Measurement Aided Training of Machine Learning Techniques for Fault Detection Using PLC Signals

Yinjia Huo[1], Gautham Prasad[1], Lutz Lampe[1], Victor C. M. Leung[2,1], Rathinamala Vijay[3], and TV Prabhakar[3]
[1]The University of British Columbia, Canada, [2]Shenzhen University, China, [3]Indian Institute of Science, India

***Abstract*—The re-use of channel estimation performed by power line communication (PLC) modems for monitoring of cable health conditions has recently been investigated in several works. In particular, cable diagnostics solutions based on machine learning techniques have been shown to process the PLC channel-estimation samples intelligently to differentiate fault conditions from the benevolent load changes. Previous studies have been based on synthetically generated training and test signals to optimize and validate the machine learning models. To deal with the mismatches between the purely synthetically generated signal samples and those encountered in a real implementation, in this paper, we propose S-parameter measurement aided generation of channel estimation samples. Specifically, we describe the behaviour of our device under test (DUT) through its S-parameter measurement and synthetically generate varying terminal load conditions. Then we train and use machine learning models to determine the health of the DUT. We describe the proposed approach and apply it to data obtained from laboratory measurements.**

I. Introduction

Cable diagnostics constitutes an indispensable part for the asset monitoring to ensure the safe operation of smart power grids [1]. Various cable diagnostics solutions have been developed in the past to help preventing cable in-service failures, which could lead to potentially dangerous situations and severe economic losses [2, Ch. 6]. One approach is to determine the cable health conditions by analyzing the high-frequency signals propagated along the cable. This is based on the principle that the inception of a cable anomaly alters the electric signal propagation [2, Ch. 6].

In recent years, several contributions have been made on cable diagnostics solutions using power line communications (PLC), including [3]–[6]. They exploit the fact that power line modems (PLMs) obtain information about the propagation medium by way of channel or impedance estimation, which can be reused to infer the cable health conditions. This enables the continuous monitoring of the cable health and avoids the additional overhead of installing dedicated sensors solely for the diagnostics purposes.

In [3], [5], [7], cable diagnostics solutions based on the comparison between the currently measured PLC channel state information (CSI) and a healthy reference measurement have been developed. However, the load conditions of a PLC network are constantly changing and the network topology varies from time to time. These render any deviation from the healthy reference measurement unreliable in determining the presence of a potential cable anomaly. Furthermore, traditional methods using the principle of traveling wave, reflectometry [8], broadband impedance spectroscopy [9], and partial discharge detection [10], which also exploit high-frequency signal propagation characteristics, require manual result interpretation by a technician with expertise.

To overcome the drawback of requiring an unreliable healthy reference measurement and to enable the automatic analysis of the sampled electric signal, several machine learning (ML) based PLC cable diagnostics solutions have been developed [4], [6], [11], [12]. In this case, CSI samples of healthy and faulty cable instances are used to train a machine. Once the ML model has been trained, it can intelligently analyze the incoming CSI sample and autonomously determine the cable health condition associated with this sample.

However, one of the issues encountered in the field implementation of these ML based schemes is the acquisition of training samples. On the one hand, it is typically infeasible to obtain a sufficient amount of training data from measurements, covering different cable states and network settings. On the other hand, synthetically generated samples can be expected to be mismatched to some degree compared to CSI sampled in the field. Several factors could lead to such mismatches. Firstly, information about the setting of the field implementation could be incomplete, e.g., the impedance characteristics of loads in the PLC band. Secondly, the exact parameters of the cable under test may not be documented or known. Thirdly, approximations in the analysis of signal propagation over multi-conductor cables lead to a mismatch.

In an effort to alleviate the problem of training data mismatch, in this paper, we propose the measurement-aided generation of training data. In particular, we consider scattering parameter (S-parameter) measurements of the cable section that is being monitored as input to the synthetic generation of CSI samples used for training, and we build a larger training set through the variation of network loads at the ends of the cable. We expect that this provides a mechanism to obtain a training set that is more consistent with data observed in the actual deployment compared to synthetically generated data using only a channel emulator. As a specific use case, we consider the ML-based cable diagnostics using samples of the end-to-end channel frequency response (CFR) [6].

