*sustainability*

*Article*

# Prediction of Faults Location and Type in Electrical Cables Using Artificial Neural Network

**Ana-Maria Moldovan * and Mircea Ion Buzdugan**

Faculty of Building Services Engineering, Technical University of Cluj-Napoca, 28 Memorandumului Str., 400114 Cluj-Napoca, Romania
* Correspondence: ana.moldovan@insta.utcluj.ro

**Abstract:** Detecting and locating faults in electrical cables has been a permanent concern regarding electrical power distribution systems. Over time, several techniques have been developed aiming to manage these faulty situations in an efficient way. These techniques must be fast, accurate, but, above all, efficient. This paper develops a new approach for detecting, locating, classifying, and predicting faults, particularly in different types of short-circuits in electrical cables, based on a robust artificial neural network technique. The novelty of this approach lies in the ability of the method to predict fault's location and type. The proposed method uses the Matlab and Simulink platform and comprises four consecutive stages. The first one is devoted to the development of the Simulink model. The second one implies a large number of simulations in order to generate the necessary dataset for training and testing the artificial neural network model (ANN). The following stage uses the ANN to classify the location and the type of potential faults. Finally, the fourth stage consists of predicting the location and the type of future faults. In order to reduce the time and the resources of the simulation process, a virtual machine is used. The study reveals the efficiency of the method, and its ability to successfully predict faults in real-world electrical power systems.

**Keywords:** electrical cables; detecting, locating and predicting faults; artificial neural network; Classification Learner app

**Citation:** Moldovan, A.-M.; Buzdugan, M.I. Prediction of Faults Location and Type in Electrical Cables Using Artificial Neural Network. *Sustainability* **2023**, *15*, 6162. https://doi.org/10.3390/su15076162

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 7 March 2023
Revised: 29 March 2023
Accepted: 30 March 2023
Published: 3 April 2023

## 1. Introduction

Today, people's lives are entirely dependent on the sustainability of electrical power systems. This supposes that the continuity of the supply of the electrical power distribution systems is mandatory. In this respect, electrical cables have the important role of linking all components of a power system.

The presented method depicted in the following sections aims to contribute to a higher degree of sustainability of the distribution power systems, accelerating the maintenance process in fault cases, due to its accuracy in predicting the location and the type of faults in the energy cables.

A fault in a cable directly affects the sustainability of the system, and the duration of a power outage, being crucial to ensure the cables' integrity during their entire operation time [1–3]. However, if any defect occurs in a cable, the reaction must be as fast as possible to reduce to a minimum the duration of its clearing time [4,5].

Methodologies of detecting faults in electrical cables have evolved with the advancements in technology, with several methods being implemented: time domain reflectometry technique, impedance-based method, knowledge-based method, traveling wave methods or hybrid methods [6]. Each of them has its benefits and its limitations [7,8]. For instance, time domain reflectometry can be successfully used in the case of a single cable, being useless for systems that have more than two branches [7,9,10].

Algorithms based on artificial intelligence (AI) propose solutions that are able to manage these more complex systems [11–13]. The artificial neural network technique

(ANN) provides efficient pattern recognition algorithms that can be applied in predicting, locating and classifying faults [14,15]. The ANN technique is able to solve nonlinear problems, based on learned experiences, which implies different possible configurations of the electrical distribution systems [16–20]. At the same time, the main ANN algorithms features are robustness, generalizability, and noise immunity [21–23].

A complex method of detecting and locating faults in electrical cables should return their exact type and location [24–26]. For a three-phase electrical power system, the fault types are interruptions or short-circuits, with the last ones being: single line to ground fault, line to line fault, double line to ground fault, three-line fault and three line to ground fault [27–29].

This paper presents an efficient method of detecting, locating, and predicting the different types of short-circuits in electrical cables. In order to develop this method, a model of distribution electrical system has been modeled in Simulink, followed by the use of the ANN technique of the Classification Learner app available in Matlab.

The model selected for analysis contains a three-phase source block of 20 kV and several distributed parameters line blocks with a total of 22 km of cables. The ANN technique is based on data generated by the Simulink model of the electrical distribution system. The method aims to reach a high rate of validation accuracy of the trained model delivered by the Classification Learner app. After running a large number of simulations, more precisely 6150 simulations, a 98% rate of validation accuracy was obtained.

The generated data represent the training dataset for the ANN algorithm which has an important impact over the accuracy of the method, since the performance of the method increases with the complexity of the dataset. The case presented below highlights the complexity and advantages of using ANNs methods in predicting, detecting, locating, and classifying short-circuits in complex distribution electrical power systems.

The article is structured in five sections. The present section is the introductory one, highlighting the importance of detecting faults in electrical power distributions systems. In the second one, entitled Materials and Methods, the working principle of the method is presented. Then, the Results section comprises four consecutive stages. The first stage is devoted to the development of the Simulink model. The second stage presents a large number of simulations in order to generate the necessary dataset for training and testing the ANN model. The third one uses the ANN to classify the location and the type of potential faults. Finally, the fourth stage consists of predicting the location and the type of future faults. The fourth and fifth sections are devoted to the discussion, conclusions and further work.

## 2. Materials and Methods

The need to detect and locate faults in electrical systems has generated different methods in the attempt to solve these problems. In the present section, a new approach involving a simulated model of a distribution electrical power system, combined with the benefits of ANN applications, will be presented.

As mentioned above, the method comprises four consecutive stages. The first one is devoted to the development of the model, and the second one presents a large number of simulations in order to generate a dataset necessary for training and testing the ANN model, while the third stage uses ANN to classify the location and the type of faults. Finally, the fourth stage consists of predicting the location and the type of potential future faults.

After modeling the distribution electrical system in Simulink (R2022a), several simulations were performed for different types of short-circuits in different locations of the cables. The results of all simulations have been saved in a database which became the training dataset for the Classification Learner app from Matlab (R2022a). The input data for the Classification Learner app are the measured values of the voltage and current, the responses being either the location of the faults or both their location and type. Based on the trained neural network, the location and the type of a further fault can be predicted.

The model was created using the blocks contained in the Simscape (R2022a) electrical library, a library dedicated to electrical power systems. The developed simulation model contains a three-phase source block, distributed parameters line blocks, three-phase voltage-intensity (VI) measurement blocks, three-phase load blocks and a three-phase fault block able to induce faults in different locations of the cables. The model and these blocks will be detailed in the next section.

Since the process of simulating the model which provides the training set for the neural network is a time-consuming one, it had to be automated. The automation consisted in writing a Matlab code which ran these simulations, and at each simulation, modifying the parameters and saving the data delivered by the measurement blocks.

Once the database is accessible, the whole set of simulation results is introduced in the Classification Learner app from Matlab. At this point, the training process may start. In this last Matlab application, different types of algorithms based on artificial intelligence are available, such as decision trees, discriminant analysis, naive bayes classifiers, support vector machines, nearest neighbor classifiers, ensemble and neural network classifiers [30]. The Classification Learner app allows the training of all these algorithms based on the accuracy of validation, enabling the possibility of choosing the most efficient one. For the present case, the most accurate algorithm turned out to be the medium neural network model. This model can be exported into Matlab workspace, being later used in the prediction of the trained model response for another set of measurements, corresponding to a further fault due to the versatility and complexity of the Matlab and Simulink platform.

The presented method is synthesized in the process diagram in Figure 1.

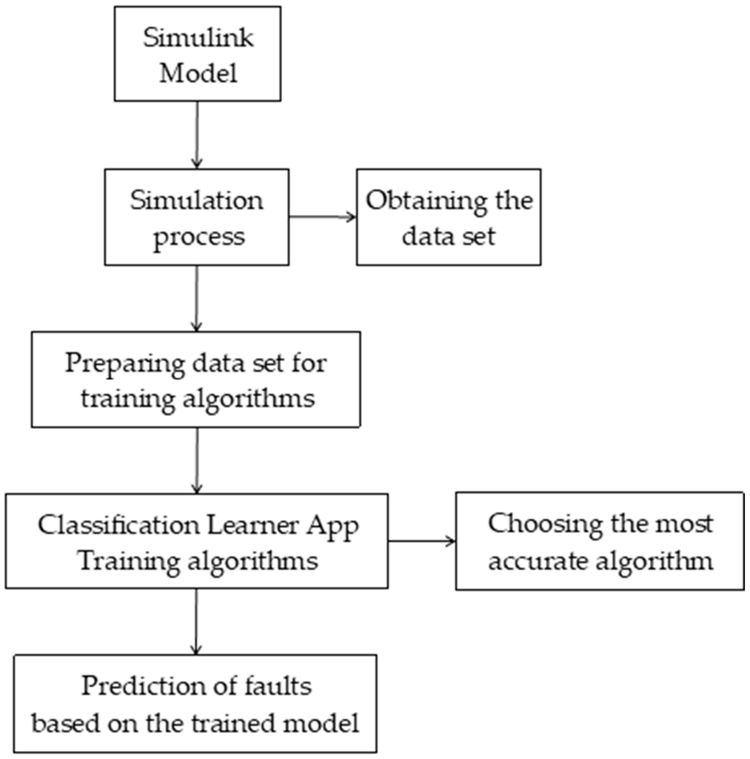

Simulink Model
Simulation process
Obtaining the data set
Preparing data set for training algorithms
Classification Learner App Training algorithms
Choosing the most accurate algorithm
Prediction of faults based on the trained model

**Figure 1.** The process diagram of the method presented.

## 3. Results

As stated above, the case under analysis proposes a method of detecting and locating faults in electrical systems based on the medium neural network algorithm, which can be successfully used in solving faults detection, location, and prediction.

### *3.1. Simulink Model*

This first stage of the presented method is devoted to the development of the Simulink model of a distribution electrical system. The development of the model concept consists of inserting different types of short-circuits in different locations of the system and observing their influence on the measured voltage–current pairs. Based on it, the training dataset necessary in the third stage will be used.

As mentioned in Section 2, in order to develop the simulation model, the Simscape electrical library was used. The selected model of the distribution electrical system presented in Figure 2 contains: a three-phase source block, depicted in green; six subsystems for the six lines L1 to L6, depicted in dark green, which contains the distributed parameters line blocks; eight three-phase voltage-intensity VI measurement blocks B1 to B8, depicted in blue; three-phase load blocks, noncolored; a three-phase fault block, bordered in red; and powergui, the environment block for Simscape electrical specialized power system models, set to discrete simulation type with the sample time of $2 \times 10^{-6}$ s.

