

# Partial Discharge Identification in MV switchgear using Scalogram representations and Convolutional AutoEncoder

Sonia Barrios, David Buldain, Maria Paz Comech and Ian Gilbert

**Abstract--** This work proposes a methodology to automate the recognition of Partial Discharges (PD) sources in Electrical Distribution Networks using a Deep Neural Network (DNN) model called Convolutional Autoencoder (CAE), which is able to automatically extract features from data to classify different sources. The database used to train the model is constructed with real defects commonly found in MV switchgear in service, and it also includes noise and interference signals that are present in these installations. PD sources consist of defective mountings, such as the loss of sealing cap of cable terminations, or an earth cable in contact with cable termination insulation. Four sources were replicated in a Smart Grid Laboratory and on-line measurement techniques were used to obtain the PD signal data. The Continuous Wavelet Transform (CWT) was applied to post-process the PD signal into a time-frequency image representation. The trained model predicts with high accuracy new data, demonstrating the effectiveness of the methodology to automate the recognition of different partial discharges and to differentiate them from noise and other interference sources.

**Index Terms--** Condition monitoring, convolutional neural networks, deep learning, fault diagnosis, image classification, partial discharge, signal processing, substations, wavelet transforms

## I. INTRODUCTION

PARTIAL discharge (PD) is a phenomenon generated as a consequence of dielectric insulation imperfections such as voids in solid dielectrics or sharp edges in air. In this phenomenon, an avalanche of electrons is produced, creating a very fast transient current pulse at very high frequencies whose width and rise time depend on the PD source that has generated it. This is because each PD defect has its own particular degradation mechanism, where discriminatory features from the measured pulses can be extracted [1].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 676042.

S. Barrios is with School of Engineering, University of Zaragoza, 50018 Zaragoza, Spain (email: soniabarriospr@gmail.com)

P. Comech is with Instituto CIRCE, University of Zaragoza, 50018 Zaragoza, Spain (e-mail: mcomech@unizar.es)

D. Buldain is with Department of Electronic Engineering and Communications, University of Zaragoza, 50018 Zaragoza, Spain (email: buldain@unizar.es)

I. Gilbert is with Ormazabal Corporate Technology, 48340, Amorebieta, Spain (e-mail: igi@ormazabal.com)

The detection of PD in the energized electrical system is crucial to prevent costly outages of the electrical network and thus, the identification of the source before total breakdown is essential. For this reason, the classification (or identification) of PD has been widely investigated over the last years. A review of different approaches of these methods can be found in [2]-[3].

Nowadays, to automate classification tasks, new algorithms based on Deep Neural Networks (DNN) have been implemented [4]-[6]. This technique, which has demonstrated to be highly effective, can be included within a larger maintenance system and used as a PD diagnosis tool in a "smart condition monitoring" context. However, this technique presents interesting challenges to its general applicability and certain improvements in the methodology have to be done to obtain a good classifier.

The methodology of a DNN technique is illustrated in Fig. 1, and consists of several steps: a) Create a database containing relevant information of the phenomena that will be classified, this data could be images, audio, etc.; b) Define and train the DNN model with the database; and, c) Use the trained model to predict (or classify) new test data.

The Partial Discharge data representations that have been used in literature to build the databases are: Phase Resolved Pattern (PRPD) [5]-[6], Spectrogram image [7]-[8] and, Time-Domain Waveform [9]-[11]. Most of the datasets used in these references were collected from artificial PD sources constructed in laboratories, and not enough data from real installations was used.

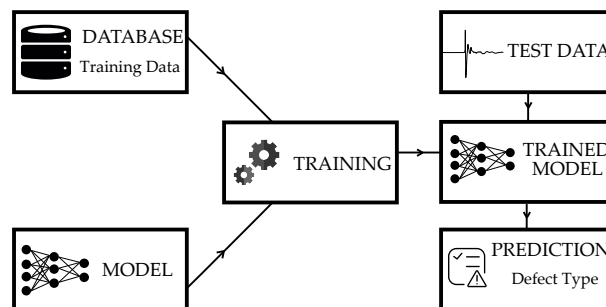


Fig. 1. Methodology for classification technique using Deep Neural Networks.