The remainder of this paper is organized as follows. The details of S-parameter measurements and synthetic channel generation based on the extracted information are presented in Section II. In Section III, we describe the ML-based cable diagnostics solution based on the synthetically generated channel. In Section IV, we conduct numerical evaluations, and

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

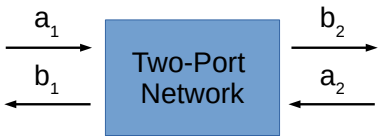

Fig. 1: Abstraction of a two-port network.

Fig. 2: Two-port network with signal voltages and currents.

Section V concludes this work.

II. CFR Synthesis using S-Parameter Measurements

We first briefly review the basics of the S-parameter description of two-port networks and then proceed to the method of obtaining CFR samples for the training of the ML model.

A. S-Parameter Measurement

S-parameters are used to describe the electrical behavior of a multi-port network composed of linear electrical components operated under the steady state electrical signal stimuli [13, Ch. 4].

The behaviour of a two-port network as shown in Fig. 1 is described by four S-parameters, i.e., S_{11}, S_{12}, S_{21} and S_{22}. These are defined through

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \tag{1}$$

where a_1 is the incident power wave at port 1 propagated towards the two-port network, a_2 is the incident power wave at port 2, b_1 is the reflected power wave at port 1 propagated outward from the two-port network, and b_2 is the reflected power wave at port 2 (see Fig. 1). Note that to determine the S-parameters, it is assumed that each port of the network is terminated with a system impedance Z_0, which usually takes the value of $50\ \Omega$.

B. PLC Signal Transmission Through Two-Port Networks

While the S-parameters can be obtained through measurements, e.g. from a vector network analyzer, it is more convenient to consider the ABCD parameters of the two-port network to analyze the PLC signal propagation. The ABCD parameters relate voltages and currents as shown in Fig. 2. In particular, we define the voltage across port 1 as V_1, the voltage across port 2 as V_2, the current entering the two-port network from port 1 as I_1, the current leaving the two-port network from port 2 as I_2. Then we obtain the ABCD parameters from

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} V_2 \\ I_2 \end{bmatrix}. \tag{2}$$

The relationship between the quantities shown in Figs. 1 and 2 is given by

$$a_i = \frac{1}{2} \frac{V_i + Z_0 I_i}{\sqrt{|\Re\{Z_0\}|}}, \tag{3}$$

$$b_i = \frac{1}{2} \frac{V_i - Z_0^* I_i}{\sqrt{|\Re\{Z_0\}|}}, \tag{4}$$

where $(\cdot)^*$ is taking the complex conjugate, $\Re\{\cdot\}$ is taking the real part of a complex number and $i \in \{1, 2\}$. Using Equations (1) to (4), we can obtain the ABCD parameters from the measured S-parameters as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{2S_{21}} & Z_0 \frac{(1+S_{11})(1+S_{22})-S_{12}S_{21}}{2S_{21}} \\ \frac{1}{Z_0} \frac{(1-S_{11})(1-S_{22})-S_{12}S_{21}}{2S_{21}} & \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{2S_{21}} \end{bmatrix} \tag{5}$$

C. Data Synthesis for CFR Samples

We investigate a PLC network as shown in Fig. 3, where TX is the transmitter power line modem (PLM), RX is the receiver PLM, Z_S is the source impedance of TX, the two-port network 1 is the cable, or generally the device-under-test (DUT), we wish to monitor, and Z_{L1} is the network load impedance at the receiver side. In this setting, we assume that the PLM at the transmitter can provide the signal voltage regardless of the experienced network impedance. Then, if we define the voltage across the transmitter as V_S and the voltage across the receiver as V_L, measured from the '+' end to the '-' end as shown in Fig. 3, we would like to faithfully generate the end-to-end CFR samples between TX and RX, H, defined as[1]

$$H = \frac{V_\mathsf{L}}{V_\mathsf{S}}. \tag{6}$$

Using [14, Eq. 2.26], we can calculate H using

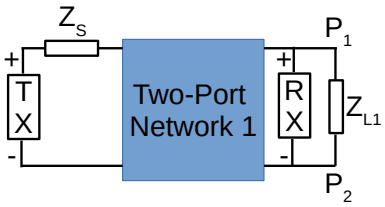

Fig. 3: Model for the setup for cable monitoring using PLMs.

