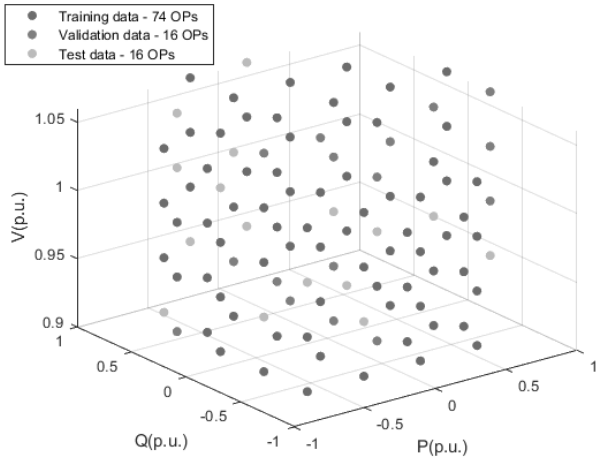


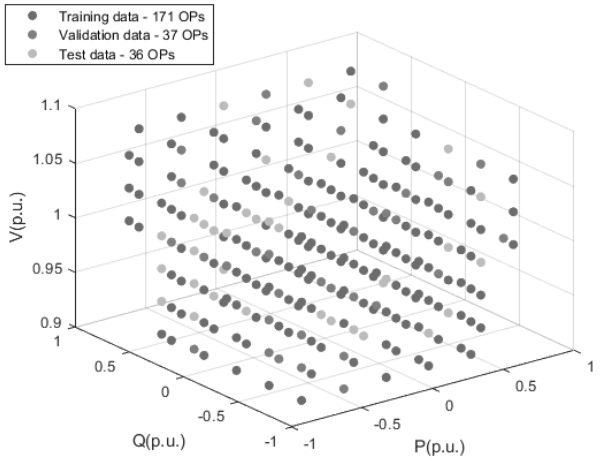
Fig. 8 Training results of Y_{dq} based on trial-and-error design of FNN considering different N.



(a)



(b)



(c)

Fig. 9 Datasets considering different samples of operating points. (a) Dataset A with 49 OPs; (b) Dataset B with 106 OPs; (c) Dataset C with 244 OPs.

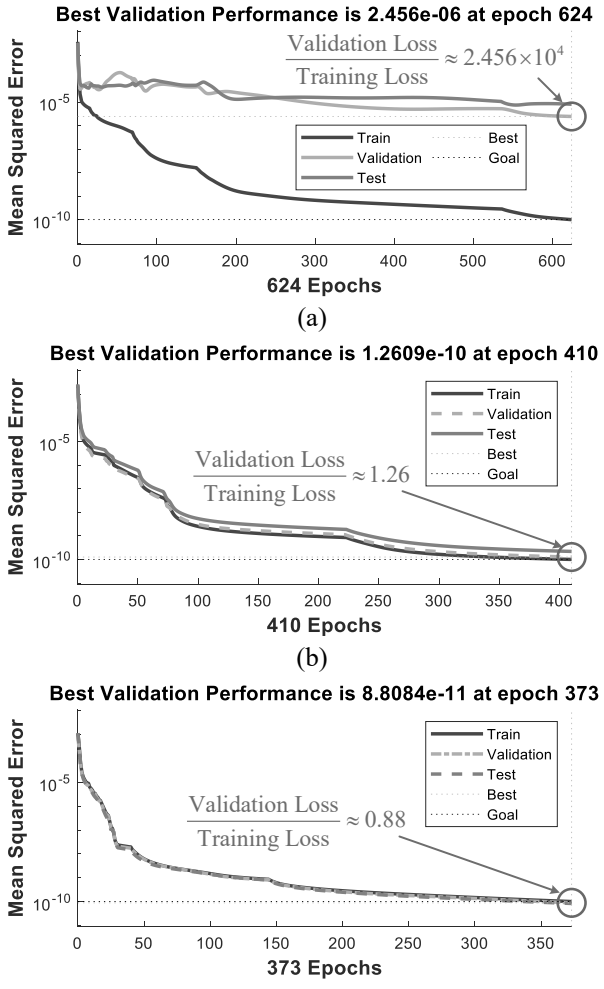
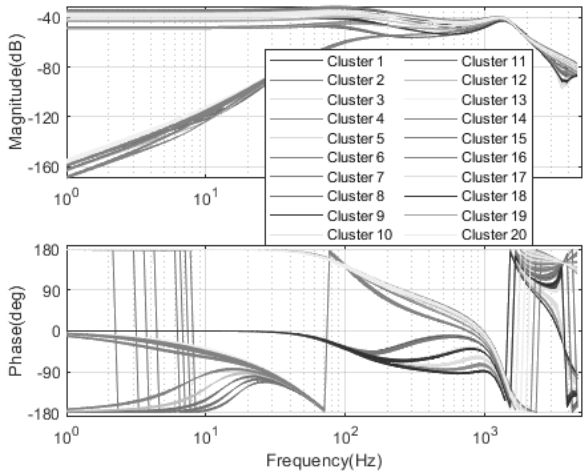


