*energies*

MDPI

*Article*

# Convolutional Neural Network for the Classification of the Control Mode of Grid-Connected Power Converters

**Rabah Ouali [1,*], Martin Legry [2], Jean-Yves Dieulot [1], Pascal Yim [1], Xavier Guillaud [2] and Frédéric Colas [2]**

[1] Centre de Recherche en Informatique Signal et Automatique de Lille, University Lille, CNRS, Centrale Lille, UMR 9189, 59655 Lille, France; jean-yves.dieulot@univ-lille.fr (J.-Y.D.); pascal.yim@centralelille.fr (P.Y.)
[2] L2EP—Laboratoire d'Electrotechnique et d'Electronique de Puissance, University Lille, Centrale Lille, Arts et Métiers Paris Tech, HEI, EA 2697, 59655 Lille, France; martin.legry@ensam.eu (M.L.); xavier.guillaud@centralelille.fr (X.G.); frederic.colas@ensam.eu (F.C.)
* Correspondence: rabah.ouali@centralelille.fr

**Abstract:** With the integration of power converters into the power grid, it becomes crucial for the Transmission System Operator (TSO) to ascertain whether they are operating in Grid Forming or Grid Following modes. Due to intellectual properties, classification can only be performed based on non-intrusive measurements and models, such as admittance at the PCC. This classification poses a challenge as the TSO lacks precise knowledge of the actual control structures and algorithms. This paper introduces a novel classification algorithm based on Convolutional Neural Networks (CNN), capable of detecting patterns in sequential data. The proposed CNN utilizes a new architecture to separate admittances along the d and q axes, and a decision layer allows to determine the correct converter mode. The performance of the proposed CNN model was assessed through two tests and compared to the traditional feedforward model. The proposed CNN architecture demonstrates significant classification capabilities, as it is able to identify the control mode of the converter even when its control structure is not part of the training dataset.

**Keywords:** grid forming; grid following; deep learning; convolutional neural network; frequency admittance



Academic Editors: Donghe Li and Yu Xiao

Received: 28 October 2024
Revised: 2 December 2024
Accepted: 20 December 2024
Published: 22 December 2024

**Citation:** Ouali, R.; Legry, M.; Dieulot, J.-Y.; Yim, P.; Guillaud, X.; Colas, F. Convolutional Neural Network for the Classification of the Control Mode of Grid-Connected Power Converters. *Energies* **2024**, *17*, 6458. https://doi.org/10.3390/en17246458



## 1. Introduction

In recent years, conventional energy production based on synchronous machines is being replaced by renewable energy sources, connected to the electrical grid through power converters. Two converter control modes, called Grid Following (GFL) and Grid Forming (GFM) [1,2], have been developed. GFL, which is the most widespread, controls the converter as a current source, using a current loop to control the power at the output of the converter while synchronizing with the grid using a Phase-Locked Loop (PLL) [3]. GFM controls the converter as a voltage source to emulate the behavior of a synchronous machine, acting as a Virtual Synchronous Machine [4]. The choice of a control mode by Transmission System Operators (TSOs) is an important step to ensure the proper functionality and stability of the grid [5,6]. Moreover, the European Network of TSOs for Electricity (ENTSO-E) has published a report detailing various requirements for control modes [7], and a guide of best practices to specify dynamics has been proposed in the United Kingdom [8].

In their process of ensuring stable operation and avoiding stability issues, TSOs conduct studies and simulations during the planning phase for various scenarios to assess the grid ability to integrate converters (N-1, N-2 criteria, Low Voltage Ride Through, etc.). The results of these scenarios are translated into contractual requirements, and therefore,

manufacturers provide black-box simulation models to assess a correct operation of their converters. However, TSOs have no access to the real control, which is protected by intellectual properties, and have no mean to verify that the specified control mode of a converter has actually been implemented by the manufacturer. The main objective of this paper is to provide a verification tool for TSOs that allow them to check whether a converter is operating in GFM or GFL mode according to the contractual specifications. This verification should use the measurements at the PCC that are the only ones accessible to the TSOs.

The conventional black box models used for power converters are small-signal models known as admittance [9]. One of the advantages to use admittance instead of time-domain model is that it is easy to implement experimentally and can be found from the measurements at the PCC [10]. The admittance is a behavioral modeling which strongly depends on the control, which are unknown to the TSO, and on the value of their parameters, which can vary in a large range. Hence, the challenge of verifying whether a converter operates in GFM or GFL mode only based on admittance patterns is a real challenge that has never been addressed in the literature.

In a previous study [11], the authors have simulated a large database of GFM and GFL converters, based on a number of conventional control structures of the literature. Since the set of admittance with GFL and GFM present some overlap, they explored whether the classification of control modes using admittances of power converters could be solved with traditional machine learning algorithms (Decision Tree, Random Forests, Gradient Boosting, and Support Vector Machines). Table 1 presents the results in terms of accuracy for two tests: classification and generalization. These algorithms demonstrated an excellent ability to classify data when the control structures originated from a training set based on simulations of conventional structures. However, in a real-world context, manufacturers may design variants of these control structures and the ability of these algorithms was evaluated through a generalization test and demonstrated poor performances. To alleviate this issue with generalization, neural networks have been proposed in the literature for different classification problems. In [12], visible spectrum images and classification algorithms were used to detect the soiling of photovoltaic panels. The convolutional neural network (CNN) provided the best results compared to other methods. In [13], the CNN effectively captures the spatiotemporal coupling of heterogeneous loads, thereby improving forecasting accuracy.

The main contribution of this paper is the proposal of a powerful new algorithm designed to provide a verification tool for the control mode of a grid-connected converter. This classification algorithm is based on an innovative CNN architecture that leverages the specific characteristics of the converter admittances. The superior performance of the algorithm, compared to traditional neural networks and classical machine learning methods, will be demonstrated through various classification tests and generalization assessments.

**Table 1.** Machine learning algorithms for power converter classification [11].

| Machine Learning Algorithm | Classification | Generalization |
|---|---|---|
| Linear Regression (LR) | 94% | 80% |
| Random Forest (RF) | 98% | 91% |
| Decision Tree (DT) | 92% | 40% |
| Naive Bayes Classifier (NBC) | 92% | 80% |
| XGBoost (XGB) | 95% | 79% |
| Support Vector Machine (SVM) | 95% | 84% |
| K-Nearest Neighbors (KNN) | 94% | 84% |

This article is organized as follows. Section 2 introduces the different control modes and their structures. Admittances are presented and the challenge in classifying the converter is highlighted in Section 3. Section 4 presents how the datasets were generated and the classification of control modes using a combination of Convolutional Neural Networks. Special attention will be paid on the generalization of the algorithms and conclusion will be drawn in Section 5.

## 2. Power Converter Control Modes and Structures

The following subsections present the GFL and GFM control modes, based on Proportional–Integral (PI) regulators, along with the corresponding control structures for each mode. These structures, widely used in HVDC converter systems, Type-4 wind turbines, and photovoltaic systems, typically neglect DC-side dynamics by considering the DC bus as an ideal source, focusing the control exclusively on the AC side.

### *2.1. Structures of GFL Mode*

In GFL mode, the converter is controlled as a current source. GFL operates within an existing grid that sets the voltage and frequency. A PLL synchronizes the local d-q frame of the converter with the grid voltage at the PCC, with $\tilde{\theta}_g$ being an estimation of the grid phase angle $\theta_g^*$. The estimated angle is used in the Park transformation and its inverse, denoted as $P(\tilde{\theta}_g)$ and $P(\tilde{\theta}_g)^{-1}$ [14]. The output current of the converter is controlled with current control PI controllers. The reference currents $i_d^*$ and $i_q^*$ for this loop are generated from different structures: active and reactive power control GFL (pqGFL), active power and voltage control GFL (pvGFL), and direct current control GFL (viGFL). For pqGFL and pvGFL structures, the direct reference current $i_d^*$ is generated from a PI loop of active power control. However, the quadrature reference current $i_q^*$ is generated using two different loops [14]. For pqGFL, the current $i_q^*$ is generated from a PI loop of reactive power control, and in pvGFL, it is generated from a PI control loop of voltage at PCC. The control structures for pqGFL and pvGFL are presented in Figure 1.