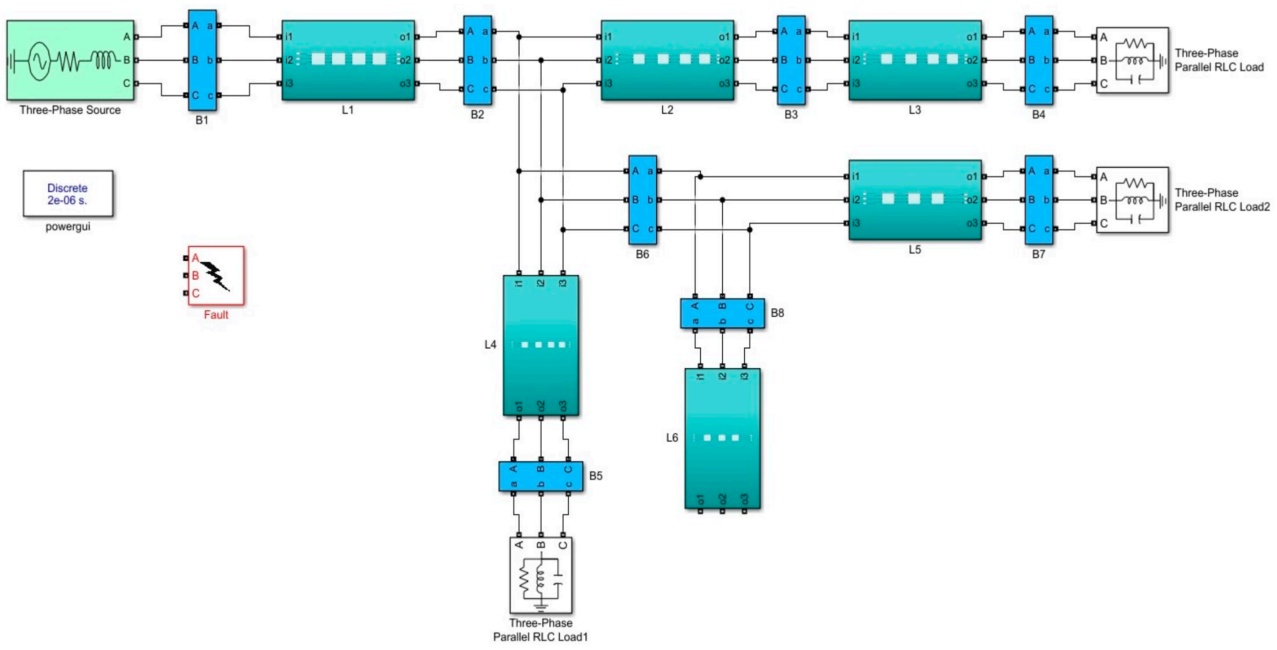

Three-Phase Source
B1
L1
B2
L2
B3
L3
B4
Three-Phase Parallel RLC Load
Discrete 2e-06 s.
powergui
Fault
B6
L5
B7
Three-Phase Parallel RLC Load2
L4
B8
L6
B5
Three-Phase Parallel RLC Load1

**Figure 2.** Distribution electrical system—Simulink model.

To introduce the fault block in different locations, each line is divided into three or four sectors of 1 km length, totaling 22 sectors (see Figure 3 and Table 1). From Figure 4, one can see that the sectors are modeled using three phase distributed parameter line blocks.

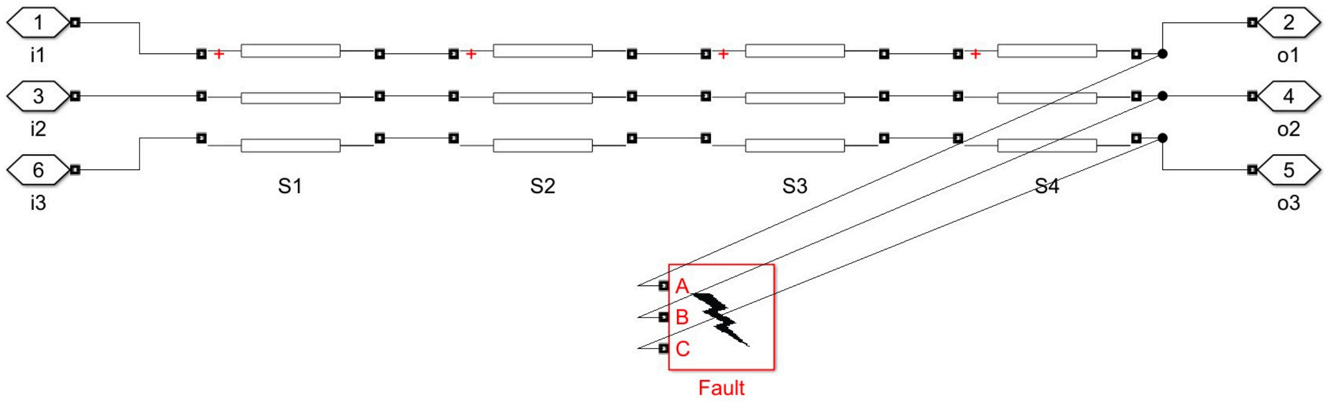
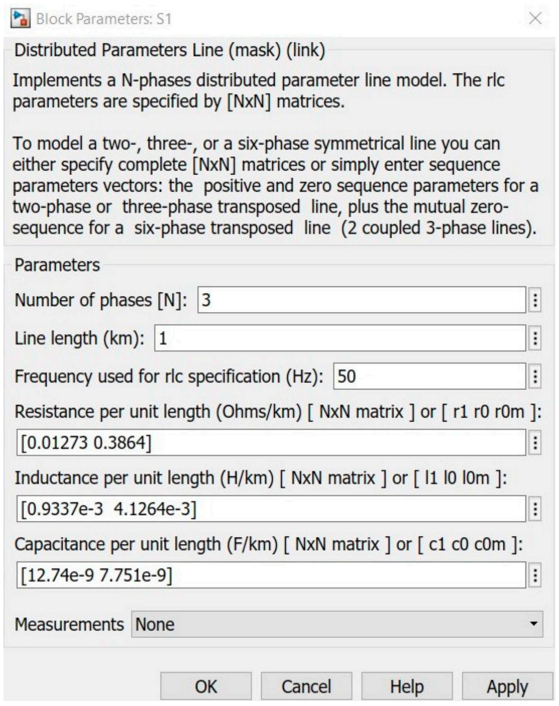


**Figure 3.** Fault block connected to L1—Sector 4.

**Table 1.** Subsystems of Lines.

| Line | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Number of Sectors | 4 | 4 | 4 | 4 | 3 | 3 |


Block Parameters: S1

Distributed Parameters Line (mask) (link)

Implements a N-phases distributed parameter line model. The rlc parameters are specified by [NxN] matrices.

To model a two-, three-, or a six-phase symmetrical line you can either specify complete [NxN] matrices or simply enter sequence parameters vectors: the positive and zero sequence parameters for a two-phase or three-phase transposed line, plus the mutual zero-sequence for a six-phase transposed line (2 coupled 3-phase lines).

Parameters

Number of phases [N]: 3

Line length (km): 1

Frequency used for rlc specification (Hz): 50

Resistance per unit length (Ohms/km) [ NxN matrix ] or [ r1 r0 r0m ]: [0.01273 0.3864]

Inductance per unit length (H/km) [ NxN matrix ] or [ l1 l0 l0m ]: [0.9337e-3 4.1264e-3]

Capacitance per unit length (F/km) [ NxN matrix ] or [ c1 c0 c0m ]: [12.74e-9 7.751e-9]

Measurements None

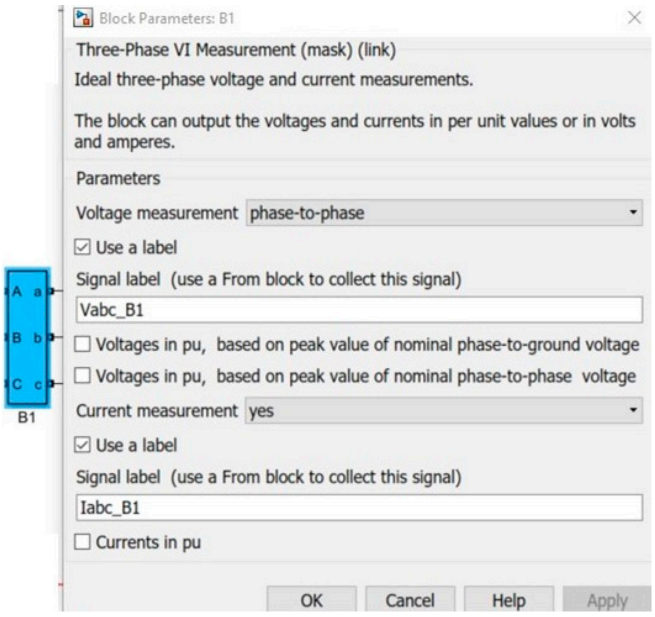
**Figure 4.** Distributed parameter line block.

The three-phase fault block can be set for twelve types of faults, the fault resistance, the ground resistances, and the no-fault situation included (see Table 2). The letters a, b, and c indicate the three power lines, while g indicates the ground plane.

**Table 2.** Types of faults.

| Fault | no | ag | bg | cg | ab | bc | ac | abg | bcg | acg | abc | abcg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

The three-phase voltage-intensity VI measurement blocks are mandatory to collect the values of the voltage–current pairs. As an example, Figure 5 presents the three-phase VI measurement block B1.



Block Parameters: B1

Three-Phase VI Measurement (mask) (link)

Ideal three-phase voltage and current measurements.

The block can output the voltages and currents in per unit values or in volts and amperes.

Parameters

Voltage measurement phase-to-phase

☑ Use a label

Signal label (use a From block to collect this signal) Vabc_B1

☐ Voltages in pu, based on peak value of nominal phase-to-ground voltage

☐ Voltages in pu, based on peak value of nominal phase-to-phase voltage

Current measurement yes

☑ Use a label

Signal label (use a From block to collect this signal) Iabc_B1

☐ Currents in pu

**Figure 5.** Three-phase VI measurement block—B1.

All measurements collected from the Simulink model are exported into the Matlab data acquisition workspace. Its workflow is described in the structure depicted in Figure 6a,b. For the sake of clarity, a cropped detail is presented in Figure 7.

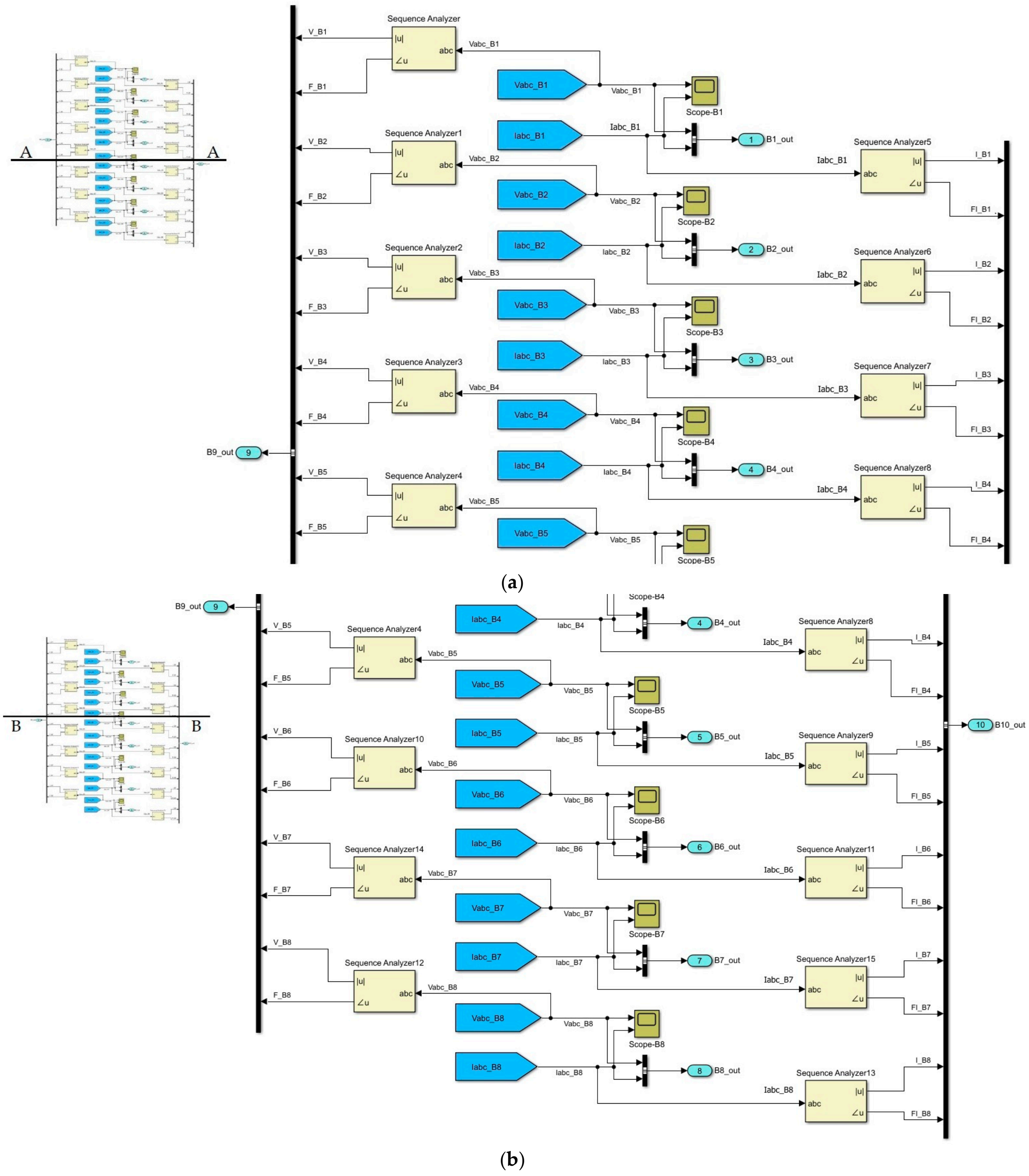


**Figure 6.** Data acquisition from Simulink model. (**a**) first half of the structure (A-A); (**b**) second half of the structure (B-B).