[1]Note that the values of H, as for the quantities introduced earlier, are a function of the wave or signal frequency f, which we do not make explicit for compactness of notation.

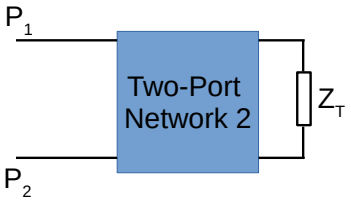

Fig. 4: Setup to draw network loads Z_{L1} for the model in Fig. 3.

$$H = \frac{Z_{\mathsf{L}}}{A_1 Z_{\mathsf{L}} + B_1 + C_1 Z_{\mathsf{S}} Z_{\mathsf{L}} + D_1 Z_{\mathsf{S}}}, \tag{7}$$

where A_1, B_1, C_1, D_1 are ABCD parameters of the two-port network 1 and Z_{L} is the total equivalent impedance at the receiver side. We can further calculate Z_{L} using $\frac{1}{Z_{\mathsf{L}}} = \frac{1}{Z_{\mathsf{RX}}} + \frac{1}{Z_{\mathsf{L1}}}$, where Z_{RX} is the impedance of RX.

We obtain the values for A_1, B_1, C_1, D_1 through S-parameter measurements. For this, we assume that both healthy and faulty states of the cable can be generated. Since this is typically not possible in an actual deployment, we expect that the DUT can be represented through an experimental setup in a laboratory.

The remaining task to generate CFR samples is to obtain values for Z_{L}. For this, we use a synthetic approach in an attempt to capture the nature of typical and varying network load conditions. In [6], the terminal loads were assumed to be drawn from a uniform distribution, which was done to introduce uncertainty due to varying load conditions. But this approach is quite simplistic as it results in frequency-flat impedances.

In this work, considering the typically frequency-selectivity nature of load impedances in the PLC network, we consider a generator network as shown in Fig. 4, to draw samples of Z_{L1}. The ABCD parameters of the two-port network 2 would be chosen to mimic what is seen into the network at the receiver side. For example, if this part of the network is similar to the monitored cable, we can set A_2, B_2, C_2, D_2 according to the S-parameter measurements obtained for two-port network 1. The remainder of the network, whose effect on Z_{L1} is largely "filtered" through two-port network 2 is then modeled through a terminal impedance Z_{T} generated uniformly at random within a certain range. The network load impedance used in Fig. 3 can then be calculated as [14, Eq. 2.27]

$$Z_{\mathsf{L1}} = \frac{A_2 Z_{\mathsf{T}} + B_2}{C_2 Z_{\mathsf{T}} + D_2}. \tag{8}$$

III. ML-based Cable Diagnostics

The task for the ML-based cable diagnostics is to identify whether there is an anomalous condition between the transmitter PLM and the receiver PLM, i.e., in the DUT represented by two-port network 1 in Fig. 3. In this work, we formulate this as a supervised classification task. In the following, we first link the discussion about S-parameter measurements above to the training and testing data generation for the ML model. Then, we briefly review the ML models used in this study.

A. Training and Test Data Generation

For the purpose of supervised classification, the training data should consist of measurements from healthy and abnormal conditions of the specific DUT. Especially the latter may be difficult to obtain from in-field measurements. We therefore expect that in a practical scenario the DUT with healthy and faulty states can be reproduced in a laboratory setting. There may also be a number of n_{h} and n_{f} instances of S-parameter measurements under healthy and abnormal conditions, respectively. The n_{h} and n_{f} measurements would reflect variations of the DUT, due to for example, load changes when the DUT is a cable section with a branch point. To account for the effect of varying network loads outside the two-port network 1, we use each of the $(n_{\mathsf{h}} + n_{\mathsf{f}})$ measurements as seeds to generate n_{L} CFR instances associated with different values Z_{L}. The randomness is introduced through choosing the load Z_{T} in Fig. 4 uniformly at random from a certain range of impedances. As described in Section II-C, the ABCD parameters of the two-port network 2 through which Z_{T} is transformed would be selected to model the neighbouring cable segment. For the results shown in Section IV, we choose one of the S-parameter measurements for the DUT corresponding to one cable type uniformly at random with each realization for Z_{T}. This means that we mimic the case where the connecting line segment is similar in nature to the DUT but could be in different states, including healthy or faulty.