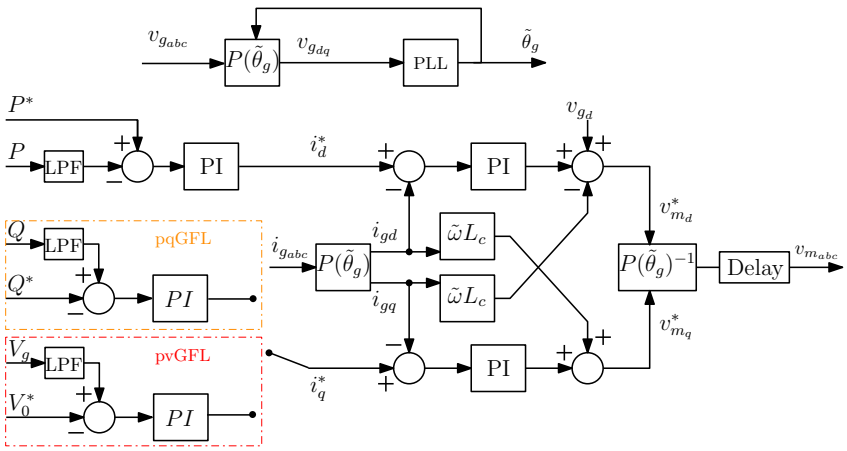



**Figure 1.** pqGFL and pvGFL control structures.

For viGFL, the direct reference current $i_d^*$ is generated through a simple division, where the active power reference $P^*$ is divided by the direct component of the voltage measured at the PCC and filtered through a Low-Pass Filter (LPF). The quadrature reference current $i_q^*$ is also generated by dividing the reference reactive power $Q^*$ by the same filtered voltage and multiplying it by a negative unity gain. Figure 2 presents the control structure of viGFL [14].

The parameters and bandwidths of the different control loops in the GFL structures are presented as follows: $\omega_p$, $\omega_q$, $\omega_v$, and $\omega_{LPF}$, respectively, represent the bandwidths of

the active power, reactive power, voltage, and low-pass filter control loops, $\omega_{pll}$ and $\omega_{cc}$ denote the bandwidths of the PLL and the current control loop, respectively, and finally, *Delay* represents the modulation-related delay.

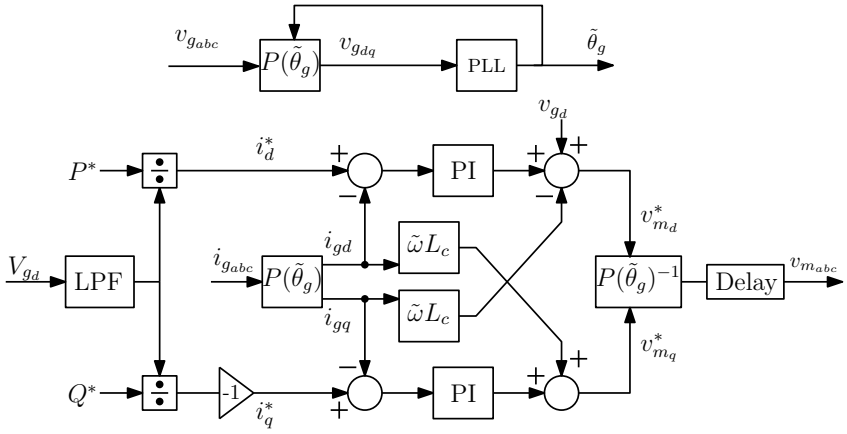



**Figure 2.** The viGFLcontrol structure.

### *2.2. Structures of GFM Mode*

In GFM mode, the converter is controlled as a voltage source. This mode does not require an existing grid for synchronization, as it enables simultaneous control of both frequency and voltage at the output, allowing autonomous operation. Several control structures have been proposed for this mode. The common feature of all GFM approaches is that they define references for the angle and amplitude of the voltage modulated by the converter. The angle reference $\theta_m^*$ can be obtained using various methods, including droop control, power synchronization control, synchro-converters, Virtual Synchronous Machine (VSM), or PI control [15–17]. It has been demonstrated that, when properly tuned, these schemes lead to equivalent dynamic behavior [18]. Among these, the VSM and PI schemes are the most commonly adopted in the literature for transmission system applications, as they offer a more intuitive transition from traditional synchronous generation [19]. To control the voltage amplitude of the converter, several structures are also proposed, such as voltage-controlled GFM (vcGFM), current-controlled GFM (ccGFM), and PI-based GFM (piGFM).

The vcGFM control structure, illustrated in Figure 3, uses the VSM scheme proposed in [15] to generate the angle reference $\theta_m^*$ for active power control. The gains of this loop are designed to provide inertia and damping effects, represented by $H$ and $\xi$, respectively. Additionally, current dynamics are actively damped by adjusting the voltage references $v_{md,q}^*$ using a Transient Virtual Resistor (TVR) [19,20]. The modulated voltage is controlled by directly setting the references $v_{md,q}^*$ to $(V^*, 0)$, where $V^*$ represents the voltage amplitude reference. These modulated voltages are then rotated according to the generated angle reference $\theta_m^*$.

The ccGFM control structure, also shown in Figure 3, employs the same VSM scheme to generate $\theta_m^*$. Similar to vcGFM, voltage references are set for the modulated voltage. However, a current control loop is used to generate the modulated voltage references, as described in [20]. The Quasi-Static Electrical Model (QSEM) of the circuit between the modulated voltage and the Point of Common Coupling (PCC) voltage is utilized to determine the reference values for the currents $i_{dq}^*$ [20].

A third structure, piGFM, is also proposed. In this structure, the voltage amplitude is controlled using the same loop as in vcGFM, while the angle $\theta_m$ is controlled via a dedicated PI control loop.

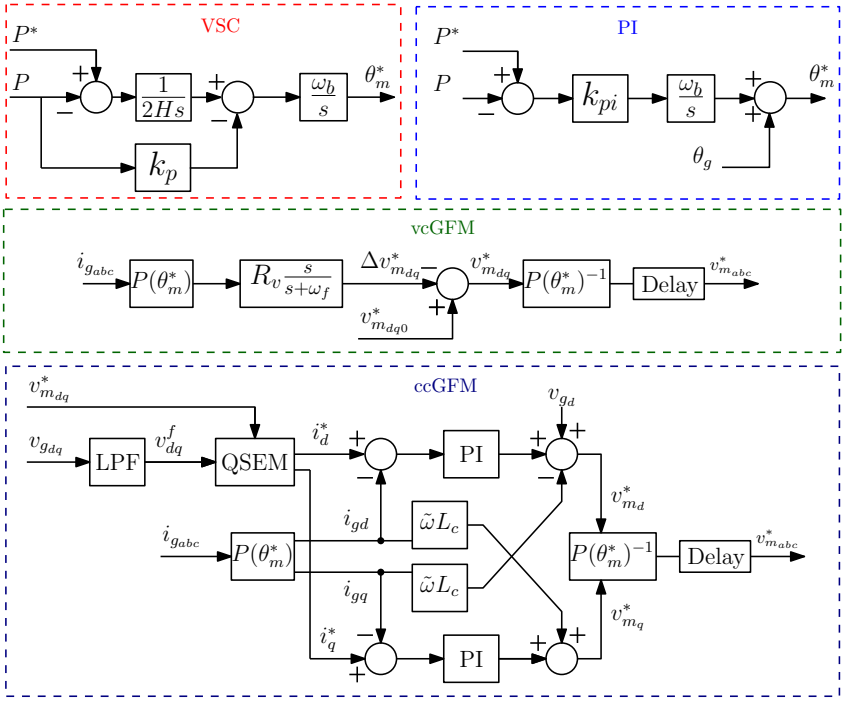



**Figure 3.** The vcGFM, ccGFM and piGFM control structures.

The parameters and bandwidths of the different control loops in the GFM structures are defined as follows: $H$ and $\xi$ represent the parameters of the VSM power loop, $\omega_{QSEM}$, $\omega_{LPF}$, $\omega_{cc}$, and $\omega_{pi}$ correspond to the bandwidths of the QSEM model, the low-pass filter (LPF), the current control loop, and the PI power control loop, respectively, and finally, *Delay* represents the modulation-related delay.