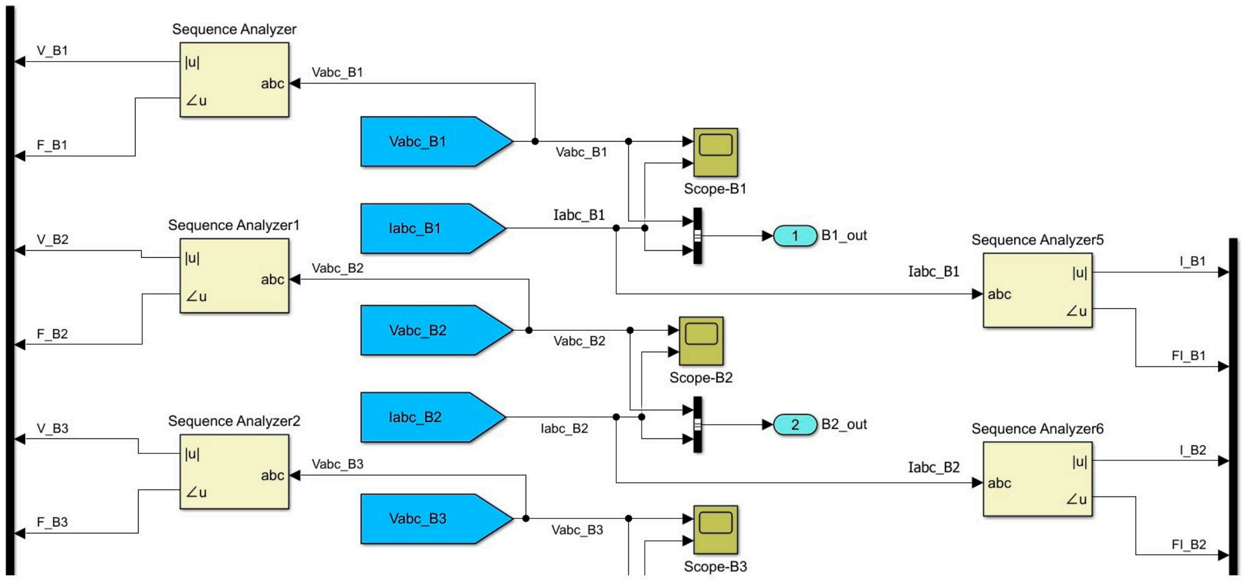


**Figure 7.** Data acquisition from Simulink model (cropped detail).

The voltage and current values are passed from the three-phase VI measurement blocks to the sequence analyzer, and then are exported to the Matlab variables. At each simulation, data are saved in a table which in the end will become the training dataset for the artificial intelligence algorithm.

## *3.2. Simulation Process*

After implementing the model of the distribution electrical system, the simulation process of different types of short-circuits may start for different values of fault and ground resistances in different locations of the system.

The fault block is moved along the system and is positioned at the end of each sector of all the six lines, which totals 22 positions. For each of these positions and for 25 situations of different chosen values for the fault and ground resistances, all the twelve types of faults are simulated. Performing all these simulations requires that for each simulation the location of the fault block, the type of fault, the values of fault or ground resistances be changed. The time needed for running a simulation is approximately two minutes, not considering the process of changing the parameters. This means that over the course of an hour, less than 30 simulations can be performed. Due to the large number of necessary simulations, and the time required to perform them, automation was almost mandatory.

The automation process has been achieved by implementing a Matlab program, which runs the Simulink model. Consequently, at each run, either the faults block position or the faults block parameters are automatically changed. Thus, the automation reduces the time of each simulation to 1.5 min./simulation, having benefits over the duration of the entire simulation process. Data from the measurement blocks are saved, and used later for the training algorithm of ANN.

The code of the program is presented in Appendix A.

Once the running process is completed, the 6150 simulations performed led to a dataset, which represents the input of the Classification Learner app. Examples of these data can be seen in Tables 3 and 4, which contain examples of the voltage–current values, provided by the eight measurement blocks along with the faults' location and type.

**Table 3.** Data from simulations—voltage measurements.

| V_B1 | F_B1 | V_B2 | F_B2 | V_B3 | F_B3 | V_B4 | F_B4 | V_B5 | F_B5 | V_B6 | F_B6 | V_B7 | F_B7 | V_B8 | F_B8 | Fault |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 27,055.78 | 21.49 | 15,580.69 | −10.52 | 15,569.99 | −11.19 | 15,561.13 | −11.86 | 15,571.83 | −11.19 | 15,580.69 | −10.52 | 15,574.22 | −11.02 | 15,580.69 | −10.52 | normal |
| 18,171.13 | 16.65 | 10,185.01 | −16.34 | 10,175.79 | −17.06 | 10,168.00 | −17.78 | 10,177.02 | −17.06 | 10,185.01 | −16.34 | 10,179.12 | −16.88 | 10,185.01 | −16.34 | L1/S1/ab |
| 21,772.26 | 17.81 | 11,901.15 | −16.32 | 11,884.51 | −17.03 | 11,869.45 | −17.74 | 11,885.95 | −17.03 | 11,901.15 | −16.32 | 11,889.87 | −16.85 | 11,901.15 | −16.32 | L1/S4/abg |
| 18,535.94 | 17.04 | 10,107.14 | −16.74 | 10,071.28 | −17.43 | 10,036.67 | −18.12 | 10,072.50 | −17.43 | 10,107.14 | −16.74 | 10,081.27 | −17.26 | 10,107.14 | −16.74 | L1/S2/abc |
| 18,049.38 | 17.43 | 9543.07 | −17.40 | 9507.63 | −18.09 | 9473.39 | −18.78 | 9508.79 | −18.09 | 9543.07 | −17.40 | 9517.47 | −17.92 | 9543.07 | −17.40 | L1/S3/abcg |
| 26,086.87 | 19.90 | 14,907.98 | −12.67 | 14,817.87 | −13.79 | 14,809.03 | −14.45 | 14,899.10 | −13.34 | 14,907.98 | −12.67 | 14,901.49 | −13.17 | 14,907.98 | −12.67 | L2/S3/ag |
| 22,056.05 | 22.85 | 12,037.82 | −9.73 | 11,419.66 | −11.16 | 11,385.26 | −11.86 | 12,007.42 | −10.43 | 12,037.82 | −9.73 | 12,015.14 | −10.25 | 12,037.82 | −9.73 | L2/S4/bc |
| 24,523.68 | 20.02 | 13,801.96 | −12.93 | 13,711.52 | −13.84 | 13,690.30 | −14.46 | 13,781.46 | −13.55 | 13,801.96 | −12.93 | 13,786.77 | −13.39 | 13,801.96 | −12.93 | L2/S1/ac |
| 23,704.60 | 19.76 | 13,229.93 | −13.43 | 13,025.51 | −14.78 | 13,000.68 | −15.41 | 13,206.50 | −14.06 | 13,229.93 | −13.43 | 13,212.52 | −13.91 | 13,229.93 | −13.43 | L2/S2/acg |
| 24,365.01 | 21.04 | 13,688.99 | −11.52 | 13,357.55 | −12.78 | 13,186.54 | −13.79 | 13,674.71 | −12.22 | 13,688.99 | −11.52 | 13,678.42 | −12.04 | 13,688.99 | −11.52 | L3/S2/bg |
| 25,762.93 | 21.45 | 14,670.19 | −10.81 | 14,515.01 | −11.80 | 14,363.95 | −12.80 | 14,653.02 | −11.47 | 14,670.19 | −10.81 | 14,657.48 | −11.31 | 14,670.19 | −10.81 | L3/S4/cg |
| 21,336.88 | 22.09 | 11,538.24 | −10.91 | 10,839.79 | −12.85 | 10,642.02 | −13.90 | 11,509.08 | −11.60 | 11,538.24 | −10.91 | 11,516.48 | −11.43 | 11,538.24 | −10.91 | L3/S1/bcg |
| 19,859.98 | 19.21 | 10,529.43 | −15.11 | 10,520.14 | −15.83 | 10,512.29 | −16.54 | 10,087.15 | −17.18 | 10,529.43 | −15.11 | 10,523.51 | −15.65 | 10,529.43 | −15.11 | L4/S2/ab |
| 20,159.75 | 19.86 | 10,725.32 | −14.26 | 10,693.38 | −14.93 | 10,662.72 | −15.61 | 9922.10 | −17.64 | 10,725.32 | −14.26 | 10,702.41 | −14.77 | 10,725.32 | −14.26 | L4/S4/abc |
| 23,886.24 | 20.15 | 13,352.04 | −12.90 | 13,176.45 | −13.44 | 13,151.85 | −14.07 | 13,328.82 | −13.52 | 13,352.04 | −12.90 | 13,229.38 | −14.42 | 13,352.04 | −12.90 | L5/S3/ac |
| 19,908.07 | 18.69 | 10,574.29 | −15.89 | 10,382.83 | −17.06 | 10,352.96 | −17.75 | 10,545.93 | −16.58 | 10,574.29 | −15.89 | 10,267.29 | −17.60 | 10,574.29 | −15.89 | L5/S2/abg |
| 18,752.83 | 18.50 | 9767.32 | −16.67 | 9722.06 | −17.48 | 9688.60 | −18.17 | 9734.01 | −17.36 | 9767.32 | −16.67 | 9518.96 | −18.10 | 9767.32 | −16.67 | L5/S1/abcg |
| 25,953.14 | 19.64 | 14,816.10 | −13.01 | 14,805.45 | −13.68 | 14,796.51 | −14.34 | 14,807.22 | −13.68 | 14,816.10 | −13.01 | 14,809.61 | −13.51 | 14,816.10 | −13.01 | L6/S1/ag |
| 26,087.22 | 18.75 | 14,955.34 | −14.04 | 14,943.03 | −14.74 | 14,932.66 | −15.44 | 14,944.86 | −14.74 | 14,955.34 | −14.04 | 14,947.64 | −14.56 | 14,955.34 | −14.04 | L6/S3/ab |
| 20,906.62 | 21.96 | 11,237.47 | −11.20 | 11,205.23 | −11.89 | 11,174.43 | −12.59 | 11,206.53 | −11.89 | 11,237.47 | −11.20 | 11,214.37 | −11.72 | 11,237.47 | −11.20 | L6/S3/bcg |