From the total number of terminal load conditions, i.e., n_{L} instances of Z_{L}, we will have $n_{\mathsf{h}} n_{\mathsf{L}}$ CFR samples for healthy cables and $n_{\mathsf{f}} n_{\mathsf{L}}$ CFR samples for cables with anomalies. For our numerical tests in Section IV, we will apply equal (balanced) numbers of seed measurements for healthy and faulty cable conditions, i.e., $n_{\mathsf{h}} = n_{\mathsf{f}}$. Furthermore, we also consider the scenario of using only a fraction of the measurements as seeds to generate samples for training and validation, so as to explore the generalization ability of the diagnostics method.

B. Considered ML Technique

We apply the support vector machine (SVM) in this paper, which is a common and well-investigated machine learning technique that has shown overall good performance across various applications [15] and also for cable diagnostics [4], [6]. The SVM model encourages a parsimonious solution through the hinge loss function, leading to a small number of support vectors that define the ML model [15, Ch. 14]. Such sparsity derived from the large margin principle of the SVM algorithm avoids over-fitting and enables a good generalization capability of the trained machine, i.e., the ability to predict well for unseen data samples.

C. Feature Extraction

The extraction of features is an important element for many machine learning tasks. Our previous works [4], [16] have

suggested that moments statistics of the CFR and the channel impulse response are informative features to determine the cable health. We thus adopt the 16 features presented in [16, Section IV] also in this work. We note however that feature extraction runs the risk of incurring a critical information loss, if not well or fully suited for capturing the essence of the targeted label contained in the input data. In Section IV, we therefore also consider the option of using the measured CFR directly as input to the machine learning model.

IV. Numerical Evaluations

In this section, we present numerical results for the proposed approach. That is, we perform data-set generation, and training and testing of ML models based on measurements for a DUT. Ideally, we would then test the effectiveness of our proposed scheme in in-field test. However, here we only show the results for testing based on the generated data, as we did not have the option for in-field testing.

A. S-Parameter Measurements

The S-parameter measurement were conducted at the Zero Energy Networks Laboratory at Indian Institute of Science in Bangalore, India. We consider a 30 m symmetrical 4-conductor PVC insulated YY control cable [17]. The cable is connected to a variable load, that is added as a branch point at one end of the cable, so that the cable together with the variable load is considered as the DUT and the variable load can be viewed as a DUT internal load. All the S-parameter measurements were performed for 201 equally spaced frequency points from 2 MHz to 100 MHz.

The fault introduced to the YY control cable consists of the removal of a 5 mm wide piece of the outer sheath in the middle of the cable. Note that the individual insulation wrapped around each conductor is kept intact. We consider two different lengths for the removed patches of insulation, namely of 5 mm and 3 cm. This gives us three different health states of the DUT in total, i.e., healthy state, 5 mm fault and 3 cm fault. For each health state, the measurements are repeated 30 times for each of the DUT internal load conditions. A total of eight DUT internal load conditions are considered, including no load, one light bulb (LB) load at R, B, Y, two LB loads at BY, RB, RY and three LB loads at RBY. Here, one end of each LB is connected to one of the three conductors, R, B, Y, while the other end of the LB is connected to the neutral conductor N.

B. Generation of Data Sets

We generate 1,000 samples for each of the DUT internal load conditions to train and test the ML models under various external load conditions of Z_{L1}. For this, we use the 30 available measurements and pair them with randomly generated samples of Z_{L1}. To obtain Z_{L1}, we sample $Z_{\mathsf{T}} \sim (\mathscr{U}[0,50] + j\mathscr{U}[-50,50])\ \Omega$, where $\mathscr{U}[a,b]$ is the uniform distribution between a and b, and j is the imaginary unit. Then, we uniformly at random select one of the measurements to obtain the ABCD parameters of the two-port network 2, as shown in Fig. 4, using (5), and Z_{L1} follows from (8). Furthermore, we set $Z_{\mathsf{S}} = 0$, as it is usually very small, and $Z_{\mathsf{RX}} = 100\ \Omega$ [18].