In summary, the control schemes described above form the core of the GFM control strategies. External frequency and voltage control loops can be added to complement the fundamental loops, enabling additional functionalities.

## 3. Admittance Model (AM) of the Power Converter

### *3.1. Definition and Estimation of AM*

The AM of a power converter is typically represented in the $dq$ frame by a $2 \times 2$ matrix for each frequency, as shown in Equation (1). The diagonal admittances, $Y_{dd}$ and $Y_{qq}$, represent the main admittances of the $d$ and $q$ control axis, while the anti-diagonal admittances, $Y_{dq}$ and $Y_{qd}$, represent the coupling admittances between the $d$ and $q$ control axis. Each admittance for a given frequency is a complex number, and for ease of presentation, it is expressed as a magnitude and a phase gain [21]:

$$\begin{bmatrix} \Delta I_d \\ \Delta I_q \end{bmatrix} = \begin{bmatrix} Y_{dd} & Y_{dq} \\ Y_{qd} & Y_{qq} \end{bmatrix} \begin{bmatrix} \Delta V_d \\ \Delta V_q \end{bmatrix} \tag{1}$$

Generally, the AM can be obtained through two different methods. The first method involves analytical derivation when the structure and control parameters are known. In this approach, the equations of both the control and power parts of the converter are expressed in small signals and linearized around the operating point. Detailed derivation procedures are provided in [21]. The second method involves obtaining AM from external measurements of voltage and current at the PCC. This requires the converter to be excited by specific perturbations, such as shunt, series, and mixed excitations, as outlined in [22]. The objective of all these methods is to disturb the output voltage of the converter at the PCC with an excitation signal. The Pseudo-Random Binary Sequence (PRBS) signal

is commonly used as the excitation signal because it can excite the system at multiple frequencies with a single measurement. Typically, two uncorrelated PRBS signals (PRBS1, PRBS2) are employed to measure the AM along both axes, one injected along the d-axis and the other along the q-axis, as explained in [10]. The AM is derived from external measurements using the following Equation (2):

$$\begin{aligned} Y_{dd} &= \left.\frac{\Delta I_d}{\Delta V_d}\right|_{\Delta V_d = PRBS1} \quad , \quad Y_{qq} = \left.\frac{\Delta I_q}{\Delta V_q}\right|_{\Delta V_q = PRBS2} \\ Y_{dq} &= \left.\frac{\Delta I_d}{\Delta V_q}\right|_{\Delta V_q = PRBS2} \quad , \quad Y_{qd} = \left.\frac{\Delta I_q}{\Delta V_d}\right|_{\Delta V_d = PRBS1} \end{aligned} \tag{2}$$

### 3.2. AM Patterns and Characteristics

The definition of the frequency interval for measuring the AM is a crucial step in studying the influence of control on the AM.

Figure 4 illustrates four magnitudes of the admittance $Y_{dd}$: three for the control structures, i.e., ccGFM, vcGFM, and pqGFL, and a fourth for the passive filter L connected to the converter output. The power and control parameters required to obtain the models are presented in Table A1 in the Appendix A. The three AMs of the control structures converge at low frequencies to the same point, referred to as the static admittance related to the operating point [23]. At high frequencies, all AMs converge towards the admittance of the passive filter L. Consequently, in the interval from 1 Hz to 1000 Hz, the three AMs exhibit different behavior because the AMs in this range are related to the structure and control parameters.

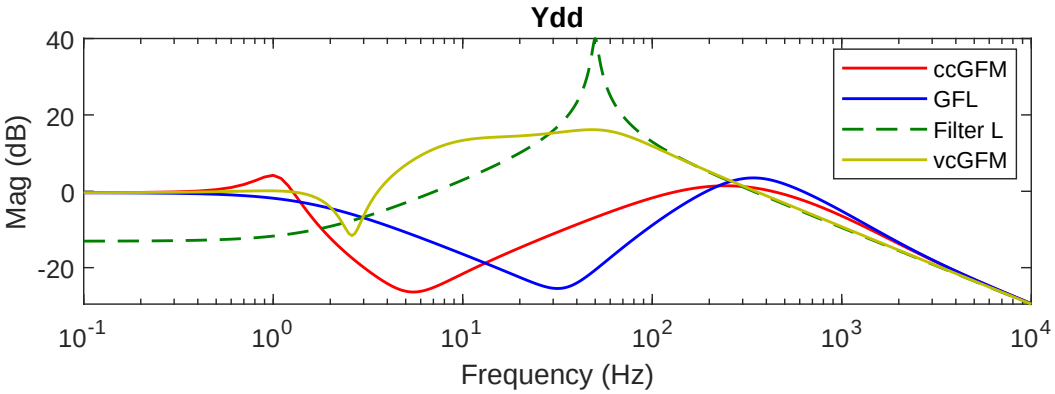



**Figure 4.** The variation of the admittance of the converters with respect to frequency.

The variation of control parameters and operating points alters the shapes of the AM, potentially leading to overlaps in the AM of the two control modes. This overlap may occur in both the magnitude and the phase of the admittances $Y_{dd}$ and $Y_{qq}$. Figure 5 illustrates the magnitudes of the admittance $Y_{dd}$ for the ccGFM and pqGFL structures, with control parameters and operating points varying within realistic intervals. The ranges of variation are detailed in Appendix A Table A2. It is important to note that the overlap does not occur across the entire frequency range but rather in specific zones, depending on the structure and the variation intervals of the parameters. Classical machine learning techniques, as presented in the introduction, generally focus on areas where separation is feasible, without considering the shape of the data. Therefore, in cases where a shift occurs that modifies these zones—due to, for example, an added filter or a variation of a parameter outside the training intervals—the classifiers proposed by these algorithms may fail. Due to these limitations, we are motivated to use more advanced neural network techniques, which

have gained popularity in recent years and have proven their ability to handle data of various types, including sequential data.

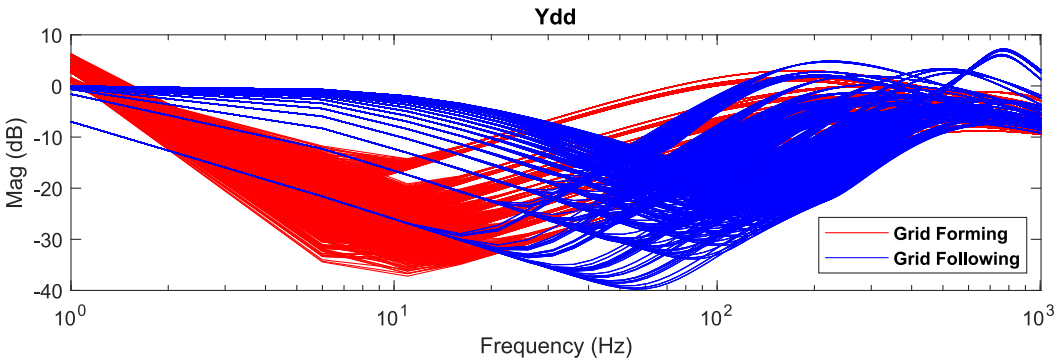



**Figure 5.** AMs of ccGFM and pqGFL in the range of parameters.

## 4. Classification of Control Modes with Neural Networks

### *4.1. Dataset Generation*

As previously mentioned, the main objective is to develop a neural network (NN) model capable of classifying the generated datasets into two classes, referred to as a classification test. The second test aims to evaluate the ability of the trained NN model to generalize the classification to other datasets generated from different control structures, which are not included in the training datasets, in order to assess their generalization capability. This is referred to as a generalization test.