**Table 4.** Data from simulations—current measurements.

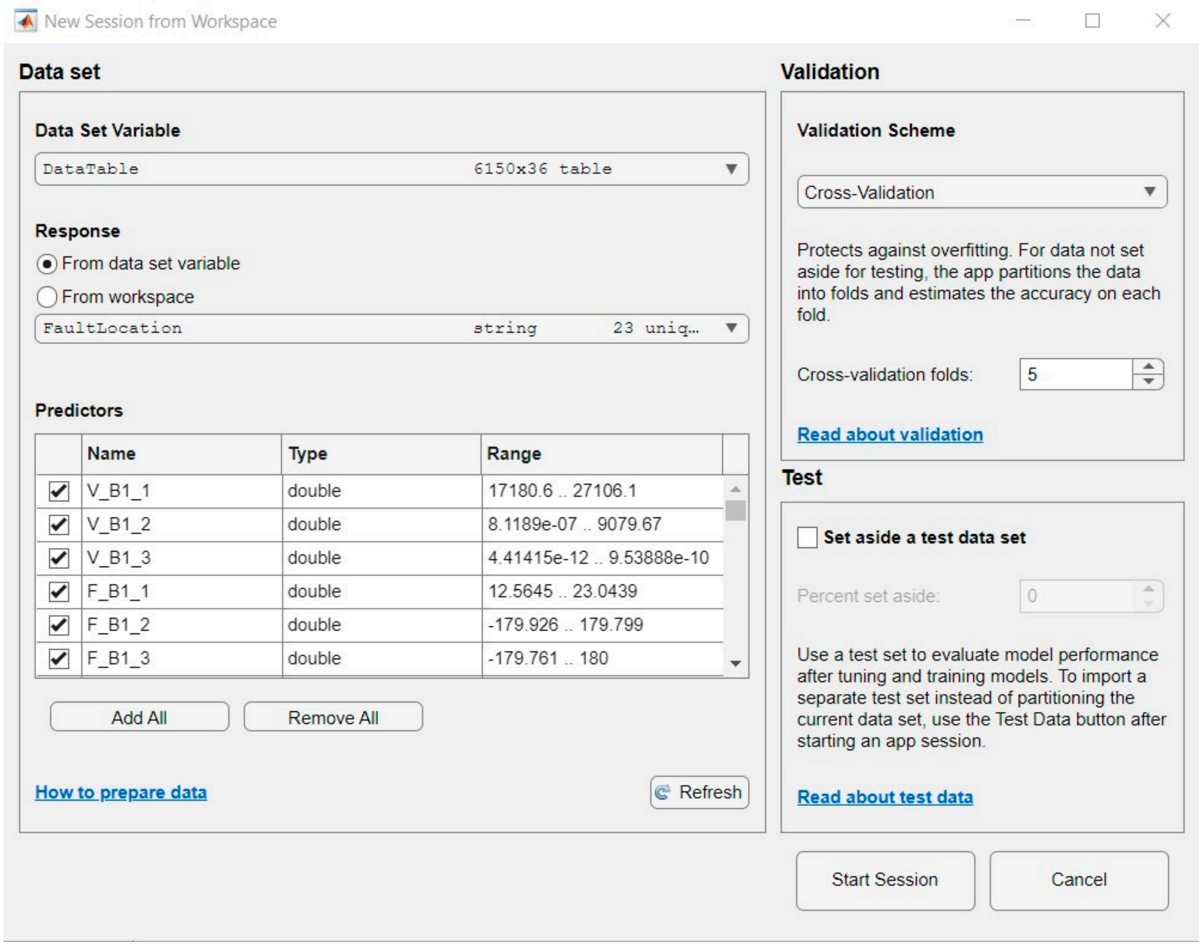
| I_B1 | FI_B1 | I_B2 | FI_B2 | I_B3 | FI_B3 | I_B4 | FI_B4 | I_B5 | FI_B5 | I_B6 | FI_B6 | I_B7 | FI_B7 | I_B8 | FI_B8 | Fault |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 467.06 | −11.19 | 467.06 | −11.22 | 155.61 | −11.77 | 155.61 | −11.86 | 155.72 | −11.19 | 155.74 | −10.88 | 155.74 | −11.02 | 0.19 | 79.48 | normal |
| 986.80 | −71.99 | 306.09 | −17.09 | 101.96 | −17.68 | 101.96 | −17.77 | 102.05 | −17.05 | 102.08 | −16.72 | 102.07 | −16.87 | 0.13 | 72.64 | L1/S1/ab |
| 683.43 | −63.32 | 357.35 | −17.05 | 118.99 | −17.63 | 118.98 | −17.72 | 119.15 | −17.02 | 119.20 | −16.69 | 119.19 | −16.84 | 0.15 | 70.38 | L1/S4/abg |
| 1130.02 | −77.99 | 303.45 | −17.46 | 100.87 | −18.02 | 100.83 | −18.11 | 101.19 | −17.43 | 101.33 | −17.11 | 101.27 | −17.25 | 0.13 | 59.89 | L1/S2/abc |
| 1174.33 | −78.30 | 286.58 | −18.11 | 95.25 | −18.68 | 95.21 | −18.77 | 95.56 | −18.08 | 95.71 | −17.77 | 95.65 | −17.91 | 0.12 | 58.55 | L1/S3/abcg |
| 434.18 | −31.17 | 434.26 | −31.20 | 148.20 | −14.36 | 148.20 | −14.45 | 149.09 | −13.33 | 149.11 | −13.03 | 149.11 | −13.17 | 0.18 | 77.20 | L2/S3/ag |
| 755.49 | −56.98 | 755.62 | −56.99 | 114.14 | −11.77 | 114.10 | −11.87 | 120.29 | −10.44 | 120.41 | −10.13 | 120.36 | −10.27 | 0.15 | 71.03 | L2/S4/bc |
| 635.90 | −62.07 | 636.05 | −62.08 | 137.14 | −14.38 | 137.13 | −14.46 | 138.03 | −13.55 | 138.10 | −13.27 | 138.08 | −13.40 | 0.15 | 72.26 | L2/S1/ac |
| 686.53 | −64.04 | 686.68 | −64.05 | 130.30 | −15.32 | 130.28 | −15.41 | 132.32 | −14.07 | 132.41 | −13.78 | 132.38 | −13.91 | 0.15 | 70.37 | L2/S2/acg |
| 552.69 | −36.89 | 552.79 | −36.91 | 325.72 | −57.67 | 131.99 | −13.79 | 136.85 | −12.22 | 136.90 | −11.90 | 136.89 | −12.04 | 0.17 | 76.22 | L3/S2/bg |
| 529.59 | −33.87 | 529.67 | −33.89 | 282.59 | −57.47 | 143.75 | −12.80 | 146.61 | −11.47 | 146.67 | −11.17 | 146.66 | −11.31 | 0.17 | 76.21 | L3/S4/cg |
| 816.81 | −61.74 | 816.95 | −61.75 | 692.95 | −76.62 | 106.72 | −13.90 | 115.34 | −11.62 | 115.46 | −11.30 | 115.41 | −11.44 | 0.14 | 69.85 | L3/S1/bcg |
| 827.00 | −69.65 | 827.15 | −69.65 | 105.38 | −16.44 | 105.38 | −16.54 | 101.14 | −17.17 | 105.49 | −15.50 | 105.49 | −15.65 | 0.13 | 73.96 | L4/S2/ab |
| 920.05 | −75.40 | 920.19 | −75.40 | 107.05 | −15.53 | 107.02 | −15.62 | 99.68 | −17.63 | 107.46 | −14.64 | 107.41 | −14.78 | 0.13 | 64.71 | L4/S4/abc |
| 673.96 | −63.11 | 674.11 | −63.13 | 131.79 | −13.99 | 131.76 | −14.08 | 133.52 | −13.53 | 393.81 | −76.21 | 132.55 | −14.41 | 0.15 | 71.04 | L5/S3/ac |
| 934.87 | −73.82 | 935.01 | −73.83 | 103.97 | −17.65 | 103.94 | −17.75 | 105.85 | −16.58 | 524.97 | −82.92 | 103.09 | −17.60 | 0.13 | 64.11 | L5/S2/abg |
| 1086.03 | −77.37 | 1086.16 | −77.38 | 97.38 | −18.07 | 97.34 | −18.17 | 97.79 | −17.36 | 964.12 | −87.53 | 95.66 | −18.08 | 0.12 | 59.99 | L5/S1/abcg |
| 434.45 | −34.11 | 434.54 | −34.13 | 148.07 | −14.25 | 148.07 | −14.34 | 148.18 | −13.68 | 187.24 | −67.36 | 148.20 | −13.51 | 156.01 | −117.51 | L6/S1/ag |
| 614.49 | −18.20 | 614.51 | −18.22 | 149.37 | −15.32 | 149.37 | −15.42 | 149.49 | −14.72 | 316.61 | −21.34 | 149.52 | −14.54 | 169.11 | −27.41 | L6/S3/ab |
| 862.27 | −63.42 | 862.41 | −63.43 | 112.05 | −12.51 | 112.01 | −12.60 | 112.34 | −11.91 | 743.09 | −77.07 | 112.41 | −11.73 | 703.76 | −85.42 | L6/S3/bcg |

Through the sequence analyzer from the data acquisition of the Simulink model, the magnitude and phase angle of the three-phase signals are obtained. For instance, measurement block B1 provides values of the voltage magnitude V_B1, the voltage phase angle F_B1, the current magnitude I_B1 and the current phase angle FI_B1 (see also Figure 6a,b).

### 3.3. Classification Learner App

Once the simulation process is completed, the large amount of obtained data is used in the Classification Learner App from Matlab to train the artificial intelligence (AI) algorithms. This application can classify data based on the training dataset and return a single response for a further situation [30].

To start the training session, it is necessary to set the parameters observed in Figure 8. The table named "DataTable", containing the results of 6150 simulations, becomes the dataset variable in the Classification Learner App. Data from this table are divided into two types of data, namely predictors and response. Predictors are represented by the values of voltage–current pairs measured at each simulation and the response represents the fault location of the exposed situation. The validation scheme was set as a cross-validation with five folds. After setting these parameters, the session starts by clicking the Start Session button.


New Session from Workspace

**Data set**

**Data Set Variable**

DataTable 6150x36 table

**Response**

(●) From data set variable

( ) From workspace

FaultLocation string 23 uniq…

**Predictors**

| | Name | Type | Range |
|---|---|---|---|
| ✓ | V_B1_1 | double | 17180.6 .. 27106.1 |
| ✓ | V_B1_2 | double | 8.1189e-07 .. 9079.67 |
| ✓ | V_B1_3 | double | 4.41415e-12 .. 9.53888e-10 |
| ✓ | F_B1_1 | double | 12.5645 .. 23.0439 |
| ✓ | F_B1_2 | double | -179.926 .. 179.799 |
| ✓ | F_B1_3 | double | -179.761 .. 180 |

**Validation**

**Validation Scheme**

Cross-Validation

Protects against overfitting. For data not set aside for testing, the app partitions the data into folds and estimates the accuracy on each fold.

Cross-validation folds: 5

**Test**

☐ **Set aside a test data set**

Percent set aside: 0

Use a test set to evaluate model performance after tuning and training models. To import a separate test set instead of partitioning the current data set, use the Test Data button after starting an app session.