The design of a proper database is the main key to make this technique applicable in real electrical installations because, if the data used to train the model do not represent exactly the same phenomenon found in the installations or if they present biases, the trained DNN model will not make a reasonable prediction.

This work is focused on the identification of PD sources found in MV substation facilities by using a Convolutional Autoencoder (CAE) to extract the features from PD data and classify the different sources. The data used for training and testing the CAE model has been developed in this work and consists of real defective mountings found in MV switchgears. Furthermore, to build the database, an alternative PD data representation is used, which is a time-frequency image of a single PD signal, called *Scalogram*. This representation optimizes the time of data acquisition and guarantees a faithful representation of the PD phenomenon.

Hereafter, the technique used to obtain the time-frequency representation is explained in section 2. In Section 3, the PD sources and the experimental setup to collect the data are presented. The processing steps to construct the database are detailed in section 4. Section 5 introduces the CAE model and the methodology implemented. The practical considerations of the proposed method and the results are presented in section 6, followed by the conclusions in section 7.

## II. TIME-FREQUENCY SIGNAL REPRESENTATION

A common time-frequency representation used to analyze and classify PD sources is the *Spectrogram* obtained with Fourier Transform. Since this technique may be inadequate to represent PDs, an alternative representation given by Continuous Wavelet Transform (CWT) is applied in this work.

Spectrogram is defined as the square modulus of the Short-Time Fourier Transform (STFT) and it provides a distribution of the energy of the signal in the time-frequency plane, i.e. it shows the changes of the spectrum density of the signal power in time. Next equation shows its mathematical expression:

$$STFT(\tau, f) = \int x(t)h^*(t - \tau)e^{-2\pi jft} dt \quad (1)$$

As shown in (1), the STFT maps the signal into a two-dimensional function in the time-frequency plane  $(\tau, t)$ . The signal  $x(t)$  is assumed to be stationary when it is seen through a window  $h(t)$  of a limited extent, centered at time location  $\tau$ . The analysis here depends critically on the choice of this window function.

The main problem with the STFT approach is known as the *Uncertainty principle* or *Heisenberg inequality*. This principle originally applied to the momentum and location of moving particles, can be applied to time-frequency information of a signal. This principle states that is not possible to know what frequency components exist at what precise time, it is just possible to know the time intervals in which certain bands of frequencies exist, which is a resolution problem.

To overcome the resolution limitation of the STFT, which gives fixed resolution all the time, the CWT analyzes the signal at different frequencies with different resolutions. Therefore, the CWT is a generalization of the STFT that can be used to perform multi-resolution signal analysis [14].

The mathematical definition of CWT is expressed in (2) and, as it can be seen, the transformed signal is a function of two variables: the translation parameter,  $\tau$  and the scale parameter,  $a$ . The transforming function is  $g(\tau, a)$  and is called the *mother wavelet*. The constant  $1/\sqrt{a}$  is used for energy normalization.

$$CWT(\tau, a) = \frac{1}{\sqrt{a}} \int x(t)g^*\left(\frac{t - \tau}{a}\right) dt \quad (2)$$

The distribution of the energy signal in the time-scale plane is given by the *Scalogram*, which is defined as the squared modulus of the CWT.

Fig. 2 illustrates the interpretation of time and frequency resolutions for the *Spectrogram* and *Scalogram* respectively. In the definition of the STFT (Fig. 2(a)) the same window length has to be chosen and used at all times so, every box has the same area determined by the *Heisenberg inequality* (also known as the Gabor limit for signal processing), lower bounded by  $1/4\pi$ . In CWT (Fig. 2(b)), the widths and heights of the boxes change, giving different proportions to time and frequency, even so the area remains constant. In contrast to the *spectrogram*, in this representation the energy of the signal is distributed with different resolutions.