To generate the datasets required for both tests, Matlab/Simulink simulations were developed using the control structures presented in Section 2. The operating points and control parameters of each control structure were sequentially varied within realistic intervals, as shown in Appendix A Table A2. These intervals are determined by physical constraints, such as the switching frequency, where the control loop speed should not exceed the switching frequency [24]. Additional constraints are imposed by Grid Codes [25], such as voltage and power response times [23].

For each set of control parameters, the magnitude and phase of the admittances $Y_{dd}$ and $Y_{qq}$, along with the mode, were recorded as sample in the datasets, the coupling admittances $Y_{dq}$ and $Y_{qd}$ are not considered in this study. The flowchart in Figure 6 illustrates the steps involved in generating the datasets.

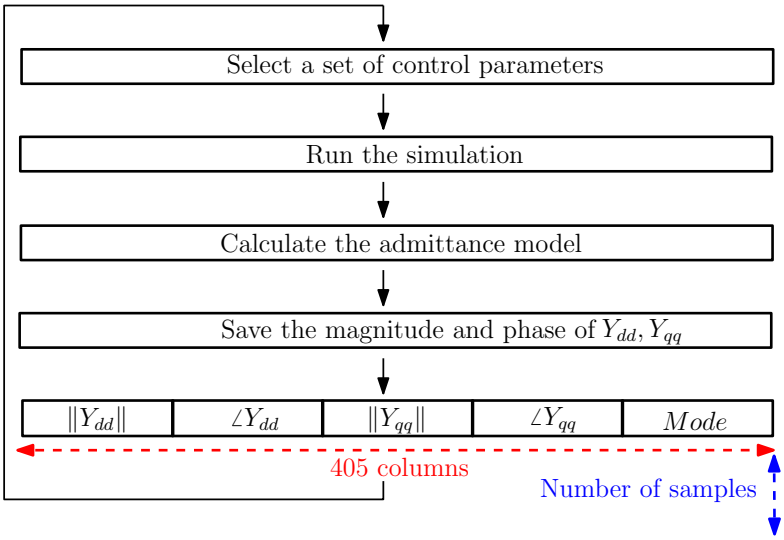



**Figure 6.** Flowchart of dataset generation.

AM data are measured at 101 distinct frequencies, covering the range from 1 Hz to 1000 Hz, resulting in a total of 404 distinct inputs (magnitudes and phases). Details on the frequencies generated by the PRBS can be found in Appendix A Table A2. The model output is the control mode, expressed in Boolean form (0 for GFM and 1 for GFL).

#### 4.1.1. Classification Test

In the context of our classification study, a balanced dataset comprising 5000 samples for each control mode was generated. These samples were distributed across four control structures: cvGFM and ccGFM for the GFM mode, and pvGFL and pqGFL for the GFL mode. To analyze this dataset, the t-distributed Stochastic Neighbor Embedding (t-SNE) method was employed. t-SNE is a non-linear dimensionality reduction technique designed to preserve local relationships between samples while projecting high-dimensional data into a lower-dimensional space. Unlike linear approaches such as PCA, t-SNE excels at revealing complex structures in data, making it particularly suitable for visualizing clusters or overlaps in high-dimensional datasets. Using t-SNE, each sample, initially consisting of 404 points, was represented as a single point in a two-dimensional space, providing an intuitive and synthetic view of the data. The results of the t-SNE analysis, depicted in Figure 7, illustrate the distribution of samples corresponding to the different control modes in the two-dimensional space. Distinct clusters emerge, though overlapping regions are also observed between the control modes. These overlaps highlight the complexity of the classification task and emphasize the need for robust and precise methods to effectively differentiate between these control modes.

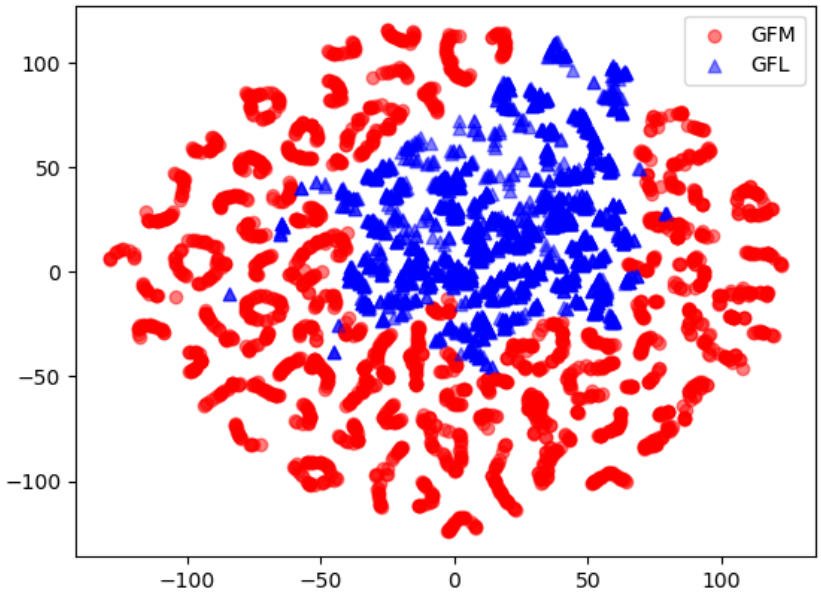



**Figure 7.** t-SNE of the full training dataset.

#### 4.1.2. Generalization Test

In the generalization test, a dataset of 5000 samples was generated by varying the operating points and control parameters, as previously described, but using two control structures: viGFL and piGFM. However, these two datasets were not used during the training of the NN model. The objective is to evaluate the ability of the trained NN model to generalize its classification to new data derived from control structures different from those used during training. A t-SNE analysis was also performed on the classification and generalization datasets to visualize the position of the generalization data relative to the training data. Figure 8 suggests that the generalization data are positioned between the

two modes, even though they belong to two distinct modes. This overlap highlights the challenge of the generalization test and makes the task particularly demanding.

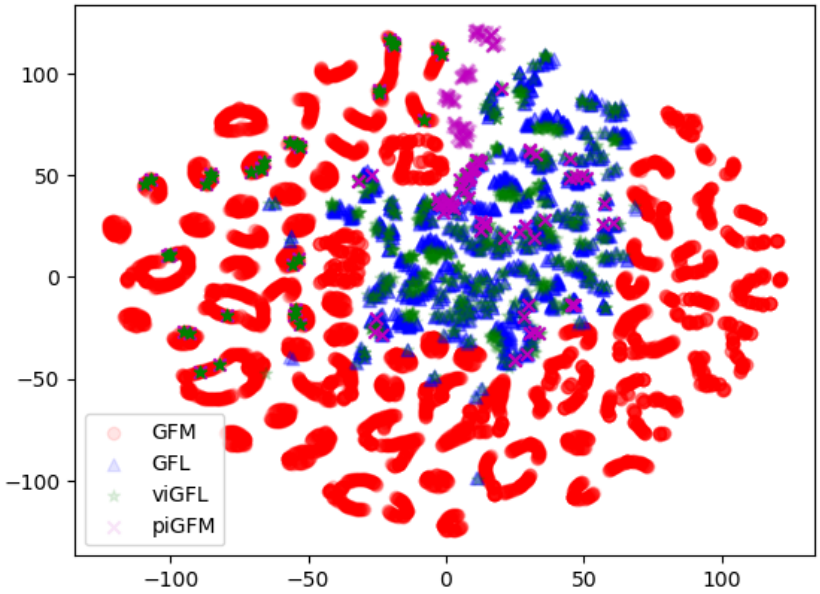



**Figure 8.** t-SNE with generalization dataset (green and magenta).

#### 4.1.3. Training and Performance Evaluation of a Neural Network Model

To conduct the tests previously described using a neural network (NN), it is essential to follow the steps outlined below:

- Selecting the type and structure of the NN: Choose an architecture suitable for the classification task.
- Data preparation: Clean the data by removing missing values, then normalize the data to facilitate the learning process.
- Data splitting: Divide the dataset into two subsets: 80% for training and 20% for testing.