**Figure 8.** Classification Learner app—training dataset.

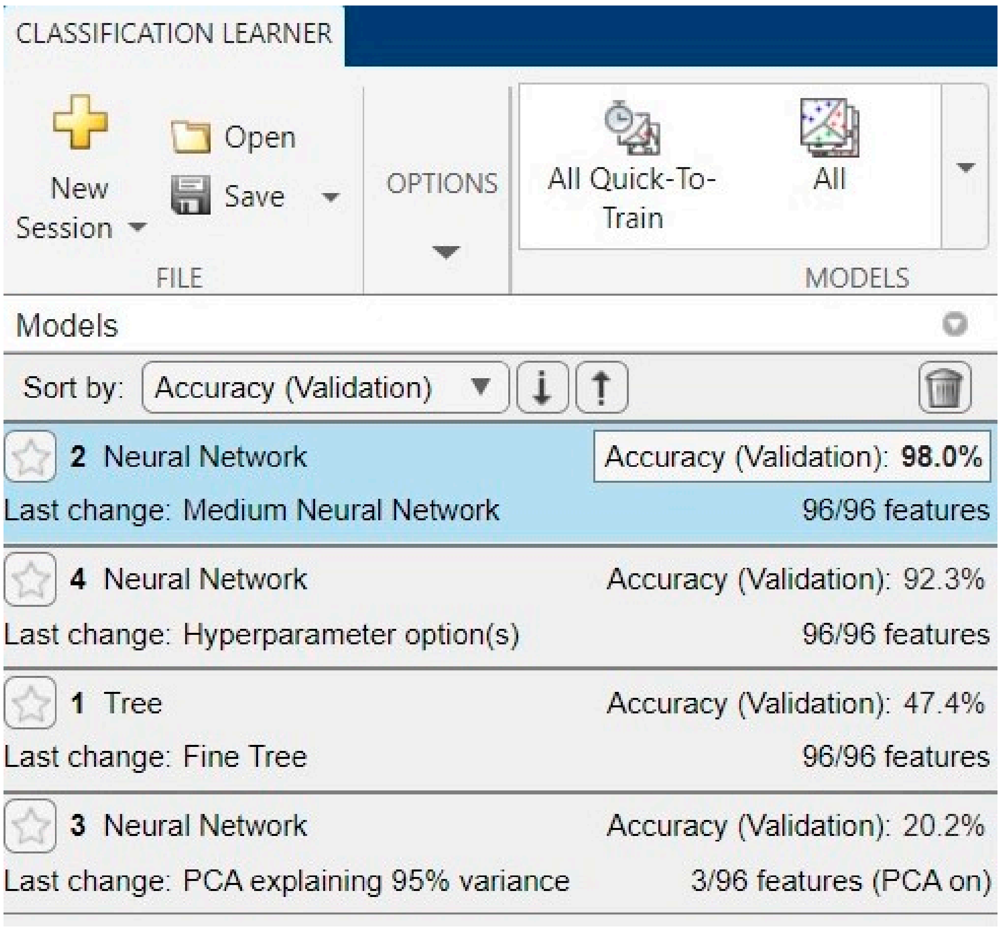
Simulations have been also performed for the cases in which the response is both the location and the type of the fault.

In the Classification Learner app, the model of the training algorithm can be set. To obtain the best accuracy validation values, the option of training all models algorithm may also be chosen.

After the analysis of several training models, Figure 9 reveals that the most efficient model sorted in terms of accuracy is the medium neural network algorithm. One can see that the accuracy validation in the case of fault location was 98% and 94.7% in the case of both location and type of the fault.

If the response is only the fault location (example L1/S1 for line 1/sector 1), which implied 23 unique answers, the validation accuracy will be better than in the case where the response is both the location and the type of the fault (example L1/S1/ab for line1/sector 1/type of fault ab) which implied 243 unique answers. These two cases will be presented comparatively in the next stage, where the fault location and the fault location and type based on the trained model are predicted.

In evaluating and observing the performance of a trained model, the analysis of the confusion matrix and of the receiver operating characteristic (ROC) curve are two useful tools.



Accuracy (Validation)

| | |
|---|---|
| 2 Neural Network | Accuracy (Validation): **98.0%** |
| Last change: Medium Neural Network | 96/96 features |
| 4 Neural Network | Accuracy (Validation): 92.3% |
| Last change: Hyperparameter option(s) | 96/96 features |
| 1 Tree | Accuracy (Validation): 47.4% |
| Last change: Fine Tree | 96/96 features |
| 3 Neural Network | Accuracy (Validation): 20.2% |
| Last change: PCA explaining 95% variance | 3/96 features (PCA on) |

**Figure 9.** Classification Learner app.

The confusion matrix contains the predicted classes in its columns and the true classes in its rows; therefore, a 100% validation accuracy assumes a perfect principal diagonal confusion matrix. Values situated outside the principal diagonal of the matrix indicate situations that are not well-predicted and need supplementary data training set [30].

In Figures 10 and 11, the validation confusion matrices for the two studied cases are presented. From Figure 10, which presents the case of 23 unique responses, one can observed that the only situation with 100% accuracy is the no-fault (normal) situation. In Figure 11, one can also observe the shape of the principal diagonal in the case of 243 unique responses. Due to the complexity of the simulation, the values situated outside the principal diagonal are not visible in the resolution of Figure 11. The 243 classes containing the location and type of faults are presented in Table 5.

From the confusion matrix depicted in Figure 10, it can be observed that for lines 5 and 6 of the Simulink model, the prediction response is poorer, which is noticeable from the larger number of values that deviate from the principal diagonal. For the other lines, the situation is better, with fewer cases where the predicted class does not coincide with the true class.

Even though some values are far from the principal diagonal, their low values indicate that the probability of a wrong prediction is unlikely. For instance, for the predicted class L4/S4, there is a single situation in which the true class is in fact L1/S2.

Unfortunately, the confusion matrix for the case with 243 unique responses is not useful to indicate the classes that were not well-predicted. This difficulty can be alleviate using other tools provided by the Classification Learner app, one of them being the ROC curve, an efficient way of comparing trained models.

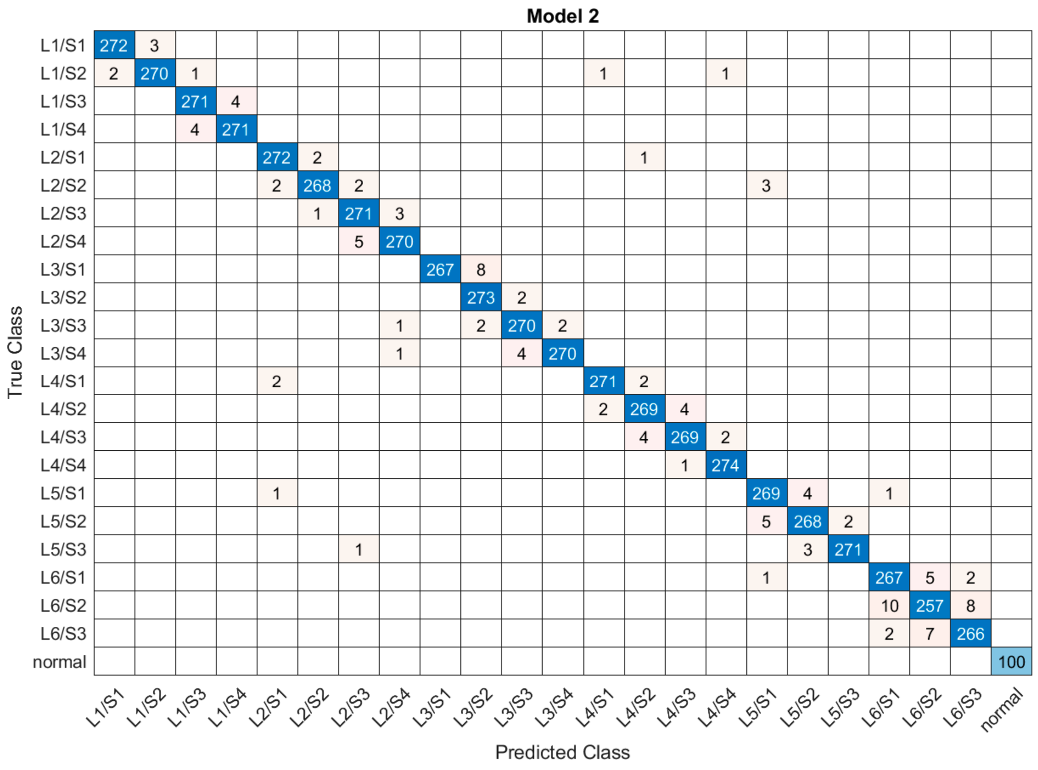


**Figure 10.** Validation confusion matrix for medium neural network model—accuracy 98.0%—for location of the fault.

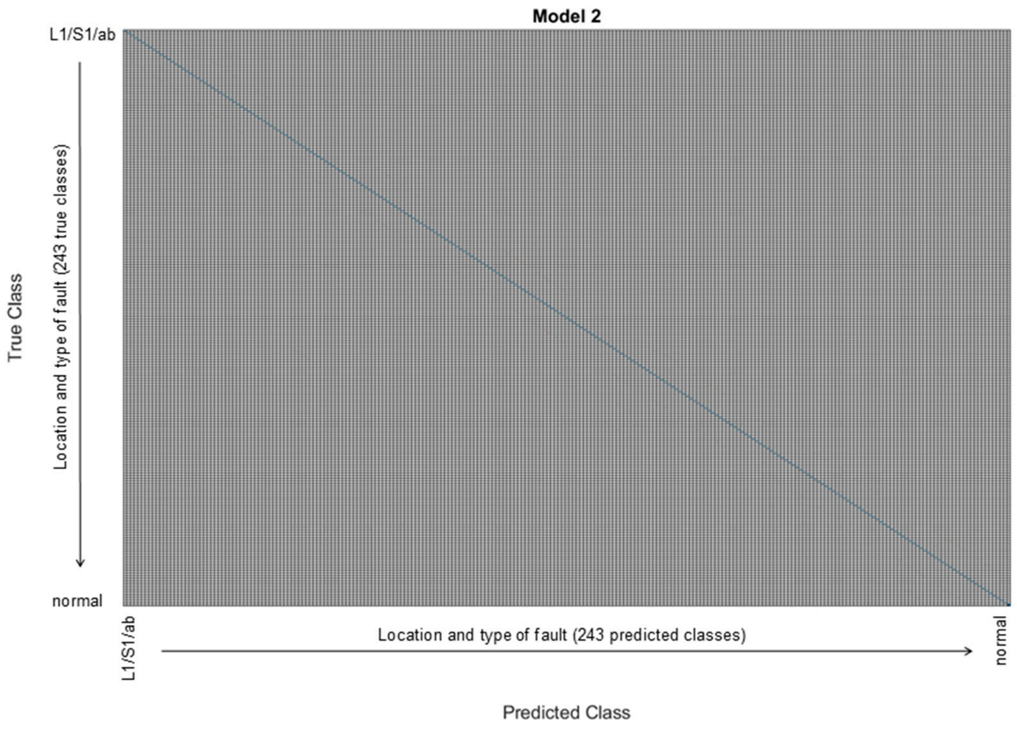


**Figure 11.** Validation confusion matrix for medium neural network model—accuracy 94.7%—for location and type of fault.