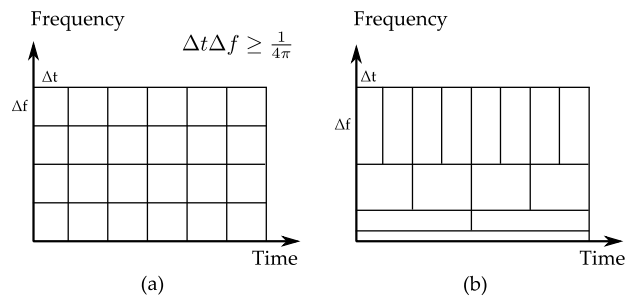


Fig. 2. Time-Frequency resolution for: a) Short-Time Fourier Transform (STFT) and b) Continuous Wavelet Transform (CWT).

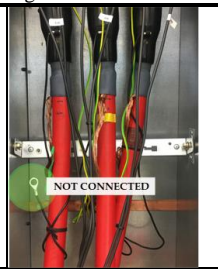

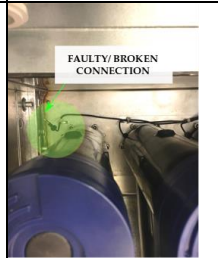

CWT is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies, i.e. higher frequencies are better resolved in time, and lower frequencies are better resolved in frequency. This kind of analysis is suitable if the signal is composed of high frequency components for a short duration and low frequency components for a long duration, which is often the case with signals encountered in practical applications, such as partial discharges.

## III. PD IN MV SUBSTATIONS ENVIRONMENT

One of the principle causes of partial discharge failures in substations are due to installation errors through rough handling or poorly assembled joints and terminations. It is more likely to find PD sources in cable accessories than in the cable itself, since the advances in material processing, manufacturing and factory test procedures assures a cable free from partial discharges.

It has been found that defective mountings, such as the loss of cable termination sealing caps, an earth cable in contact with cable terminator insulation, a poorly tightened cable termination, or a metallic part not linked to a defined potential, etc., causes partial discharges in the system. In this paper, four different sources are used and listed in Table I.

TABLE I  
PD sources found in MV switchgear.

<p><b>PDS1</b> <b>Cable termination floating earth.</b> This defect is caused when the earthing eye and ground lead of the tee-connector to the switchgear is not (or badly) connected to ground, thus generating a floating earth potential.</p>	
<p><b>PDS2</b> <b>Earth cable in contact with cable termination insulation.</b> If the earth cable does not respect electrical distance with the cable terminator insulation, partial discharges are produced. In this case, the defect source was located in a metering cubicle.</p>	
<p><b>PDS3</b> <b>VPIS Bushing screen disconnected</b> The Voltage Presence Indicating Systems (VPIS) is an indicator integrated into the cubicle that shows the presence of voltage in the phases through permanent light signals and is connected to the bushing screen of each phase. A faulty or broken connection will generate partial discharges.</p>	
<p><b>PDS4</b> <b>Earth grounding spring missing on busbar connector.</b> The link connection is used to complete the electrical connection between different switchgear cubicle modules. It maintains the rated insulation values, along with the rated and short-circuit currents. It also controls the electric field. Faulty or missing components of this link connection will generate PDs.</p>	

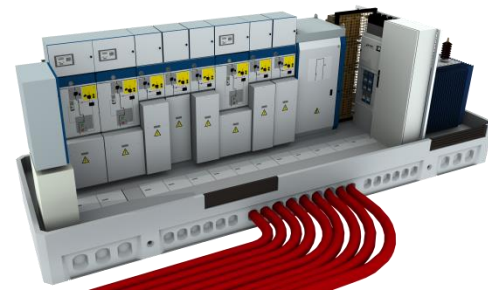


Fig. 3. UDEX substation equipped with modular cubicles.