Fig. 10 Training, validation and test performance for training Y_{dq} with different datasets. (a) Dataset A with 49 OPs; (b) Dataset B with 106 OPs; (c) Dataset C with 244 OPs.

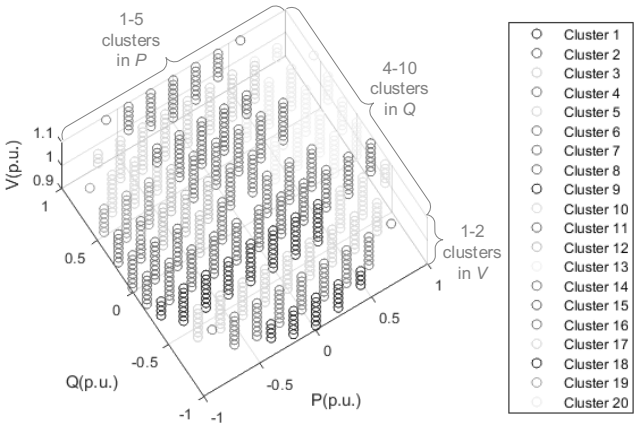
The clustering is applied based the dataset of 823 OPs shown in Fig. 5. Since the OPs can influence all the elements in the admittance matrix, the clustering is implemented based on Y_{dd}, Y_{dq}, Y_{qd}, and Y_{qq}. The number of clusters is taken to be 20 for clustering, which is sufficiently large to compare the multivariable sensitivity relationship. Fig. 11 shows the clustered Bode diagrams using Y_{dq} as an example and the clusters in the 3-D OP space. Each cluster is denoted by one color. It can be seen from Fig. 11(b) that the admittance data is clustered into more clusters in the dimension of Q and fewer clusters in the dimension of V, indicating the variable sensitivity ranking from high to low as: $Q > P > V$.

Thereafter, the dataset is increased based on Dataset A, by selecting different samples in V, P, and Q. Four new datasets that have close numbers of OPs are compared, which are listed as follows:

- Dataset D only increases the stepsize in V to 0.03 p.u. and keeps the same stepsizes in P and Q as Dataset A, resulting in 89 OPs.
- Dataset E only increases the stepsize in P to 0.3 p.u. and keeps the same stepsizes in V and Q as Dataset A, resulting in 91 OPs.
- Dataset F only increases the stepsize in Q to 0.3 p.u. and keeps the same stepsizes in V and P as Dataset A, resulting in 92 OPs.
- Dataset G considers a stepsize of 0.07 p.u. in V, a stepsize of 0.4 p.u. in P and a stepsize of 0.3 p.u. in Q, resulting in 86 OPs. This data selection is based on the variable sensitivity ranking from clustering analysis.