**Table 5.** Location and type of faults.

| **Location and Type of Faults** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| L1/S1/ab | L1/S3/ab | L2/S1/ab | L2/S3/ab | L3/S1/ab | L3/S3/ab | L4/S1/ab | L4/S3/ab | L5/S1/ab | L5/S3/ab | L6/S2/ab |
| L1/S1/abc | L1/S3/abc | L2/S1/abc | L2/S3/abc | L3/S1/abc | L3/S3/abc | L4/S1/abc | L4/S3/abc | L5/S1/abc | L5/S3/abc | L6/S2/abc |
| L1/S1/abcg | L1/S3/abcg | L2/S1/abcg | L2/S3/abcg | L3/S1/abcg | L3/S3/abcg | L4/S1/abcg | L4/S3/abcg | L5/S1/abcg | L5/S3/abcg | L6/S2/abcg |
| L1/S1/abg | L1/S3/abg | L2/S1/abg | L2/S3/abg | L3/S1/abg | L3/S3/abg | L4/S1/abg | L4/S3/abg | L5/S1/abg | L5/S3/abg | L6/S2/abg |
| L1/S1/ac | L1/S3/ac | L2/S1/ac | L2/S3/ac | L3/S1/ac | L3/S3/ac | L4/S1/ac | L4/S3/ac | L5/S1/ac | L5/S3/ac | L6/S2/ac |
| L1/S1/acg | L1/S3/acg | L2/S1/acg | L2/S3/acg | L3/S1/acg | L3/S3/acg | L4/S1/acg | L4/S3/acg | L5/S1/acg | L5/S3/acg | L6/S2/acg |
| L1/S1/ag | L1/S3/ag | L2/S1/ag | L2/S3/ag | L3/S1/ag | L3/S3/ag | L4/S1/ag | L4/S3/ag | L5/S1/ag | L5/S3/ag | L6/S2/ag |
| L1/S1/bc | L1/S3/bc | L2/S1/bc | L2/S3/bc | L3/S1/bc | L3/S3/bc | L4/S1/bc | L4/S3/bc | L5/S1/bc | L5/S3/bc | L6/S2/bc |
| L1/S1/bcg | L1/S3/bcg | L2/S1/bcg | L2/S3/bcg | L3/S1/bcg | L3/S3/bcg | L4/S1/bcg | L4/S3/bcg | L5/S1/bcg | L5/S3/bcg | L6/S2/bcg |
| L1/S1/bg | L1/S3/bg | L2/S1/bg | L2/S3/bg | L3/S1/bg | L3/S3/bg | L4/S1/bg | L4/S3/bg | L5/S1/bg | L5/S3/bg | L6/S2/bg |
| L1/S1/cg | L1/S3/cg | L2/S1/cg | L2/S3/cg | L3/S1/cg | L3/S3/cg | L4/S1/cg | L4/S3/cg | L5/S1/cg | L5/S3/cg | L6/S2/cg |
| L1/S2/ab | L1/S4/ab | L2/S2/ab | L2/S4/ab | L3/S2/ab | L3/S4/ab | L4/S2/ab | L4/S4/ab | L5/S2/ab | L6/S1/ab | L6/S3/ab |
| L1/S2/abc | L1/S4/abc | L2/S2/abc | L2/S4/abc | L3/S2/abc | L3/S4/abc | L4/S2/abc | L4/S4/abc | L5/S2/abc | L6/S1/abc | L6/S3/abc |
| L1/S2/abcg | L1/S4/abcg | L2/S2/abcg | L2/S4/abcg | L3/S2/abcg | L3/S4/abcg | L4/S2/abcg | L4/S4/abcg | L5/S2/abcg | L6/S1/abcg | L6/S3/abcg |
| L1/S2/abg | L1/S4/abg | L2/S2/abg | L2/S4/abg | L3/S2/abg | L3/S4/abg | L4/S2/abg | L4/S4/abg | L5/S2/abg | L6/S1/abg | L6/S3/abg |
| L1/S2/ac | L1/S4/ac | L2/S2/ac | L2/S4/ac | L3/S2/ac | L3/S4/ac | L4/S2/ac | L4/S4/ac | L5/S2/ac | L6/S1/ac | L6/S3/ac |
| L1/S2/acg | L1/S4/acg | L2/S2/acg | L2/S4/acg | L3/S2/acg | L3/S4/acg | L4/S2/acg | L4/S4/acg | L5/S2/acg | L6/S1/acg | L6/S3/acg |
| L1/S2/ag | L1/S4/ag | L2/S2/ag | L2/S4/ag | L3/S2/ag | L3/S4/ag | L4/S2/ag | L4/S4/ag | L5/S2/ag | L6/S1/ag | L6/S3/ag |
| L1/S2/bc | L1/S4/bc | L2/S2/bc | L2/S4/bc | L3/S2/bc | L3/S4/bc | L4/S2/bc | L4/S4/bc | L5/S2/bc | L6/S1/bc | L6/S3/bc |
| L1/S2/bcg | L1/S4/bcg | L2/S2/bcg | L2/S4/bcg | L3/S2/bcg | L3/S4/bcg | L4/S2/bcg | L4/S4/bcg | L5/S2/bcg | L6/S1/bcg | L6/S3/bcg |
| L1/S2/bg | L1/S4/bg | L2/S2/bg | L2/S4/bg | L3/S2/bg | L3/S4/bg | L4/S2/bg | L4/S4/bg | L5/S2/bg | L6/S1/bg | L6/S3/bg |
| L1/S2/cg | L1/S4/cg | L2/S2/cg | L2/S4/cg | L3/S2/cg | L3/S4/cg | L4/S2/cg | L4/S4/cg | L5/S2/cg | L6/S1/cg | L6/S3/cg |
| | | | | | | | | | | normal |

The ROC curve is a plot tool that provides the false positive rate and the true positive rate of each predicted class. The area under curve (AUC) is an indicator of the quality of the classifier. The AUC values range between 0 and 1, with a higher value indicating a better performance of the classifier [30].

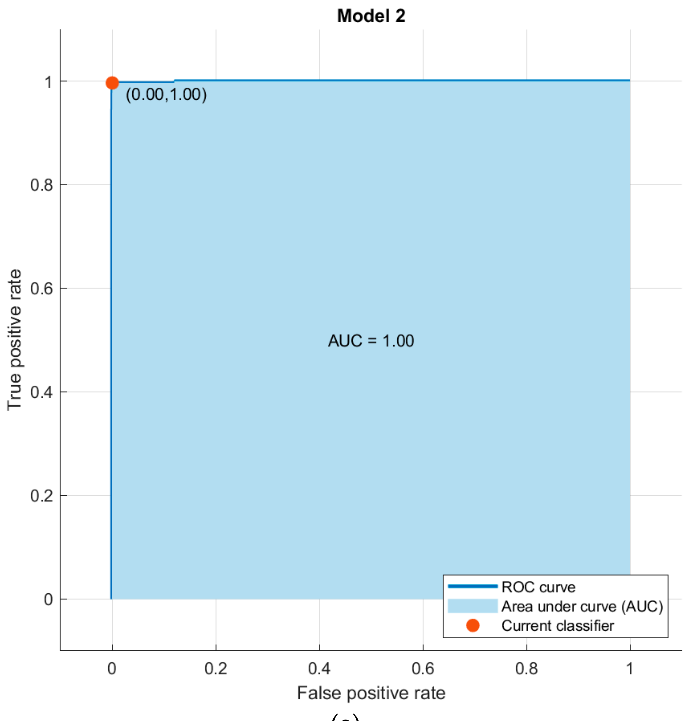
In Figures 12 and 13, the ROC curves for some representative cases studied are presented. Figure 12 presents the ROC curve for the medium neural network model with an accuracy validation of 98.0% and Figure 13 presents the ROC curve for the medium neural network model with an accuracy validation of 94.7%. In both figures, the highest and the lowest values of AUC are shown.

For the model which predicts only the location of the fault, has an accuracy validation of 98.0%, and encompasses 23 situations, the maximum value of AUC (1.00) is reached for several classifiers. Figure 12a presents the ROC curve in one of these situations, while Figure 12b presents the lower value of AUC, which, in this case, is of 0.99 and occurs for Line 6-Sector 2.

For the model which predicts both location and type of the fault, has an accuracy validation of 94.7%, and encompasses 243 situations, the maximum value of AUC (1.00) is also reached for several classifiers. Figure 13a presents the ROC curve in one of these situations, while Figure 13b presents the lower value of AUC, which, in this case, is of 0.94 and also occurs for Line 6-Sector 2.

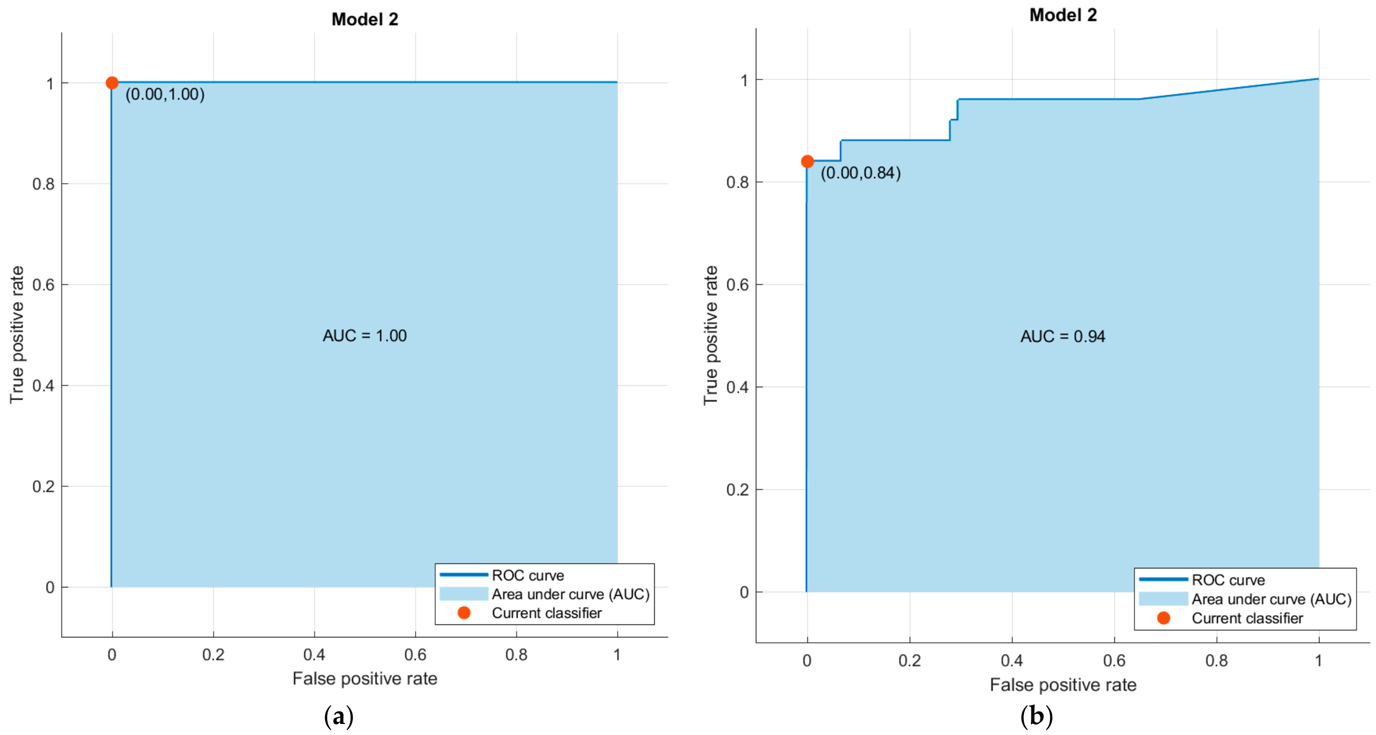
Both situations reveal a good accuracy validation prediction.

By analyzing the ROC curves and comparing their results with the ones of the confusion matrices, it is obvious that the two plotting tools offer the same results but in different ways. Since the confusion matrix presents an overview of all classes, the ROC curve presents specific results for each class.



(**a**) (**b**)

**Figure 12.** ROC curve for medium neural network model—accuracy 98.0% (**a**) the AUC 1.00 for Line 4-Sector 4, (**b**) the AUC 0.99 for Line 6-Sector 2.