The NN model is trained using the training dataset, while the test dataset is used for evaluation [26]. The selection of hyperparameters, such as the number of layers, kernel sizes, etc., is performed using the Grid Search method, in which multiple combinations of hyperparameters are tested, and the best combination is selected.

The most commonly used metric for evaluating NN performance in classification tasks is accuracy, which measures the proportion of correct predictions among all predictions. Other metrics may be employed in cases where class imbalance is present; however, this is not applicable here, as the designer can generate an equal number of samples for each class through simulations.

To assess the classification capability of the NN model for the two control modes, accuracy is calculated using the following equation:

$$\text{Accuracy} = \frac{PC1 + PC2}{\text{Total}}, \tag{3}$$

where:

- $PC1$ is the number of correctly classified samples in the GFM mode;
- $PC2$ is the number of correctly classified samples in the GFL mode;
- Total is the total number of samples, including both correctly and incorrectly classified ones.

### *4.2. Feedforward Neural Network (FNN)*

An FNN is a type of NN model where information flows through the different layers of the network in a unidirectional manner, from input to output. The typical structure of an FNN consists of three types of layers: an input layer that receives initial data, one or more hidden layers that perform non-linear transformations on the data, and an output layer that produces the final prediction, as in this case the control mode Figure 9. Each neuron in a layer is connected to every neuron in the subsequent layer, and each connection is associated with a weight that is adjusted during the learning process. Neurons in the hidden layers use non-linear activation functions to introduce complexity into the representation of data. During the training phase, the network adjusts its weights using backpropagation to minimize the gap between the model predictions and the actual values. This process enables the FNN to learn complex patterns and generalize to new data during the testing phase [27].

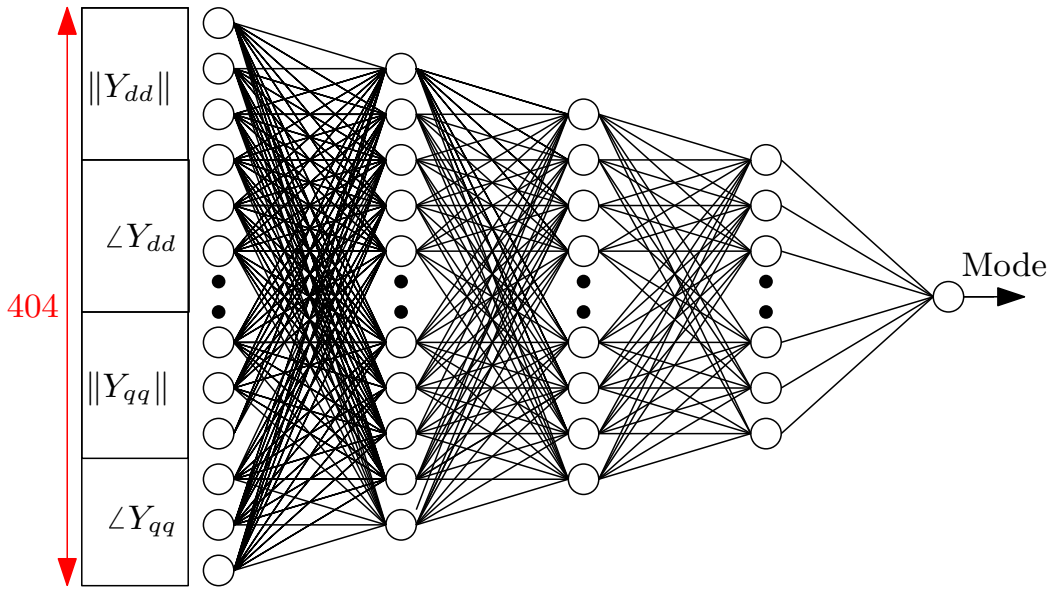



**Figure 9.** Classification with a Feedforward Neural Network.

#### 4.2.1. Proposed Structure of the FNN

A model of the FNN has been designed to conduct the two tests mentioned earlier. This model was implemented using the TensorFlow library of Python. The developed FNN consists of five layers, including an input layer, an output layer, and three hidden layers. Details regarding the number of neurons, the type of activation function in each layer, as well as the solver, loss function, and the number of epochs are specified in Table 2.

**Table 2.** The structure and options of the FNN model.

| Structure | | | |
|---|---|---|---|
| **Layer** | **Type** | **Parameters** | **Activation Function** |
| 1 | Input | 404 | - |
| 2 | Dense | 256 | relu |
| 3 | Dense | 128 | relu |
| 4 | Dense | 32 | relu |
| 5 | Dense | 1 | sigmoid |
| Options | | | |
| Optimizer | | adam | |
| Loss Function | | binary crossentropy | |
| Epochs | | 50 | |
| Batch Size | | 64 | |

#### 4.2.2. Results and Discussion

Figure 10 and Table 3 present the confusion matrix and accuracy metric for both the classification and generalization tests. According to these results, the FNN model effectively classified the two control modes, achieving 100% accuracy in the classification tests. However, during the generalization test, the accuracy decreased to 87%. Thus, while the FNN model excels in classifying the two control modes when control structures are present in the training dataset, it exhibits limitations in generalizing to new structures from both modes that do not exist in the training data.


(**a**) Classification test (**b**) Generalization test

**Figure 10.** FNN test results: confusion matrices.

**Table 3.** The results of the FNN classification and generalization tests.

| Test | Classification | Generalization |
|---|---|---|
| Accuracy | 100% | 87% |

To visualize how the data from both tests are classified with the FNN, a modification was made to the structure of the FNN model. In this adaptation, the last layer (output layer) was removed while retaining the other trained layers with their respective weights. The data used in both tests were then fed into the FNN model, and the output of these data in the embedding space was visualized using t-SNE. Figure 11 represents the data from Figure 8 after processing by the FNN, demonstrating that the FNN successfully classified both modes (red and blue) but generalized poorly (green and magenta datasets).

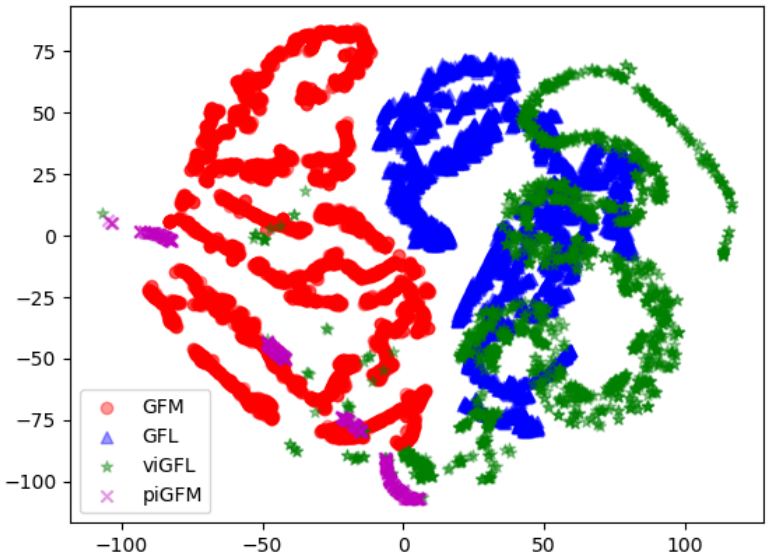



**Figure 11.** t-SNE of the dataset in the embedding space classified by the FNN.

The inability of FNNs to process data presented in grid-like structures and the need to transform the data into a one-dimensional vector (1D) result in the loss of various indicators within the data, as demonstrated in our example. Measurements are presented sequentially, with each measurement representing a series of information taken at different frequencies. Additionally, for each admittance measurement at a given frequency, there are two components: magnitude and phase. Thus, the transformation into a 1D vector leads to the loss of all this information. Another important aspect in our example is that the two admittances, $Y_{dd}$ and $Y_{qq}$, represent two distinct axes. Therefore, it is preferable to handle each admittance individually and make a classification decision based on both. Many details about the features used in this example are lost due to the transformation into a 1D vector. To enhance the results of the two tests presented earlier, it is recommended to use NN algorithms capable of handling sequential data.