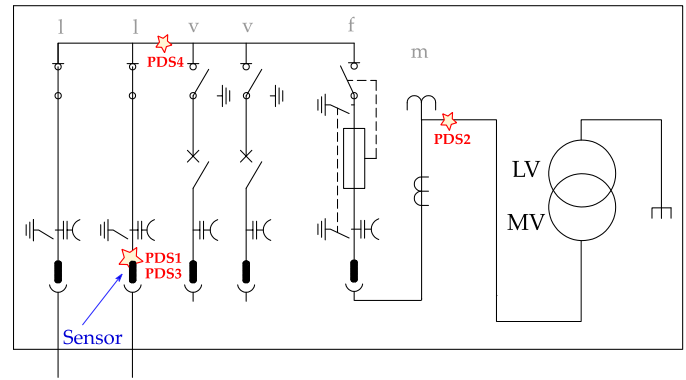


Fig. 4. Electrical connection diagram of the switchgear cubicles used for the experimental setup.

#### A. Experimental Setup

The experimental measurements used in this work have been carried out in Ormazabal's Smart Grid Laboratory, called UDEX. This is a highly configurable medium voltage network, whose single-line diagram with different line topologies that are possible to interconnect and other relevant information can be found in [15].

The four defect sources described in Table I are replicated in the UDEX laboratory, and partial discharge activity is detected using an on-line measurement technique. In Fig. 3 components of UDEX substations are shown. These substations include modular cubicles with feeder (l), fuse protection (f), circuit-breaker protection (v) and metering (m).

The electric scheme used for the test is illustrated in Fig. 4. Star symbols indicate the relative locations of the different PD sources. Sensors are located in the closest feeder cubicle to the PD source to avoid as much as possible the distortion of the pulse when it propagates in the circuit [16].

The measure of the PD activity is realized with a coupling capacitor (*Ormazabal ekor.EVT-C*) with a frequency bandwidth from 2 to 32MHz, which is installed directly into the T-junction cable end-plug within the switchgear cubicle.

The system is energized with high current flow, from the utility company. Even though measurements are made on the three phases, only the measurements from the phase corresponding to the PD source location are taken into account to build the database described in next section. These tests were performed at three different voltage levels for each PD source.

#### IV. DATA PROCESSING

The generated database includes real PD sources found in real MV substations, as shown in the previous section.

The construction of a large dataset is important to train a deep neural network and avoid bias. Usually this data is not available because of the complexity in performing repetitive tests and some authors [17]–[19] have used generative models [20] to create more data and complete their database.

##### A. Signal pre-processing

In order to build the dataset, raw pulses acquired by the sensor are required. The PD pulses have to be extracted from the raw data recorded by the instrument. The minimum time step that can be recorded with the considered PD instrument is 100ms, which correspond to five consecutive voltage cycles, sampled at 80 MHz. Fig. 5 shows an example of this data. The image represents pulses generated by source PDS1 at 25kV, showing a repetitive pattern after each AC voltage cycle.

The pre-processing step includes the extraction of an individual pulse from the raw data in a 6.4  $\mu$ s window length (512 samples), locating the maximum amplitude of each signal at 1.25  $\mu$ s. On the left side of Fig. 6, some examples of the PD signals recorded from different PD sources are shown.

Depending on the PD rate of each defect source, several individual pulses can be extracted from this raw data, which allows the generation of a large dataset.

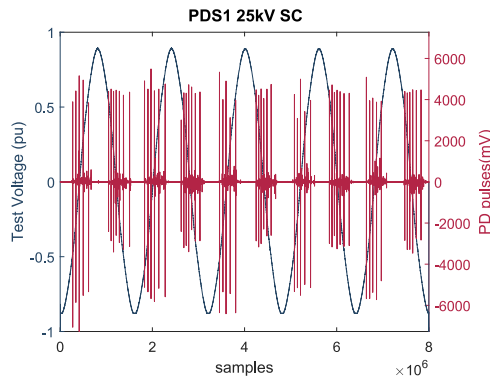


Fig. 5. Raw data recorded for PDS1 at 25kV by the instrument.