(**a**) (**b**)

**Figure 13.** ROC curve for medium neural network model accuracy—94.7%—(**a**) the AUC 1.00 for Line 4-Sector 4, (**b**) the AUC 0.94 for Line 6-Sector 2.

## *3.4. Prediction of Faults Based on the Trained Model*

After analyzing the different types of trained models, these can be exported from the Classification Learner app to the Matlab workspace as a new variable (“trainedModel”). This variable can be used to predict responses for other faults which may occur in the same distribution system, and which have not been considered in the previous training dataset. As an example, Tables 6 and 7 contain four situations in which different fault and ground resistances have been used.

**Table 6.** Data for predictions—voltage measurements.

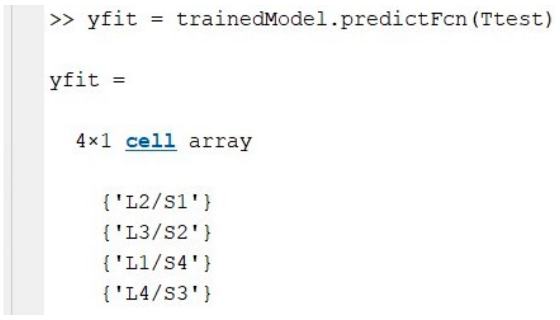
| V_B1 | F_B1 | V_B2 | F_B2 | V_B3 | F_B3 | V_B4 | F_B4 | V_B5 | F_B5 | V_B6 | F_B6 | V_B7 | F_B7 | V_B8 | F_B8 | Fault |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 23,387.39 | 20.70 | 13,006.09 | −12.14 | 12,881.25 | −13.06 | 12,864.69 | −13.77 | 12,989.73 | −12.85 | 13,006.09 | −12.14 | 12,993.94 | −12.67 | 13,006.09 | −12.14 | L2/S1/bg |
| 20,579.33 | 20.40 | 11,010.63 | −13.41 | 10,238.55 | −16.38 | 9846.89 | −18.38 | 10,981.53 | −14.08 | 11,010.63 | −13.41 | 10,988.92 | −13.91 | 11,010.63 | −13.41 | L3/S2/abc |
| 25,874.47 | 19.42 | 14,763.31 | −13.30 | 14,752.70 | −13.97 | 14,743.79 | −14.63 | 14,754.46 | −13.97 | 14,763.31 | −13.30 | 14,756.85 | −13.80 | 14,763.31 | −13.30 | L1/S4/ag |
| 19,363.64 | 19.39 | 10,176.50 | −15.19 | 10,143.97 | −15.87 | 10,112.66 | −16.55 | 9503.63 | −18.32 | 10,176.50 | −15.19 | 10,153.10 | −15.70 | 10,176.50 | −15.19 | L4/S3/abcg |

**Table 7.** Data for predictions—current measurements.

| I_B1 | FI_B1 | I_B2 | FI_B2 | I_B3 | FI_B3 | I_B4 | FI_B4 | I_B5 | FI_B5 | I_B6 | FI_B6 | I_B7 | FI_B7 | I_B8 | FI_B8 | Fault |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 23,387.39 | 20.70 | 13,006.09 | −12.14 | 12,881.25 | −13.06 | 12,864.69 | −13.77 | 12,989.73 | −12.85 | 13,006.09 | −12.14 | 12,993.94 | −12.67 | 13,006.09 | −12.14 | L2/S1/bg |
| 20,579.33 | 20.40 | 11,010.63 | −13.41 | 10,238.55 | −16.38 | 9846.89 | −18.38 | 10,981.53 | −14.08 | 11,010.63 | −13.41 | 10,988.92 | −13.91 | 11,010.63 | −13.41 | L3/S2/abc |
| 25,874.47 | 19.42 | 14,763.31 | −13.30 | 14,752.70 | −13.97 | 14,743.79 | −14.63 | 14,754.46 | −13.97 | 14,763.31 | −13.30 | 14,756.85 | −13.80 | 14,763.31 | −13.30 | L1/S4/ag |
| 19,363.64 | 19.39 | 10,176.50 | −15.19 | 10,143.97 | −15.87 | 10,112.66 | −16.55 | 9503.63 | −18.32 | 10,176.50 | −15.19 | 10,153.10 | −15.70 | 10,176.50 | −15.19 | L4/S3/abcg |

Measurements generated for the predictions must be introduced in a variable ("Ttest") with the same structure as ("DataTable") as the variable used in the training dataset. After processing the variable to be tested, using the prediction function for the trained model, the response based on the trained model will be obtained.

Figures 14 and 15 present the responses of the two cases studied, e.g., for the medium neural network model with an accuracy of 98.0% and for the medium neural network model with an accuracy of 94.7%.



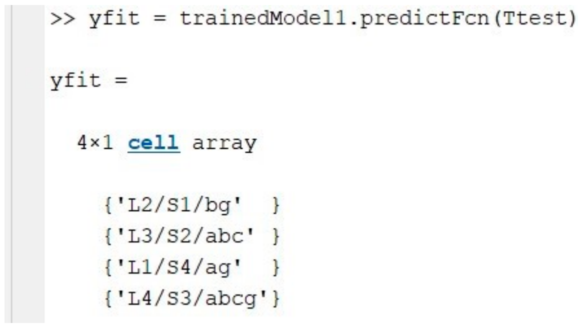
```
>> yfit = trainedModel.predictFcn(Ttest)

yfit =

  4×1 cell array

    {'L2/S1'}
    {'L3/S2'}
    {'L1/S4'}
    {'L4/S3'}
```

**Figure 14.** Predicted results for test examples (for medium neural network model accuracy 98.0%—for the fault location).



```
>> yfit = trainedModel1.predictFcn(Ttest)

yfit =

  4×1 cell array

    {'L2/S1/bg'  }
    {'L3/S2/abc' }
    {'L1/S4/ag'  }
    {'L4/S3/abcg'}
```

**Figure 15.** Predicted results for test examples (for medium neural network model accuracy 94.7%—for location and type of the fault).

By applying the prediction function to Tables 6 and 7, which contain the voltage and current measurement data, two cell arrays are created, namely two column vectors (see Figures 14 and 15).

It can be observed that in the first case, the response contains only the location of the fault, and in the second case, the response includes in addition the type of the fault. By comparing the result of the prediction function with the last column ("Fault") of the ("Ttest") table, one can observe that the trained model operates properly.

## 4. Discussion

The paper presents a solution for detecting and locating faults in electrical distribution systems using a Simulink model combined with the ANN algorithms of the Classification Learner app provided by Matlab.

The high performance of the proposed technique emphasizes the potential of using this principle for real-world distribution electrical systems.

After the first stage devoted to the development of the simulation model and the simulations presented in the second stage, in the third stage, it can be observed that the performance of the trained model is correlated to the training dataset. The larger the database is, the higher the performance of the trained model is. In order to have a good accuracy validation value, the number of simulations must be adapted to the complexity of the analyzed system.

Simulating all the cases which can offer a solid test set for the presented case study implies the use of high-level hardware and software resources. These needs can be covered by using local hardware resources; however, much better results can be obtained by using a virtual machine which can significantly shorten the required simulations time. If several virtual machines are simultaneously used, the simulation process can be accelerated, and in the end, the data collected from all of them can be processed and integrated into a single database.

When applying the method presented above for a real-world distribution electrical system, it is mandatory that the Simulink model must contain its very real components. This method can also be used in other situations, for instance, in industrial estates, where the consumers, components of the system and their structure are already well-known.

## 5. Conclusions

The developed method has a good accuracy and its use in real-world situations is therefore recommended. The trained model which predicted the location of faults (with 23 possible responses) had an accuracy validation of 98%, while the trained model which predicted both location and type of faults (with 243 possible responses) had a slightly lower accuracy validation (94.7%). Both trained models were based on the same data training set (measurements from 6150 of simulations), which revealed that for obtaining a good accuracy validation value, a larger number of responses, and of the data training set were needed.

The major advantage of the presented method lies in its good precision in detecting, locating, and predicting faults. At the same time, running a large number of simulations could be considered a tedious operation. Clearly, the simulations' time was closely related to the complexity of the Simulink model and the number of parameters that needed to be modified at each simulation. Additionally, the developed Simulink model of the electrical power system could also decisively contribute by performing different updates of the system if necessary. It could be used, for example, to simulate the impact of a new structure of the system or to analyze possible future improvements performed on the electrical system.

Methods of faults detecting, locating, and predicting in electrical power systems, based on ANN algorithms, could also solve complex problems encountered at electric lines or branched systems of cables, situations difficult to be managed using classic methods of detecting faults.

In further research, this method could be improved in the direction of generating dynamic changes of the loads simulations, and in developing a friendly graphical user interface to the application.

The presented topic, regarding faults detection and location in electrical systems, was and remains a main and permanent concern of utility companies, which continuously seek to improve and adapt proprietary methods in order to have an optimal operation of their electrical power systems.