### *4.3. Convolutional Neural Network (CNN)*

CNN are specialized NN designed specifically for processing grid-like data, such as signals and images [28,29]. Acknowledged for their effectiveness, particularly in tasks related to object recognition and computer vision, CNN leverage convolution operations to extract crucial features from input data, thereby reducing complexity while preserving the hierarchical structure of features [30,31]. Within a CNN, various types of layers contribute to the network functionality:

- The Input layer, which is the first layer of the CNN, is initially sized based on the input data and corresponds to the size of an individual data sample [32].
- The Convolutional layers, crucial components of CNNs, are widely used to detect features and patterns in the inputs. A convolutional layer applies a set of filters (or kernels) in the form of matrices to an input by performing convolution operations, thereby producing an output called a feature map. The filters enable the detection of local patterns such as edges, textures, or more complex motifs [29]. The learning of filter weights occurs through gradient backpropagation during the network training phase, minimizing a loss function. The equation of the convolutional layer can be represented as follows:

$$y = f\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} * w_{ij}) + b\right), \quad (4)$$

where $x$ represents the inputs, $w$ denotes the filter weights that change during training, $b$ is the bias term that also changes during model training, and $*$ is the convolution operator [26]. To identify complex patterns, the outputs of convolution pass through a non-linear function called the activation function $f$, helping in the detection of complex patterns.

- The Max Pooling layers, positioned after convolutional layers, are responsible for reducing the spatial dimension of the feature map. By subdividing the map into disjointed regions (typically $2 \times 2$ or $3 \times 3$ windows) and taking the maximum value from each region, max pooling retains the most important features while reducing the size of the feature map. The learning process for max pooling layers is generally non-trainable, meaning this layer has no weights to adjust during backpropagation [30].

Additional layers, such as normalization and reshaping layers, can be incorporated in specific cases to enhance network performance and reshape data into multiple sizes. The selection of these layers depends on the nature of the problem and the specific objectives of the CNN.

CNN architectures can take various forms, incorporating the mentioned layers or integrating dense layers from an FNN. This adaptability enables CNNs to address diverse tasks and challenges in processing grid-like data, showcasing their versatility across various applications [33].

#### 4.3.1. Proposed Structure of the CNN

Figure 12 illustrates the structure of the proposed CNN, designed to classify the two control modes GFM and GFL. This structure is developed using Python TensorFlow library. The proposed structure consists of two parts: a convolution part and a part with dense layers. Table 4 presents details about both parts and the parameters used.

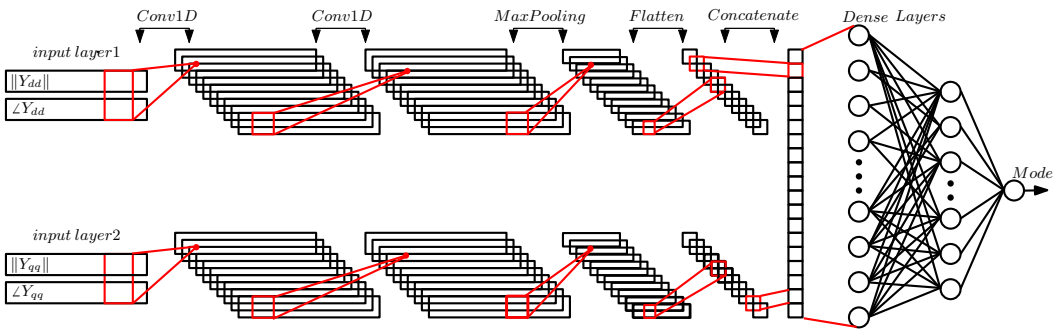



**Figure 12.** The proposed CNN structure.

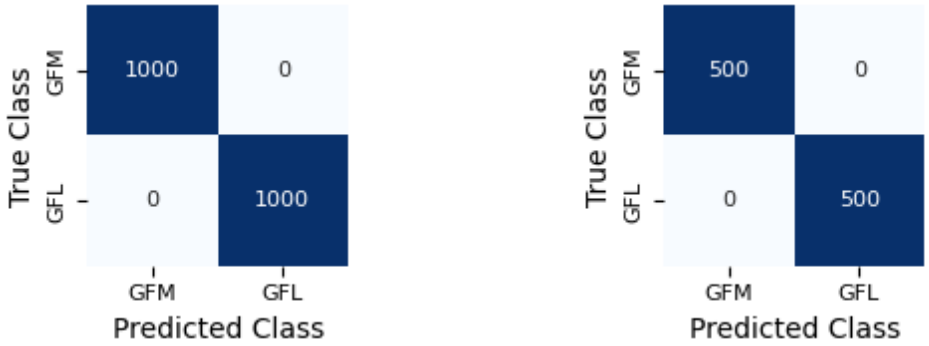
The objective of the convolution part is to identify features and patterns in the admittances. After this step, the second part aims to classify the extracted features into two classes. For the convolution part, the important features in the two admittances are identified separately; two convolution inputs are used, one for $Y_{dd}$ and the other for $Y_{qq}$. The two admittances are processed in parallel with the same type of convolution layers, which preserves the relationship between magnitude and phase as well as the frequency sequence. After identifying the feature maps of the two admittances with the max-pooling layers, another layer called flatten is used to reshape the features into a 1D vector to facilitate its connection to the second part of the CNN structure. The 1D vectors of important features from the two admittances are concatenated and connected with dense layers, whose objective is to classify them into two classes.

**Table 4.** The structure and options of the CNN model.

| Structure | | | |
|---|---|---|---|
| **Layer** | **Type** | **Parameters** | **Activation Function** |
| CNN 1D | | | |
| 1 | Input | shape = (101, 2) | - |
| 2 | Conv1D | 16 filters, kernel size = 3 | relu |
| 3 | Conv1D | 32 filters, kernel size = 3 | relu |
| 4 | MaxPooling1D | Pool size = 2 | - |
| 5 | Flatten | - | - |
| FNN | | | |
| 6 | Dense | 256 | relu |
| 7 | Dense | 16 | relu |
| 8 | Dense | 1 | sigmoid |
| Options | | | |
| Optimizer | | adam | |
| Loss Function | | binary crossentropy | |
| Epochs | | 50 | |
| Batch Size | | 64 | |

#### 4.3.2. Classification Results and Discussion

To assess the capability of the proposed CNN model in classifying the two control modes, the classification and generalization tests, previously conducted on the FNN model, were replicated on the CNN model. However, the input data for this model were reformulated differently. As explained earlier, the proposed CNN structure accepts two inputs, meaning each AM sample is separated into two components: the magnitudes and phases of $Y_{dd}$ on input 1, and the magnitudes and phases of $Y_{qq}$ on input 2. Both tests were carried out using the same data as that used for the FNN structure. Figure 13 and Table 5 present the accuracy of the classification and generalization tests. According to the results of the proposed CNN model, it achieved 100% success in both classification and generalization, showing a 13% improvement over the FNN structure in the generalization test.




(**a**) Classification test (**b**) Generalization test.

**Figure 13.** CNN test results: confusion matrices.

**Table 5.** The results of CNN classification and generalization tests.

| Test | Classification | Generalization |
|---|---|---|
| Accuracy | 100% | 100% |

To visualize how the data from both tests are classified with the CNN, the same modification applied to the FNN is carried out on the CNN, where the last layer of the structure is removed. Then, the data used in both tests are fed into the CNN, and the output of these data in the embedding space is visualized using t-SNE. Figure 14 represents the data from Figure 8 after processing by the CNN, demonstrating that the CNN successfully classified and generalized in both tests.

Table 6 compares the results of the two NN with those from other methods presented in [11]. It is observed that the CNN algorithm proposed in this paper yielded the best results in both tests conducted.