The PRPD pattern shown on the right side of Fig. 6, was used to analyze the PD sources. However, after several measurements at different voltage levels and time step recorded, it has been found that apart from the time needed to obtain the PRPD, the regularity of the pattern in the image depends on the PD development over time, the state of the insulation at the time of measurement, applied voltage level, recording time, the acquisition sensor used, the location of the sensor with respect to the PD source and the frequency integration range for the computation. These drawbacks make it difficult to build a large database with this type of representation. Therefore, the alternative *scalogram* image representation of the PD data is used in this paper, which optimize the time of data acquisition and guarantee a faithful representation of the PD phenomenon.

### B. Signal post-processing

Once the individual pulse time-window has been extracted, it is converted to its *scalogram* image representation.

To create the *scalograms*, a CWT filter bank is precomputed with Matlab, which is the preferred method when the CWT of many signals using the same parameters has to be obtained. The analytic *Morse* (3, 60) wavelet is used and the scalograms are converted to RGB images in pseudo-color, in which each image is an array of size 224x224x3 as can be seen in the images in the middle column of Fig. 6.

Besides the scalograms of PD pulses, the dataset also includes images from other high frequency signals, such as electromagnetic interferences and noises encountered in the MV network. Fig. 7 shows examples of their *scalograms*.

## V. CONVOLUTIONAL AUTOENCODER FOR PD IDENTIFICATION

The *Convolutional Autoencoder* (CAE) model combines the benefits of convolutional filtering in *Convolutional Neural Networks* (CNN) [21] with unsupervised pre-training of *Autoencoders* [22], allowing the training of convolutional layers independently from the classification task in order to generate a new data codification. This codification includes the most interesting comprised features from which it is possible to reconstruct the data. CAE model was firstly introduced by Masci et al. [23] and it has demonstrated that

pre-training a neural network using *Autoencoder weights* can improve the classification accuracy.

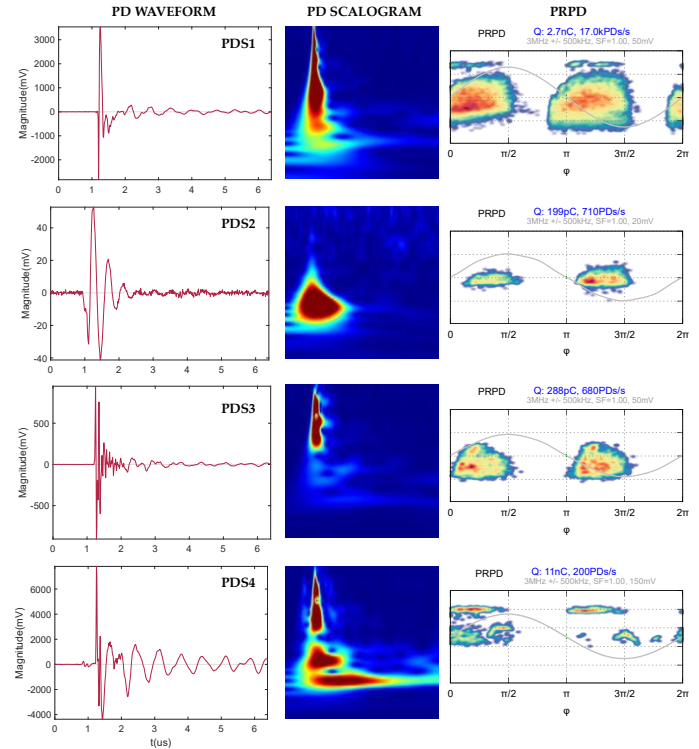


Fig. 6. Individual pulses, scalograms and PRPD representations for different PD sources.