**Author Contributions:** Conceptualization, A.-M.M. and M.I.B.; methodology, A.-M.M.; software, A.-M.M.; validation, A.-M.M. and M.I.B.; formal analysis, M.I.B.; investigation, A.-M.M.; resources, A.-M.M.; data curation, M.I.B.; writing—original draft preparation, A.-M.M.; writing—review and editing, A.-M.M. and M.I.B.; visualization, M.I.B.; supervision, M.I.B.; project administration, A.-M.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Appendix A

```
T = [];
model = 'name_model';
tstop = 0.1;
%          1     2    3    4    5    6
Line.Name = {'L1', 'L2', 'L3', 'L4', 'L5','L6'};
Line.Sectors = [ 4    4     4    4    3    3 ];
Faults = {'0','ag','bg','cg','ab','bc','ac','abg','bcg','acg','abc','abcg'};
nLine = 1:6;
nFault = 1:12;
Faulty.Lines.Name = {Line.Name{nLine}};
Faulty.Sectors = Line.Sectors(nLine);
Type_Faults = {Faults{nFault}};
open_system(model);
for iFaultyLines = 1:length(Faulty.Lines.Name)
  for iFaultType = 1:length(Type_Faults)
    Fault_Type = Faults(iFaultType);
  Faulty.Lines = [model,'/',Faulty.Lines.Name{iFaultyLines}];
  open_system(Faulty.Lines);
  Nsectors = Faulty.Sectors(iFaultyLines);
  FaultBlock = [Faulty.Lines,'/Fault'];
  try
    addbd = add_block([model,'/Fault'],FaultBlock,'Commented','off','Position', [545
-545    595     -495]);
  catch
    Position = get_param(FaultBlock,'Position');
    delete_block(FaultBlock);
    add_block([model,'/Fault'],FaultBlock,'Commented','off','Position',Position + 10);
  end
if contains(Fault_Type,'g')
  set_param(FaultBlock,'GroundFault','on')
else
  set_param(FaultBlock,'GroundFault','off')
end
if contains(Fault_Type,'a')
  set_param(FaultBlock,'FaultA','on')
else
```

```
    set_param(FaultBlock,'FaultA','off')
end
if contains(Fault_Type,'b')
    set_param(FaultBlock,'FaultB','on')
else
    set_param(FaultBlock,'FaultB','off')
end
if contains(Fault_Type,'c')
    set_param(FaultBlock,'FaultC','on')
else
    set_param(FaultBlock,'FaultC','off')
end
for iSectors = 1:Nsectors
    NameBlock = [Faulty.Lines,'/S',num2str(iSectors)];
    hFault = get_param(FaultBlock,'PortHandles');
    hBlock = get_param(NameBlock,'PortHandles');
    hLines = add_line(Faulty.Lines,hFault.LConn,hBlock.RConn);
    Ron = logspace(log10(1e-4),log10(20),5); %FaultResistance
    Rg = logspace(log10(1e-4),log10(20),5); %GroundResistance
    for iR1 = 1:length(Ron)
        set_param(FaultBlock,'FaultResistance',num2str(Ron(iR1)));
        for iR2 = 1:length(Rg)
            set_param(FaultBlock,'GroundResistance',num2str(Rg(iR2)));
            out = sim(model);
            cellrow{1,1} = out.yout{9}.Values.V_B1.Data(end,:);
            cellrow{1,2} = out.yout{9}.Values.F_B1.Data(end,:);
            cellrow{1,3} = out.yout{9}.Values.V_B2.Data(end,:);
            cellrow{1,4} = out.yout{9}.Values.F_B2.Data(end,:);
            cellrow{1,5} = out.yout{9}.Values.V_B3.Data(end,:);
            cellrow{1,6} = out.yout{9}.Values.F_B3.Data(end,:);
            cellrow{1,7} = out.yout{9}.Values.V_B4.Data(end,:);
            cellrow{1,8} = out.yout{9}.Values.F_B4.Data(end,:);
            cellrow{1,9} = out.yout{9}.Values.V_B5.Data(end,:);
            cellrow{1,10} = out.yout{9}.Values.F_B5.Data(end,:);
            cellrow{1,11} = out.yout{9}.Values.V_B6.Data(end,:);
            cellrow{1,12} = out.yout{9}.Values.F_B6.Data(end,:);
            cellrow{1,13} = out.yout{9}.Values.V_B7.Data(end,:);
            cellrow{1,14} = out.yout{9}.Values.F_B7.Data(end,:);
            cellrow{1,15} = out.yout{9}.Values.V_B8.Data(end,:);
            cellrow{1,16} = out.yout{9}.Values.F_B8.Data(end,:);
            cellrow{1,17} = out.yout{10}.Values.I_B1.Data(end,:);
            cellrow{1,18} = out.yout{10}.Values.FI_B1.Data(end,:);
            cellrow{1,19} = out.yout{10}.Values.I_B2.Data(end,:);
            cellrow{1,20} = out.yout{10}.Values.FI_B2.Data(end,:);
            cellrow{1,21} = out.yout{10}.Values.I_B3.Data(end,:);
            cellrow{1,22} = out.yout{10}.Values.FI_B3.Data(end,:);
            cellrow{1,23} = out.yout{10}.Values.I_B4.Data(end,:);
            cellrow{1,24} = out.yout{10}.Values.FI_B4.Data(end,:);
            cellrow{1,25} = out.yout{10}.Values.I_B5.Data(end,:);
            cellrow{1,26} = out.yout{10}.Values.FI_B5.Data(end,:);
            cellrow{1,27} = out.yout{10}.Values.I_B6.Data(end,:);
            cellrow{1,28} = out.yout{10}.Values.FI_B6.Data(end,:);
            cellrow{1,29} = out.yout{10}.Values.I_B7.Data(end,:);
            cellrow{1,30} = out.yout{10}.Values.FI_B7.Data(end,:);
```

```
                    cellrow{1,31} = out.yout{10}.Values.I_B8.Data(end,:);
                    cellrow{1,32} = out.yout{10}.Values.FI_B8.Data(end,:);
                    FaultLocation = [Faulty.Lines.Name{iFaultyLines},'/S',num2str(iSectors)];
                    StringFaultLocation = convertCharsToStrings(FaultLocation);
                    if contains (Fault_Type,'0')
                        cellrow{1,33} = ("normal");
                    else
                        cellrow{1,33} = strcat(StringFaultLocation,"/",Fault_Type);
                    end
                    cellrow{1,34} = Ron(iR1);
                    cellrow{1,35} = Rg(iR2);
                    T1 = cellrow;
                    T = [T;T1];
                    save(date);
                    end
            end
            delete_line(hLines);
        end
            end
        DataTabel = cell2table(T);
        save('data_generation');
        end
```

## References

1. Sumit, S.V. Iterative and Non-Iterative Methods for Transmission Line Fault-Location Without using Line Parameters. *Int. J. Eng. Innov. Technol.* **2013**, *3*, 310–314.
2. Wang, L.; Liu, H.; Dai, L.V.; Liu, Y. Novel Method for Identifying Fault Location of Mixed Lines. *Energies* **2018**, *2018*, 11. [CrossRef]
3. Michau, G.; Hsu, C.-C.; Fink, O. Interpretable Detection of Partial Discharge in Power Lines with Deep Learning. *Sensors* **2021**, *21*, 2154. [CrossRef] [PubMed]
4. Muzzammel, R.; Arshad, R.; Raza, A.; Sobahi, N.; Alqasemi, U. Two Terminal Instantaneous Power-Based Fault Classification and Location Techniques for Transmission Lines. *Sustainability* **2023**, *15*, 809. [CrossRef]
5. Ali, S.; Bhargava, A.; Saxena, A.; Kumar, P. A Hybrid Marine Predator Sine Cosine Algorithm for Parameter Selection of Hybrid Active Power Filter. *Mathematics* **2023**, *11*, 598. [CrossRef]
6. Moldovan, A.M.; Oltean, S.; Buzdugan, M.I. Methods of Faults Detection and Location in Electrical Systems. In Proceedings of the 2021 9th International Conference on Modern Power Systems (MPS), Cluj-Napoca, Romania, 16–17 June 2021. [CrossRef]
7. Moldovan, A.M.; Buzdugan, M.I.; Oltean, S. Modeling a Time Domain Reflectometer using Matlab/Simulink for detection of faults in electrical cables. In Proceedings of the 2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC), Brasov, Romania, 25–28 September 2022; pp. 281–284. [CrossRef]
8. Tariq, R.; Alhamrouni, I.; Rehman, A.U.; Eldin, E.T.; Shafiq, M.; Ghamry, N.A.; Hamam, H. An Optimized Solution for Fault Detection and Location in Underground Cables Based on Traveling Waves. *Energies* **2022**, *15*, 6468. [CrossRef]
9. Lee, C.-K.; Chang, S.J. Fault Detection in Multi-Core C&I Cable via Machine Learning Based Time-Frequency Domain Reflectometry. *Appl. Sci.* **2020**, *10*, 158. [CrossRef]
10. Scarpetta, M.; Spadavecchia, M.; Adamo, F.; Ragolia, M.A.; Giaquinto, N. Detection and Characterization of Multiple Discontinuities in Cables with Time-Domain Reflectometry and Convolutional Neural Networks. *Sensors* **2021**, *21*, 8032. [CrossRef]
11. Tekli, L.; Filipovi, B.; Pavicic, I. Artificial Neural Network Approach for Locating Faults in Power Transmission System. *Eurocon* **2013**, *2013*, 1425–1430.
12. Mahafzah, K.A.; Obeidat, M.A.; Mansour, A.M.; Al-Shetwi, A.Q.; Ustun, T.S. Artificial-Intelligence-Based Open-Circuit Fault Diagnosis in VSI-Fed PMSMs and a Novel Fault Recovery Method. *Sustainability* **2022**, *14*, 16504. [CrossRef]
13. Forootan, M.M.; Larki, I.; Zahedi, R.; Ahmadi, A. Machine Learning and Deep Learning in Energy Systems: A Review. *Sustainability* **2022**, *14*, 4832. [CrossRef]
14. Kebir, S.T.; Cheggaga, N.; Ilinca, A.; Boulouma, S. An Efficient Neural Network-Based Method for Diagnosing Faults of PV Array. *Sustainability* **2021**, *13*, 6194. [CrossRef]
15. Aziz, R.M.; Mahto, R.; Goel, K.; Das, A.; Kumar, P.; Saxena, A. Modified Genetic Algorithm with Deep Learning for Fraud Transactions of Ethereum Smart Contract. *Appl. Sci.* **2023**, *13*, 697. [CrossRef]
16. Jamil, M.; Sharma, S.K.; Singh, R. Fault detection and classification in electrical power transmission system using artificial neural network. *SpringerPlus* **2015**, *4*, 334. [CrossRef]

17. Khorashadi-Zadeh, H.; Aghaebrahimi, M.R. A Novel Approach to Fault Classification and Fault Location for Medium Voltage Cables Based on Artificial Neural Network. *World Acad. Sci. Eng. Technol. Int. J. Energy Power Eng.* **2008**, *2*, 1273–1276.
18. Sharma, A.; Kumari, S.; Bhardwaj, I. Analysis of Underground Cable Fault Location Identification Using Artificial Neural Network. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 5190–5201.
19. Obuma, O.; Madueme, T. Application of artificial neural network (ANN) to enhance power systems protection: A case of the Nigerian 330 kV transmission line. *Electr. Eng.* **2018**, *100*, 1467–1479. [CrossRef]
20. Hasija, K.; Vadhera, S.; Kumar, A.; Kishore, A. Detection and location of faults in underground cable using Matlab/Simulink/ANN and OrCad. In Proceedings of the 2014 6th IEEE Power India International Conference (PIICON), Delhi, India, 5–7 December 2014; pp. 1–5. [CrossRef]
21. Hernandez-Mejia, J.C. Time Domain Reflectometry (TDR). In *Cable Diagnostic Focused Initiative (CDFI) Phase II*; Georgia Tech Research Corporation: Atlanta, GA, USA, 2016.
22. Warlyani, P.; Yadav, A.; Thoke, A.S.; Patel, R. Fault Classification and Faulty Section Identification in Teed Transmission Circuits Using ANN. *Int. J. Comput. Electr. Eng.* **2011**, *3*, 807–811. [CrossRef]
23. Yadav, A.; Thoke, A.S. Transmission line fault distance and direction estimation using artificial neural network. *Int. J. Eng. Sci. Technol.* **2011**, *3*, 110–121. [CrossRef]
24. Mantach, S.; Lutfi, A.; Tavasani, H.M.; Ashraf, A.; El-Hag, A.; Kordi, B. Deep Learning in High Voltage Engineering: A Literature Review. *Energies* **2022**, *15*, 5005. [CrossRef]
25. Wang, M.-H.; Sian, H.-W.; Lu, S.-D. Hybrid Methodology Based on Symmetrized Dot Pattern and Convolutional Neural Networks for Fault Diagnosis of Power Cables. *Processes* **2022**, *10*, 2009. [CrossRef]
26. Al-Katheri, A.A.; Al-Ammar, E.A.; Alotaibi, M.A.; Ko, W.; Park, S.; Choi, H.-J. Application of Artificial Intelligence in PV Fault Detection. *Sustainability* **2022**, *14*, 13815. [CrossRef]
27. Nsaif, Y.M.; Lipu, M.S.H.; Hussain, A.; Ayob, A.; Yusof, Y.; Zainuri, M.A.A.M. A Novel Fault Detection and Classification Strategy for Photovoltaic Distribution Network Using Improved Hilbert–Huang Transform and Ensemble Learning Technique. *Sustainability* **2022**, *14*, 11749. [CrossRef]
28. Hichri, A.; Hajji, M.; Mansouri, M.; Abodayeh, K.; Bouzrara, K.; Nounou, H.; Nounou, M. Genetic-Algorithm-Based Neural Network for Fault Detection and Diagnosis: Application to Grid-Connected Photovoltaic Systems. *Sustainability* **2022**, *14*, 10518. [CrossRef]
29. Omar, A.M.S.; Osman, M.K.; Ibrahim, M.N.; Hussain, Z.; Abidin, A.F. Fault classification on transmission line using LSTM network. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *20*, 231–238. [CrossRef]
30. Available online: https://www.mathworks.com/help/ (accessed on 6 March 2023).