**Table 6.** Classification and generalization results with different classification algorithms.

| Machine Learning Algorithm | Classification | Generalization |
|---|---|---|
| Convolutional Neural Network (CNN) | 100% | 100% |
| Feedforward Neural Network (FNN) | 100% | 87% |
| Linear Regression (LR) | 94% | 80% |
| Random Forest (RF) | 98% | 91% |
| Decision Tree (DT) | 92% | 40% |
| Naive Bayes Classifier (NBC) | 92% | 80% |
| XGBoost (XGB) | 95% | 79% |
| Support Vector Machine (SVM) | 95% | 84% |
| K-Nearest Neighbors (KNN) | 94% | 84% |

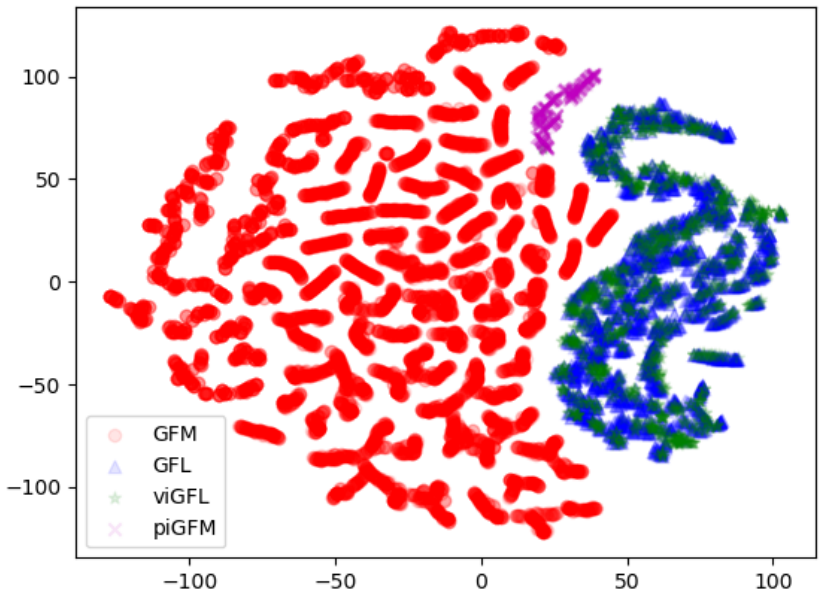



**Figure 14.** t-SNE of the dataset in the embedding space classified by the CNN.

### *4.4. Discussion on the Results of FNN and CNN*

The architecture proposed in this article, with the use of convolutional layers, allowed the CNN model to demonstrate improved generalization capability compared to the FNN model, which relies solely on dense layers. This component enabled the CNN to process the $Y_{dd}$ and $Y_{qq}$ admittances separately and in parallel, while handling the admittances in sequence form and preserving the relationship between the magnitude and phase of each frequency. Thus, this approach provides the advantage of efficiently extracting the essential features from both admittances and transmitting these features to a second part of the model, based on dense layers, for classification. However, the direct use of dense layers, as in the FNN model, with a transformation of the data into a 1D vector, impacted

the generalization results, as several crucial pieces of information were lost during this transformation, such as the sequential structure of the data and the relationship between the magnitude and phase for each frequency.

## 5. Conclusions

This article explored the classification of the control mode of a power converter connected to the grid, considered as a black box. The data used for this classification are based on the admittance measured at the PCC. The classification and generalization capabilities of two neural networks, an FNN and a CNN, were evaluated with two tests. Although the classification results are perfect for both modes whose data belong to the training set, in the generalization test, the CNN demonstrated a greater capacity compared to the FNN. The processing of the two admittances, $Y_{dd}$ and $Y_{qq}$, each with their magnitude and phase in parallel, maintains the sequential frequency relationship and combines the important information for classification, yielding very good results for the CNN. Future work could focus on testing the developed algorithms on other control structures, as well as on the identification of the parameters of these control structures.

**Author Contributions:** Conceptualization, R.O., M.L., J.-Y.D., P.Y., X.G. and F.C.; Methodology, R.O., M.L., J.-Y.D., P.Y., X.G. and F.C.; Investigation, M.L.; Writing—original draft, R.O.; Writing—review & editing, M.L., J.-Y.D., P.Y., X.G. and F.C.; Supervision, J.-Y.D., P.Y., X.G. and F.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is part of the ANR DELTWINCO Project funded by the French National Research Agency (ANR) (LabEx BASC; ANR-21-CE05-0038).

**Data Availability Statement:** The data will be made available upon request to the corresponding authors.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

**Table A1.** System parameters of the converters in Figure 4.

| Control Structures | Parameters | Values |
|---|---|---|
| VSC | $S_{nom}$, $S_b$ | 1.044 GVA |
| | $U_{nom}$, $U_b$ | 400 kV |
| | $P_{nom}$ | 1 GW |
| | $Q_{max}$ | 300 MVar |
| Filter | $L_c$ | 0.15 pu |
| | $R_c$ | 0.005 pu |
| Grid | $L_g$ | 0.5 pu |
| | $R_g$ | 0.05 pu |
| pqGFL, pvGFL | $\omega_p$ | 30 rad/s |
| | $\omega_q$ | 30 rad/s |
| | $\omega_v$ | 15 rad/s |
| | $\omega_{pll}$ | 500 rad/s |
| | $\omega_{cc}$ | 1000 rad/s |
| | $\omega_{cc}$ | 1000 rad/s |
| | $Delay$ | 100 µs |
| vcGFM, ccGFM | $H$ | 1 s |
| | $\omega_f$ | 50 rad/s |
| | $\xi$ | 1 |
| | $\omega_{LPF}$ | 83 rad/s |
| | $\omega_{cc}$ | 800 rad/s |
| | $Delay$ | 100 µs |

**Table A2.** Variation ranges of parameters for the training and generalization datasets.

| Control Structures | Parameters | Values |
|---|---|---|
| Operating Points | $P^*$ | $[-1, 1]$ p.u. |
| | $Q^*$ | $[-0.4, 0.4]$ p.u. |
| | $V^*$ | [0.9, 1.1] p.u. |
| pqGFL, pvGFL viGFL | $\omega_p$ | [6, 38 ] rad/s |
| | $\omega_q$ | [6, 38] rad/s |
| | $\omega_v$ | [3, 15] rad/s |
| | $\omega_{pll}$ | [50, 1500] rad/s |
| | $\omega_{cc}$ | [1200, 3000] rad/s |
| | $Delay$ | [50, 200] µs |
| vcGFM, ccGFM , piGFM | $H$ | [0.7, 5] s |
| | $\omega_f$ | [40, 60] rad/s |
| | $\omega_{LPF}$ | [50, 83] rad/s |
| | $\omega_{cc}$ | [1200, 3000] rad/s |
| | $\omega_{pi}$ | [10, 100] rad/s |
| | $Delay$ | [50, 200] µs |
| PRBS 1 | Order | 8 |
| | $f_{max}$ | 1001 |
| PRBS 2 | Order | 7 |
| | $f_{max}$ | 1020 |