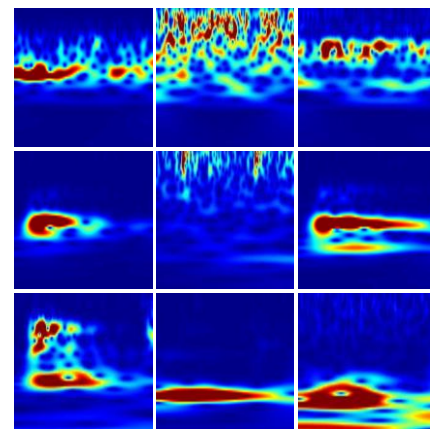


Fig. 7. Scalograms from Noises and other HF signals.

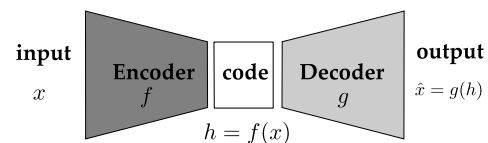


Fig. 8. Autoencoder Architecture.

An *Autoencoder* is divided in two main parts: An *Encoder* ( $f$ ) and a *Decoder* ( $g$ ) as illustrated in Fig. 8. The *Encoder* ( $f$ ) takes the original input  $x$  and maps it to the latent representation, called the *Latent Space* or code, where the



comprised representation of the input data is stored. Then, the *Decoder* ( $g$ ) has to reconstruct the original input  $x$  from the code ( $h$ ) to obtain the reconstruction  $\hat{x}$ .

Usual *Autoencoders* are composed of fully connected layers of neurons that ignore the 2D image structure. To tackle the dimensionality of 2D images, 2D convolutional layers in the encoding part replace dense layers. De-convolutional layers are implemented in the decoding part. The combination of both parts gives the resulting CAE model.

The benefit of using *Autoencoders* is that unlabeled data can be used to pre-train the feature extraction stage of the classifier. The *Encoder* extracts useful features, detecting and removing input redundancies and preserving only essential aspects of the data. It has been proven in [24] that this unsupervised initialization helps to avoid local minima and to increase the performance stability of the network.

### A. PD Diagnosis Methodology

The methodology to implement the CAE model for PD diagnosis is divided in three steps:

#### 1) Unsupervised Pre-Training of CAE - Step 1

This step is focused on the intermediate goal of learning a good data representation before beginning to work on the classification problem. The CAE is trained with unlabeled input data, and its optimal architecture has to be discovered by simulation trials.

The input for this network is the dataset containing PD scalogram images that represent several different PD sources, noises and other HF signals recorded over the years in the UDEX laboratory. They are all mixed, without labels, i.e. each image is just "labeled" by the image itself. For this reason, this kind of labeling is also called self-supervision.

In this step, as can be seen at the top of Fig. 9, the CAE tries to reconstruct the input image, learning the optimal filters that minimize the reconstruction error. However, this reconstructed image is not actually used; the aim of this step is that the network learns the relevant image information (*Feature Extraction*). Basic features, such as lines, edges, and corners appear in the initial layers, and more complex features are generated in the ascending layers as a result of combining basic features.

#### 2) Supervised Training of classifier – Step 2

The feature detectors of the latent codification that were automatically generated in the previous step can be used now to feed the classification stage. The main goal of this step is to classify the different PD sources.

In this second step, the *Decoder* of the previous model is removed from the network and a fully connected network (Multi-Layer Perceptron, MLP) is added on top of the *Encoder* stage. The training of the MLP, maintaining the *Encoder* stage unmodified, follows a supervised process with input images labeled with their correct class index. The bottom image in Fig. 9 illustrates the combined model.

The dataset used in this case is grouped according to the type of partial discharge source. In this work, due to computational limitations, only four different PD sources are chosen to be classified (PDS 1-2-3-4 described in Table I), but this methodology can be applied with more PD sources. A fifth class called NOTPDS is also included. This class is considered a rejection class where all images that do not correspond to a defect source are grouped (Noise and Other HF signals).