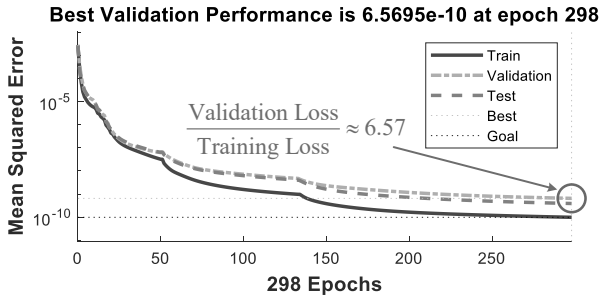
(a)



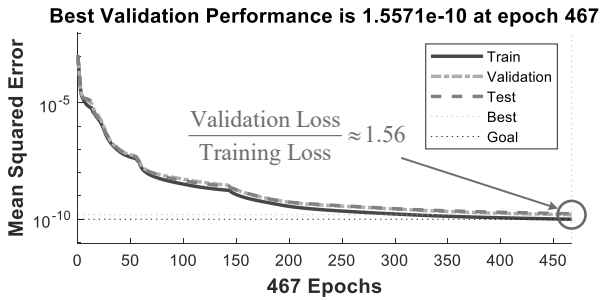
(b)

Fig. 11 Admittance clustering results. (a) Clustered Bode diagrams of Y_{dq}; (b) Clusters in the 3-D OP space.

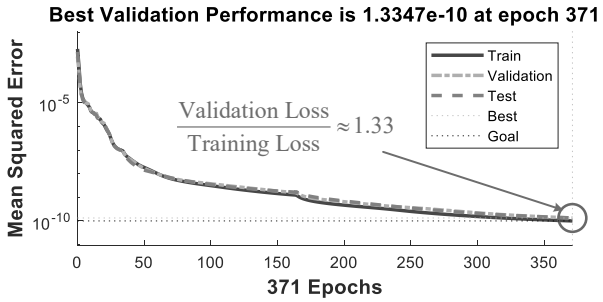
The performance results of training Y_{dq} with Datasets D-G are compared in Fig. 12. It can be seen that by increasing the OPs, the ratios of validation loss to training loss have all been reduced compared with Fig. 10(a). Increasing Q samples achieves better training performance than increasing V or P samples, because it reduces the ratio of validation loss over training loss more. This verifies the variable sensitivity ranking result concluded from clustering. Furthermore, when Dataset G takes the variable sensitivity ranking into account to select the OP samples, the data has the highest quality, which results in the lowest ratio of validation loss over training loss as 0.824. It is thus verified that the clustering can facilitate data selection with higher quality.



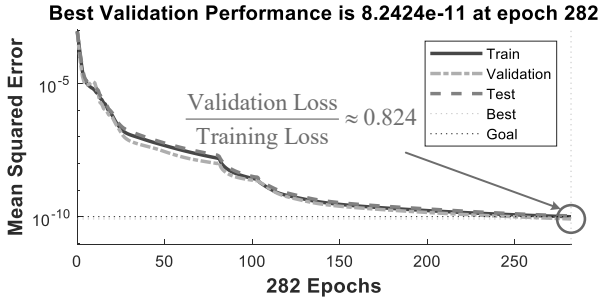
(a)



(b)



(c)



(d)

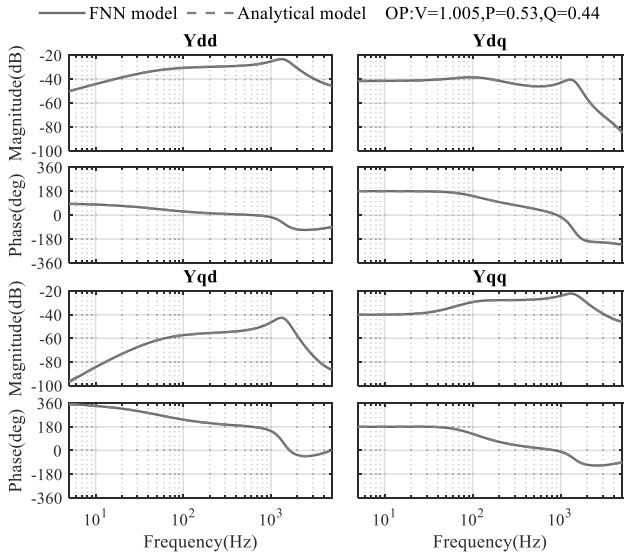
Fig. 12 Training, validation and test performance for training Y_{dq} with different datasets with similar sizes. (a) Dataset D by increasing V samples; (b) Dataset E by increasing P samples; (c) Dataset F by increasing Q samples; (d) Dataset G by selecting samples in V, P, Q based on variable sensitivity ranking.