## References


1. Bouzid, A.M.; Guerrero, J.M.; Cheriti, A.; Bouhamida, M.; Sicard, P.; Benghanem, M. A Survey on Control of Electric Power Distributed Generation Systems for Microgrid Applications. *Renew. Sustain. Energy Rev.* **2015**, *44*, 751–766. [CrossRef]
2. Rocabert, J.; Luna, A.; Blaabjerg, F.; Rodriguez, P. Control of Power Converters in AC Microgrids. *IEEE Trans. Power Electron.* **2012**, *27*, 4734–4749. [CrossRef]
3. Barnes, M.; Kondoh, J.; Asano, H.; Oyarzabal, J.; Ventakaramanan, G.; Lasseter, R.; Hatziargyriou, N.; Green, T. Real-World MicroGrids: An Overview. In Proceedings of the 2007 IEEE International Conference on System of Systems Engineering, San Antonio, TX, USA, 16–18 April 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–8.
4. Liu, C.; Zhi, Y.; Su, Z.; Yang, Z.; Yin, L.; Man, J.; Yang, Y. Active Support Pre-Synchronization Control and Stability Analysis Based on the Third-Order Model of Synchronous Machine. *Energies* **2024**, *17*, 5072. [CrossRef]
5. Liu, Y.; Huang, R.; Du, W.; Singhal, A.; Huang, Z. Highly-Scalable Transmission and Distribution Dynamic Co-Simulation with 10,000+ Grid-Following and Grid-Forming Inverters. *IEEE Trans. Power Deliv.* **2024**, *39*, 578–590. [CrossRef]
6. Lamrani, Y.; Colas, F.; Van Cutsem, T.; Cardozo, C.; Prevost, T.; Guillaud, X. Investigation of the Stabilizing Impact of Grid-Forming Controls for LCL-connected Converters. In Proceedings of the 2023 25th European Conference on Power Electronics and Applications, Aalborg, Denmark, 4–8 September 2023; IEEE: Piscataway, NJ, USA, 2023.
7. ENSTOE. Stability Management in Power Electronics Dominated Systems: A Prerequisite to the Success of the Energy Transition; 2022. Available online: https://eepublicdownloads.azureedge.net/clean-documents/Publications/Position%20papers%20and%20reports/220616_entso-e_pp_stability_management.pdf (accessed on 22 November 2024).
8. National Grid ESO. Grid Code UK. Available online: https://www.nationalgrideso.com/document/278491/download (accessed on 28 October 2024).
9. Wang, X.; Blaabjerg, F. Harmonic Stability in Power Electronic-Based Power Systems: Concept, Modeling, and Analysis. *IEEE Trans. Smart Grid* **2019**, *10*, 2858–2870. [CrossRef]
10. Gong, H.; Wang, X.; Yang, D. DQ-Frame Impedance Measurement of Three-Phase Converters Using Time-Domain MIMO Parametric Identification. *IEEE Trans. Power Electron.* **2021**, *36*, 2131–2142. [CrossRef]
11. Ouali, R.; Dieulot, J.-Y.; Yim, P.; Guillaud, X.; Colas, F.; Wu, Y.; Wu, H. Machine Learning Classification of Power Converter Control Mode. *arXiv* **2024**, arXiv:2401.10959.
12. Evstatiev, B.I.; Trifonov, D.T.; Gabrovska-Evstatieva, K.G.; Valov, N.P.; Mihailov, N.P. PV Module Soiling Detection Using Visible Spectrum Imaging and Machine Learning. *Energies* **2024**, *17*, 5238. [CrossRef]
13. Ding, Z.; Qing, H.; Zhou, K.; Huang, J.; Liang, C.; Liang, L.; Qin, N.; Li, L. Integrated Energy System Load Forecasting with Spatially Transferable Loads. *Energies* **2024**, *17*, 4843. [CrossRef]

14. Teodorescu, R.; Liserre, M.; Rodriguez, P. *Grid Converters for Photovoltaic and Wind Power Systems*; IEEE: Piscataway, NJ, USA; Wiley: Hoboken, NJ, USA, 2011.
15. Wang, X.; Taul, M.G.; Wu, H.; Liao, Y.; Blaabjerg, F.; Harnefors, L. Grid-Synchronization Stability of Converter-Based Resources—An Overview. *IEEE Open J. Ind. Appl.* **2020**, *1*, 115–134. [CrossRef]
16. Qoria, T.; Rokrok, E.; Bruyere, A.; François, B.; Guillaud, X. A PLL-Free Grid-Forming Control With Decoupled Functionalities for High Power Transmission System Applications. *IEEE Access* **2020**, *8*, 197363–197378. [CrossRef]
17. D'Arco, S.; Suul, J.A. Equivalence of Virtual Synchronous Machines and Frequency-Droops for Converter-Based MicroGrids. *IEEE Trans. Smart Grid* **2014**, *5*, 394–395. [CrossRef]
18. Rosso, R.; Wang, X.; Liserre, M.; Lu, X.; Engelken, S. Grid Forming Converters: Control Approaches, Grid-Synchronization, and Future Trends—A Review. *IEEE Open J. Ind. Appl.* **2021**, *2*, 93–109. [CrossRef]
19. Zhang, L.; Harnefors, L.; Nee, H.-P. Power-Synchronization Control of Grid-Connected Voltage-Source Converters. *IEEE Trans. Power Syst.* **2010**, *25*, 809–820. [CrossRef]
20. Mo, O.; D'Arco, S.; Suul, J.A. Evaluation of Virtual Synchronous Machines With Dynamic or Quasi-Stationary Machine Models. *IEEE Trans. Ind. Electron.* **2017**, *64*, 5952–5962. [CrossRef]
21. Wang, X.; Harnefors, L.; Blaabjerg, F. Unified Impedance Model of Grid-Connected Voltage-Source Converters. *IEEE Trans. Power Electron.* **2018**, *33*, 1775–1787. [CrossRef]
22. Shen, Z.; Jaksic, M.; Mattavelli, P.; Boroyevich, D.; Verhulst, J.; Belkhayat, M. Design and Implementation of Three-Phase AC Impedance Measurement Unit (IMU) With Series and Shunt Injection. In Proceedings of the 2013 IEEE Applied Power Electronics Conference and Exposition (APEC), Long Beach, CA, USA, 17–21 March 2013; IEEE: Piscataway, NJ, USA, 2013.
23. Morris, J.F.; Ahmed, K.H.; Egea-Àlvarez, A. Analysis of Controller Bandwidth Interactions for Vector-Controlled VSC Connected to Very Weak AC Grids. *IEEE J. Emerg. Sel. Top. Power Electron.* **2020**, *9*, 7343–7354. [CrossRef]
24. Sun, J.; Li, M.; Zhang, Z.; Xu, T.; He, J.; Wang, H.; Li, G. Renewable Energy Transmission by HVDC Across the Continent: System Challenges and Opportunities. *CSEE J. Power Energy Syst.* **2017**, *3*, 353–364. [CrossRef]
25. National Grid. The Grid Code. Available online: https://www.nationalgrid.com/sites/default/files/documents/8589935310-Complete%20Grid%20Code.pdf (accessed on 28 October 2024).
26. Wen, B.; Boroyevich, D.; Burgos, R.; Mattavelli, P.; Shen, Z. Analysis of D-Q Small-Signal Impedance of Grid-Tied Inverters. *IEEE Trans. Power Electron.* **2016**, *31*, 675–687. [CrossRef]
27. Kapoor, A.; Gulli, A.; Pal, S.; Chollet, F. *Deep Learning with TensorFlow and Keras: Build and Deploy Supervised, Unsupervised, Deep, and Reinforcement Learning Models*, 3rd ed.; Packt Publishing: Birmingham, UK, 2022.
28. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
29. Zou, J.; Gao, Y.; Frieges, M.H.; Pirkle, D.L.; Wetzel, E.D.; Mosby, J.M.; Tchoul, M.N.; Holt, A.P. Machine Learning for Battery Quality Classification and Lifetime Prediction Using Formation Data. *Energy AI* **2024**, *18*, 100451. [CrossRef]
30. Harish, A.; Asok, P.; Jayan, M.V. A Comparative Evaluation of Stacked Auto-Encoder Neural Network and Multi-Layer Extreme Learning Machine for Detection and Classification of Faults in Transmission Lines Using WAMS Data. *Energy AI* **2023**, *14*, 100301. [CrossRef]
31. Wang, M.-H.; Lu, S.-D.; Liao, R.-M. Fault Diagnosis for Power Cables Based on Convolutional Neural Network with Chaotic System and Discrete Wavelet Transform. *IEEE Trans. Power Deliv.* **2022**, *37*, 582–590. [CrossRef]
32. Ren, T.; Han, T.; Guo, Q.; Li, G. Analysis of Interpretability and Generalizability for Power Converter Fault Diagnosis Based on Temporal Convolutional Networks. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–11. [CrossRef]
33. Lu, C.; Li, J.; Chen, K.; Zhou, W.; Wu, Q.; Ke, J. System-Level Parameters Identification for DC-DC Converters Based on Artificial Neural Network Algorithm. In Proceedings of the 2023 IEEE Energy Conversion Congress and Exposition (ECCE), Nashville, TN, USA; IEEE: Piscataway, NJ, USA, 2023.