The output of the model is given by a *softmax* function, which estimates the probability  $P(y_k/x_i)$  that a given input  $x_i$  belongs to the class label  $y_k$  for  $k = \{1, 2, \dots, n\}$ , where  $n$  denotes the number of classes (5 in this case).

#### 3) Prediction with the Trained Model

Once trained, the complete model, the weights and the architecture are saved and used to predict a single image. The main parts of the procedure to predict the PD signal recorded by the PD instrument are:

- Read the PD raw data recorded by the PD instrument.
- Pre-process data: segment the time-window PD pulse.
- Post-process data: calculate scalogram of the signal.
- Load the full-trained model.
- Predict the PD source, showing the probabilities.

## VI. RESULTS

The architecture for the CAE model has to be selected for the unsupervised pre-training and for the supervised training. Experimental results are presented in following sections.

### A. Unsupervised pre-training

The optimal number of convolution layers needed for the encoding part has to be investigated. Through several simulations not shown, an *Encoder* with three layers was found to provide good results, so this architecture was selected for the implementation. The *Decoder* should have a similar or identical number of layers, trying to invert the convolutional process of the *Encoder*. It is an open design to explore. Hence, two approaches were explored: a) the use of *upsampling layers* combined with convolution layers, which is denominated as Architecture#1, and b) the use of *de-convolutional layers* (transposed convolution) denominated Architecture#2.

Fig. 10 shows both architectures. Both *Autoencoders* were designed in order to have approximately the same number of trainable parameters. Table II resumes the layer parameters of both architectures. Convolutional and deconvolutional layers

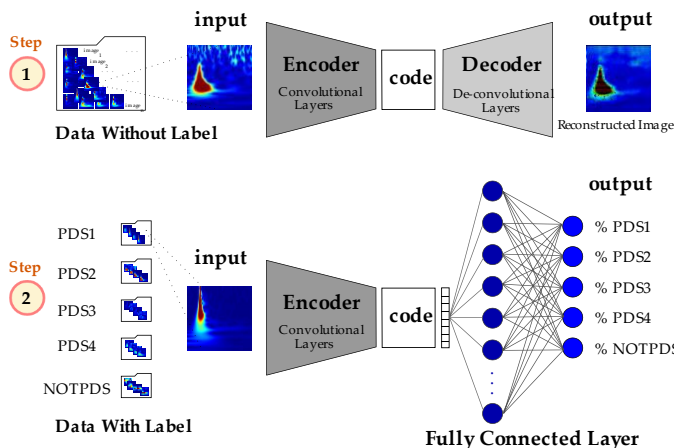


Fig. 9. Unsupervised Pre-Training of the CAE (top) and Supervised Training of the CNN (bottom).

used zero padding and SELU (*Scaled Exponential Linear Units*) as activation functions.

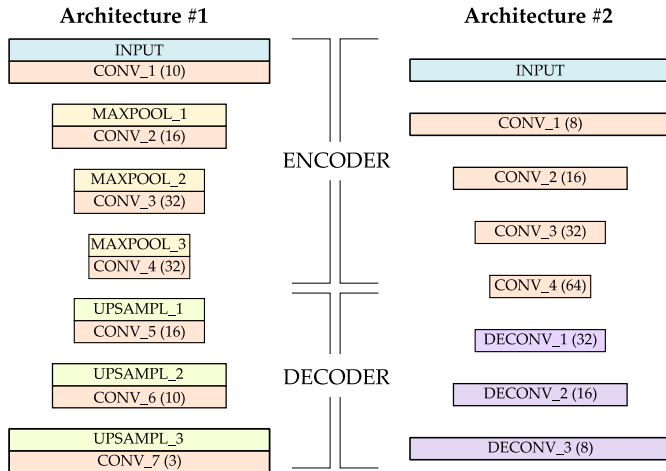


Fig. 10. CAE architectures compared in simulations. The number of filters of respective layers appear inside parenthesis.

TABLE II  
Details of CAE Architectures.