E. Summary and Discussion

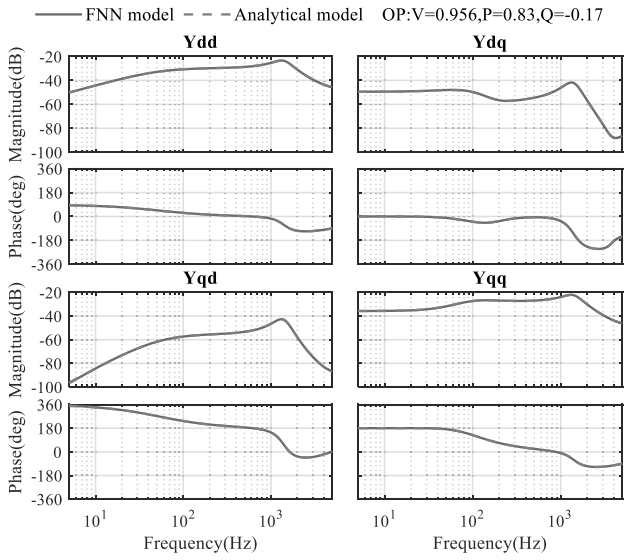
According to the above studies, the suggested procedures for FNN design of converter admittance model training are listed as follows:

- The optimal number of hidden neurons of the FNN can be found first based on searching over N within a reasonable range, which is found as N=22 for this studied case.
- The sufficient number of OPs for effective training can be found based on the flowchart of Fig. 3, which is around 100 OPs in this case.
- The clustering may further help improve the data quality by selecting the number of variable samples based on the variable sensitivity ranking, which is Q>P>V for this case.

Considering the above findings, all the admittance elements of the converter are finally trained based on a dataset of 116 OPs, with a stepsize of 0.05 p.u. in V, a stepsize of 0.4 p.u. in P and a stepsize of 0.3 p.u. in Q. The training goal of each admittance element is MSE<10^{-12}. The trained FNN model is compared with the analytical model in Fig. 13 using two randomly selected OPs, which are not included in the dataset. The good agreement shows that the trained FNN model can well predict the features of admittance models under a wide range of OPs through learning from a limited dataset.



(a)



(b)

Fig. 13 Verification of FNN models for all the admittance elements on Bode diagrams. (a) First randomly selected OP; (b) Second randomly selected OP.

Through the case study, it has been shown that to train an accurate impedance model for a converter system does not requires huge amount of data (OPs to hundred level can be sufficient). This makes it feasible for power system operators to apply this approach for model identification of black-box converter systems.

It is worth mentioning that the trained FNN impedance model is only valid for the certain converter with fixed control structure and parameters. Although in reality, the power system may be integrated by a large number of power converters, usually for a single application, e.g., in a wind or solar power plant, the converters are manufactured by the same vendor, thus, the converter control structure and parameters are the same. It enables to aggregate a large power plant using one equivalent power converter at its point of connection to the power system. This is also how transmission system operators require converter models from vendors for the system-level dynamic studies [34]. Therefore, the developed FNN impedance model for a single converter system is readily applicable for transmission-level converter system modeling and dynamic analysis.

VI. Conclusion

This paper has introduced a deep learning approach based on multilayer FNNs for the frequency-domain modeling of power electronic systems. In addition to basic implementations, it has addressed several important concerns for practical applications, including

- How to design and optimize the structure of FNN based on the latent features of the system;
- How to select a small yet high-quality dataset that achieves good learning performance.