C. Results

The SVM is implemented with data standardization (normalization) and automated kernel scale as provided in MATLAB. The type of kernel used is determined through the validation process. For most of the results presented below, we use the polynomial kernel with degree 3. The exception is the fault type classification task with feature extraction, where the results are obtained with the radial basis function kernel. Different kernels offer different tradeoffs between the detection rate, P_{det}, and the false alarm (FA) rate, P_{FA}. The desired tradeoff should be determined through the requirements of the application.

1) Individual Fault: First, we use the CFR samples for the healthy DUT and those for one particular type of fault to train and test the machine.

a) Shuffled Data: In this exercise, we randomly shuffle the healthy and faulty samples and use a proportion of $\frac{4}{5}\eta$ of the total samples for training and validating the machine, where $\eta \in \{0.1, 0.2, 0.5, 0.8, 1\}$. The remaining samples are used for testing. With this setup, we achieve perfect classification results in terms of P_{det} and P_{FA} for both types of faults and for all the values of η that we considered. This is the case for when using the SVM with and without the feature extraction, where the latter means that the SVM directly processes the samples with the CFR values (including phase and magnitude) across all the 201 subcarriers. The results for the shuffled data show that the ML task under such setting could be straightforward. Note that under the setting of the shuffled data, the training and testing samples have the same distribution. Therefore, in the following, we study the generalization capability of our trained machine under the setting of the partitioned data, where training and testing samples have different distributions.

b) Partitioned Data: We investigate the performance of our trained machine when we only use a fraction of the measurements corresponding to a subset of the DUT internal load conditions as seeds to generate the ABCD parameters of the DUT for training and validation, and the remaining measurements are used as seeds to generate CFR samples for testing. Specifically, measurements with m different DUT internal load conditions are used to generate the CFR samples for training and validation, while the measurements with the $8 - m$ remaining DUT internal load conditions are used to generate the testing samples, where $0 < m < 8$. We expect a better performance of the trained machine, both in terms of P_{det} and P_{FA}, with increasing m, since more situations are fed into the machine for training. But we are interested in how many measurements are needed. In particular, we argue that if a good performance is achieved with a lower value of m, the machine has a better generalization capability to unseen samples.

TABLE I: Classification results for partitioned data for 5 mm fault, with ("w/") and without ("w/o") feature extraction.

m	1	2	3	4	5	6	7
P_{det} w/ feature	0.53	0.69	0.59	1.00	1.00	1.00	1.00
P_{FA} w/ feature	0.00	0.00	0.00	0.00	0.00	0.00	0.00
P_{det} w/o feature	1.00	1.00	1.00	1.00	1.00	1.00	1.00
P_{FA} w/o feature	0.00	0.00	0.00	0.00	0.00	0.00	0.00

TABLE II: Classification results for partitioned data for 3cm fault, with ("w/") and without ("w/o") feature ("F") extraction.

m	1	2	3	4	5	6	7
P_{det} w/ F	0.86	0.87	0.89	1.00	1.00	1.00	1.00
P_{FA} w/ F	0.00	0.00	0.00	0.00	0.00	0.00	0.00
P_{det} w/o F	1.00	1.00	1.00	1.00	1.00	1.00	1.00
P_{FA} w/o F	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The results for the 5 mm fault case are shown in Table I, where perfect classification results are obtained for all values of m without feature extraction and $m \geq 4$ with feature extraction. The results for the 3 cm fault case are shown in Table II, where the machine trained without feature extraction has a better performance than the machine trained with feature extraction for all possible values of m. In this regard, the machine trained without feature extraction has a better generalization capability to unseen samples. This could probably be attributed to the fact that the machine is able to learn more aspects of the training samples without feature extraction while the feature extraction process could lose some of these aspects. We also note that the results for the partitioned data depend on how we do the partition. For the results presented here and in the following, we chose one partition and kept it the same for all experiments.

2) Combined Faulty Setting: Our second experiment involves training and testing using healthy samples and faulty samples with multiple types of faults.

a) Fault detection: To detect the fault under combined faulty settings, we combine the 5 mm fault CFR samples with the 3 cm fault CFR samples so that the faulty sample size is doubled. Then we generate additional 8000 CFR samples (1000 for each of the DUT internal load conditions) with the healthy DUT condition to obtained balanced healthy and faulty samples. Table III shows the results for shuffled data, where 80% of the shuffled CFR samples are used for training and validation and the remaining 20% samples are used for testing. Then, the CFR samples with m different DUT internal load conditions are used for the training and validation, while the CFRs with the remaining $8-m$ DUT internal load conditions are used for the testing, where $0 < m < 8$. The results for the partitioned data are shown in Table IV.