Although this work has primarily developed the modeling approach at the converter level, the method establishes a knowledge base for applying deep learning in frequency-domain dynamic studies of converter-based power systems. Case studies have confirmed the effectiveness of proposed methods and design procedures for practical scenarios.

References

[1] F. Blaabjerg, Z. Chen and S. B. Kjaer, "Power electronics as efficient interface in dispersed power generation systems," in IEEE Transactions on Power Electronics, vol. 19, no. 5, pp. 1184-1194, Sept. 2004, doi: 10.1109/TPEL.2004.833453.

[2] D. Boroyevich, I. Cvetkovic, R. Burgos and D. Dong, "Intergrid: A Future Electronic Energy Network?," in IEEE Journal of Emerging and Selected Topics in Power Electronics, vol. 1, no. 3, pp. 127-138, Sept. 2013, doi: 10.1109/JESTPE.2013.2276937.

[3] X. Wang and F. Blaabjerg, "Harmonic Stability in Power Electronic-Based Power Systems: Concept, Modeling, and Analysis," in IEEE Transactions on Smart Grid, vol. 10, no. 3, pp. 2858-2870, May 2019, doi: 10.1109/TSG.2018.2812712.

[4] W. Ren and E. Larsen, "A Refined Frequency Scan Approach to Sub-Synchronous Control Interaction (SSCI) Study of Wind Farms," in IEEE Transactions on Power Systems, vol. 31, no. 5, pp. 3904-3912, Sept. 2016, doi: 10.1109/TPWRS.2015.2501543.

[5] N. A. Cifuentes Otto, M. Sun, R. Gupta and B. C. Pal, "Black-Box Impedance-Based Stability Assessment of Dynamic Interactions Between Converters and Grid," in IEEE Transactions on Power Systems, doi: 10.1109/TPWRS.2021.3128812.

[6] Y. Liao, X. Wang and X. Wang, "Frequency-Domain Participation Analysis for Electronic Power Systems," in IEEE Transactions on Power Electronics, vol. 37, no. 3, pp. 2531-2537, March 2022, doi: 10.1109/TPEL.2021.3118439.

[7] L. Harnefors, M. Bongiorno and S. Lundberg, "Input-Admittance Calculation and Shaping for Controlled Voltage-Source Converters," in IEEE Transactions on Industrial Electronics, vol. 54, no. 6, pp. 3323-3334, Dec. 2007, doi: 10.1109/TIE.2007.904022.

[8] Z. Shen, "Online measurement of three-phase AC power system impedance in synchronous coordinates," Ph.D. dissertation, Dept. Elect. Eng., Virginia Polytech. Inst. State Univ., Blacksburg, VA, USA, 2012.

[9] Y. Liao and X. Wang, "Small-Signal Modeling of AC Power Electronic Systems: Critical Review and Unified Modeling," in IEEE Open Journal of Power Electronics, vol. 2, pp. 424-439, 2021, doi: 10.1109/OJPEL.2021.3104522.

[10] W. Liu, X. Xie, J. Shair and X. Li, "A Nearly Decoupled Admittance Model for Grid-Tied VSCs Under Variable Operating Conditions," in IEEE Transactions on Power Electronics, vol. 35, no. 9, pp. 9380-9389, Sept. 2020.

[11] R. Pintelon, P. Guillaume, Y. Rolain, J. Schoukens and H. V. Hamme, "Parametric identification of transfer functions in the frequency domain—