TABLE III: Fault detection results for shuffled data under combined faulty setting. Results without feature extraction in parentheses.

	Predicted healthy	Predicted faulty
Actual healthy	1.00 (1.00)	0.00 (0.00)
Actual faulty	0.00 (0.00)	1.00 (0.00)

TABLE IV: Fault detection results for partitioned data under combined faulty setting, with ("w/") and without ("w/o") feature ("F") extraction.

m	1	2	3	4	5	6	7
P_{det} w/ F	0.97	0.91	0.94	0.95	1.00	1.00	1.00
P_{FA} w/ F	0.33	0.01	0.06	0.00	0.00	0.00	0.00
P_{det} w/o F	1.00	1.00	1.00	1.00	1.00	1.00	1.00
P_{FA} w/o F	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The performance for the shuffled-data case is quite similar to that when the task was to detect a single fault type, i.e., the 5 mm or the 3 cm fault. Similarly, the performances reported in Table IV are quite similar to those in Tables I and II. The main difference is that the decreasing detection and the increasing false alarm rates with smaller values of m is notably less pronounced for the combined faulty setting in Table IV. This suggests that learning from additional albeit different faulty conditions can improve to differentiate healthy from faulty scenarios. We note that in the combined faulty setting, the characteristics of the faulty samples learned by the machine are not associated with a particular type of fault anymore. Hence, the machine is able to learn more general characteristics of different types of faults.

b) Fault classification: Next, in the case that the DUT has been classified as faulty, we want to further classify whether the fault is the 5 mm case or the 3 cm case. We achieve perfect classification for shuffled data (80% for training and validation, and 20% for testing). The results for the partitioned data, presented as a success rate for correct fault-type classification, are shown in Table V. We observe that for larger values of m, machines trained with or without feature extraction have decent classification results. However, for smaller value of m, we again note that the machine trained without feature extraction has better generalization capability to unseen samples.

TABLE V: Probability of successful fault-type classification for partitioned data under combined faulty setting, with ("w/") and without ("w/o") feature ("F") extraction.

m	1	2	3	4	5	6	7
P_{cor} w/ F	0.57	0.88	0.72	0.94	0.96	0.99	0.98
P_{cor} w/o F	0.71	1.00	1.00	1.00	1.00	1.00	1.00

3) Cross Faulty Settings: The results from the previous sections suggest the inclusion of multiple types of faults in the training samples in order to diagnose a deviation from healthy conditions. However, it may not be possible to have data for certain fault types during the training process, and thus the trained machine may experience unseen fault types. Therefore, in this last part, we investigate training with one type of fault (together with the healthy CFR samples) and testing with another type of fault. We again apply data partitioning. That is, we use the healthy and faulty samples with m different DUT internal load conditions to train and validate the machine, and

TABLE VI: P_{det} for first cross faulty setting, with ("w/") and without ("w/o") feature extraction.

m	1	2	3	4	5	6	7	8
Train with 3 cm fault, test with 5 mm fault								
w/	0.69	0.50	0.57	0.57	0.63	0.71	0.73	0.65
w/o	0.24	1.00	0.93	1.00	1.00	1.00	1.00	1.00
Train with 5 mm fault, test with 3 cm fault								
w/	0.60	0.71	0.79	0.82	0.77	0.81	0.85	0.83
w/o	0.77	0.92	0.97	0.83	0.83	0.90	0.99	0.99

we use the faulty samples with another type of fault and all the 8 different DUT internal load conditions to test the machine.

The results[2] for training with one fault and testing with the other fault are presented in Table VI. We see from the results that the machine has a decent performance for detecting the other type of the fault which has not be seen during the training, particularly for large values of m. This demonstrates a generalization capability of the trained machine to unseen fault types.