A survey", IEEE Trans. Automat. Control, vol. 39, no. 11, pp. 2245-2260, Nov. 1994.
[12] A. N. Kolmogorov, “On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition,” Dokl. Akad. Nauk SSSR, vol. 144, no. 5, pp. 953-956, Apr. 1957.
[13] P. Xiao, G. K. Venayagamoorthy, K. A. Corzine and J. Huang, "Recurrent neural networks based impedance measurement technique for power electronic systems", IEEE Trans. Power Electron., vol. 25, no. 2, pp. 382-390, Feb. 2010.
[14] Wunderlich and E. Santi, "Digital Twin Models of Power Electronic Converters Using Dynamic Neural Networks," 2021 IEEE Applied Power Electronics Conference and Exposition (APEC), 2021, pp. 2369-2376.
[15] Elias Kaufhold, Simon Grandl, Jan Meyer, Peter Schegner, "Feasibility of Black-Box Time Domain Modeling of Single-Phase Photovoltaic Inverters Using Artificial Neural Networks", Energies, vol. 14, pp. 2118, 2021.
[16] M. Zhang, X. Wang, D. Yang and M. G. Christensen, "Artificial Neural Network Based Identification of Multi-Operating-Point Impedance Model," in IEEE Transactions on Power Electronics, vol. 36, no. 2, pp. 1231-1235, Feb. 2021.
[17] H. Wu, X. Wang, Y. Liao, M. Ndreko, R. Dimitrovski, and W. Winter, “Development of the toolbox for AC/DC impedance matrix measurement of MTDC system,” in Proc. 20th Wind Integration Workshop, Sept. 2021, Berlin, Germany.
[18] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, Feedback Control of Dynamic Systems. London, U.K.: Pearson Educ., 2015.
[19] Y. Liao, H. Wu, X. Wang, M. Ndreko, R. Dimitrovski, and W. Winter, “Stability and Sensitivity Analysis of Multi-Vendor, Multi-Terminal HVDC Systems,” arXiv, Nov. 2021.
[20] D. Stathakis, “How many hidden layers and nodes?” International Journal of Remote Sensing, vol. 30, no. 8, pp. 2133-2147, Apr. 2009.
[21] C. Alippi, "Selecting accurate, robust, and minimal feedforward neural networks," in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 49, no. 12, pp. 1799-1810, Dec. 2002, doi: 10.1109/TCSI.2002.805710.
[22] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” Journal of Computational Physics, vol. 378, pp. 686-707, Feb. 2019.
[23] F. Djeumou, C. Neary, E. Goubault, S. Putot, U. Topcu, “Neural Networks with Physics-Informed Architectures and Constraints for Dynamical Systems Modeling,” arXiv, Sep. 2021.
[24] H. W. Lin, M. Tegmark, and D. Rolnick, “Why Does Deep and Cheap Learning Work So Well?” Journal of Statistical Physics, vol. 168, pp. 1223-1247, 2017.
[25] L. Carin, “Interpretation of Multilayer Perceptron,” Introduction to Machine Learning, Duke University [Online]. Available: https://www.coursera.org/lecture/machine-learning-duke/interpretation-of-multilayer-perceptron-hJIuy
[26] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," in IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, Nov. 1994, doi: 10.1109/72.329697.
[27] H. P. Gavin, “The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems”, Duke University, Sep. 2020 [Online]. Available: https://people.duke.edu/~hpgavin/ce281/lm.pdf
[28] J. Chen, Z. Liu, H. Wang, A. Núñez and Z. Han, "Automatic Defect Detection of Fasteners on the Catenary Support Device Using Deep Convolutional Neural Network," in IEEE Transactions on Instrumentation and Measurement, vol. 67, no. 2, pp. 257-269, Feb. 2018, doi: 10.1109/TIM.2017.2775345.
[29] D. J. Berndt and J. Clifford, “Using dynamic time wrapping to find patterns in time series,” in AAAI Workshop on Knowledge Discovery in Databases, pp. 359-370, 1994.
[30] P. Kundur, Power system stability and control. McGraw-hill New York, 1994.
[31] Y. Liao and X. Wang, "Impedance-Based Stability Analysis for Interconnected Converter Systems With Open-Loop RHP Poles," in IEEE Transactions on Power Electronics, vol. 35, no. 4, pp. 4388-4397, April 2020, doi: 10.1109/TPEL.2019.2939636.
[32] MathWorks. *Choose a Multilayer Neural Network Training Function*. [Online]. Available: https://www.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html
[33] M. F. Møller, “A scaled conjugate gradient algorithm for fast supervised learning,” Neural Networks, vol. 6, no. 4, pp. 525-533, 1993.
[34] Energinet, “Requirements for generators (RfG) – simulation model requirmeents,” Nov. 2018.