4) Summary: For the limited data set available to us, we found that generally a good classification performance is obtained. The results for the combined faulty setting show that it is doable and in fact preferable to include multiple types of faults during the training. The results for data partitioning and cross faulty types indicate the machine's generalization capability to unseen internal load conditions and fault types, respectively. This motivates us to conduct further investigations with more measurement data for more fault types, including the softer faults, as well as other cable types.

V. Conclusion

In this paper, we presented a scheme to extract critical information from the S-parameter measurements of cable sections to generate power line communications channel state information samples for the machine learning based cable diagnostics solution. The numerical evaluations using synthetic data show an overall good performance under various cable and fault types with a reasonable amount of measurements taken for training. This motivates us to further conduct in-field tests to verify the effectiveness of our proposed data synthesis scheme to generate faithful samples for the machine learning based PLC cable diagnostics solution.

References

[1] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18–28, 2010.

[2] P. Gill, *Electrical power equipment maintenance and testing*. CRC press, 2008.

[3] A. M. Lehmann, K. Raab, F. Gruber, E. Fischer, R. Müller, and J. B. Huber, "A diagnostic method for power line networks by channel estimation of PLC devices," in *IEEE Intl. Conf. Smart Grid Commun. (SmartGridComm)*, pp. 320–325, 2016.

[4] L. Förstel and L. Lampe, "Grid diagnostics: Monitoring cable aging using power line transmission," in *IEEE Int. Symp. Power Line Commun. and its Appl. (ISPLC)*, pp. 1–6, 2017.

[2]We only show the results for testing on faulty data (with a different type of fault), since the false-alarm rates are identical with those reported earlier (see Section IV-C1b).

[5] F. Passerini and A. M. Tonello, "Smart grid monitoring using power line modems: Anomaly detection and localization," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6178–6186, 2019.

[6] Y. Huo, G. Prasad, L. Atanackovic, L. Lampe, and V. C. M. Leung, "Cable diagnostics with power line modems for smart grid monitoring," *IEEE Access*, vol. 7, pp. 60206–60220, 2019.

[7] F. Passerini and A. M. Tonello, "full duplex power line communication modems for network sensing," in *IEEE Intl. Conf. Smart Grid Commun. (SmartGridComm)*, pp. 1 – 5, 2017.

[8] J. Wang, P. Stone, Y.-J. Shin, and R. Dougal, "Application of joint time–frequency domain reflectometry for electric power cable diagnostics," *IET Signal Process.*, vol. 4, no. 4, pp. 395–405, 2010.

[9] Z. Zhou, D. Zhang, J. He, and M. Li, "Local degradation diagnosis for cable insulation based on broadband impedance spectroscopy," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 22, no. 4, pp. 2097–2107, 2015.

[10] G. Montanari and A. Cavallini, "Partial discharge diagnostics: from apparatus monitoring to smart grid assessment," *IEEE Electr. Insul. Mag.*, vol. 29, no. 3, pp. 8–17, 2013.

[11] H. Livani and C. Y. Evrenosoglu, "A machine learning and wavelet-based fault location method for hybrid transmission lines," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 51–59, 2014.

[12] N. A. Letizia and A. M. Tonello, "Supervised fault detection in energy grids measuring electrical quantities in the PLC band," in *IEEE Int. Symp. Power Line Commun. Applicat. (ISPLC)*, pp. 1–5, 2020.

[13] D. M. Pozar, *Microwave engineering*. John wiley & sons, 2011.

[14] L. Lampe, A. M. Tonello, and T. G. Swart, *Power Line Communications: Principles, Standards and Applications from multimedia to smart grid*. John Wiley & Sons, 2016.

[15] K. Murphy, *Machine Learning: A Probabilistic Perspective*. Adaptive computation and machine learning, MIT Press, 2012.

[16] Y. Huo, G. Prasad, L. Atanackovic, L. Lampe, and V. C. M. Leung, "Grid surveillance and diagnostics using power line communications," in *IEEE Int. Symp. Power Line Commun. and its Appl. (ISPLC)*, pp. 1–6, 2018.

[17] Dungannon Electrical, "YY control cable," https://www.dungannonelectrical.co.uk/dun1-shop/pdf/cable/YY.pdf.

[18] G. Prasad, L. Lampe, and S. Shekhar, "In-band full duplex broadband power line communications," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3915–3931, 2016.