Layer	Architecture#1	Architecture#2
Convolution Kernel	3x3	2x2
Convolution Stride	1x1	2x2
MaxPooling	2x2	-
UpSampling	2x2	-
Deconvolution Kernel	-	2x2
Deconvolution Stride	-	2x2
<b>Parameters</b>		
Trainable weights	21971	21875

These architectures are implemented in *Python* using *Keras* deep learning library, with *Tensorflow* as backend. The machine on which the experiments have been run has an Intel Core i5-6200U CPU with 8GB RAM.

For training in this step, the dataset consists of 62367 RGB scalograms images (detailed in Table III). 80% of them are used for training, and the remaining 20% for validation.

TABLE III  
Detailed number of PD samples used for Step 1 and Step 2

Samples	PDS1	PDS2	PDS3	PDS4	OTHER PDS	NOTPDS
Step 1	15000	11850	12000	8190	9945	5382
Step 2	3524	3510	3573	3517	0	3530

The *Keras* class *ImageDataGenerator* was used to load the train and validation datasets, which is a suitable method when working with thousands of images that may not fit into system memory. Therefore, instead of loading all images into memory, only the data corresponding to a batch will be loaded during training time. A batch size of 128 was chosen.

The CAE models were firstly trained for many epochs, finding that 10 epochs give good results with regard to

computational time consumed and model performance. The loss function used is the Mean Squared Error with *Adam* optimizer [25]. Mean Absolute Error (MAE) is the metric evaluated by the model during training and validation.

Fig. 11. shows the synchronized training and validation loss evolutions along one simulation. Since validation losses followed decreasing curves, we can expect that both CAE models are not overfitting.

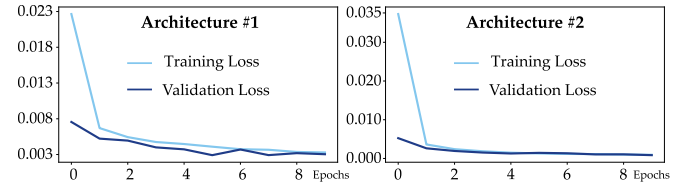


Fig. 11. Training and validation loss history for CAE models.

Comparative results for both architectures are listed in Table IV. The main difference is that training Architecture#1 takes 3.5 times longer than Architecture#2, and presents lower performance.

TABLE IV  
Comparative Results between Architectures.

	Architecture#1	Architecture#2
Loss Validation	0.0025	0.0008
MAE Validation	2.91	1.38
Training time	26.9 h	7.8 h

Fig. 12. illustrates the reconstructed images obtained with the two architectures for one example. Even though these images are not totally well reconstructed and checkboard artifacts [26] can be seen, the main goal of this step was to generate in the latent code relevant features of the images and not to reconstruct the image perfectly.

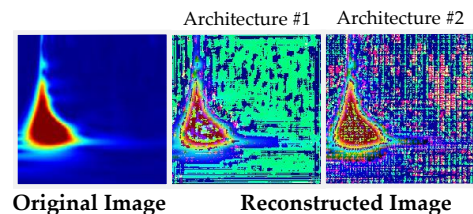


Fig. 12. Original and reconstructed images after 10 epoch training of CAE Architectures #1 and #2.

### B. Supervised training

As said before, in this step, the *Decoder* part of the previous architecture is removed and fully connected layers are stacked up with the pre-trained *Encoder* part. A flatten layer has to be added after the CONV\_4, with the aim of transforming its two-dimensional output tensor into a vector to be fed into a fully connected layer.

The number of fully connected layers and their number of neurons also has to be investigated. The output layer presents five neurons associated to the five classes of this problem, followed by a *Softmax* layer, which provides a probabilistic output distribution. In this study, classifiers without hidden

layer and architectures with one hidden layer are simulated. Simulations were made varying the number of hidden neurons from 8 to 20. As regularization method, a dropout layer with dropout rate of 50% is implemented between the flatten activation and the first dense layer, as this connectivity comprises the majority of the trainable weights in the classifier. The final architecture is illustrated in Fig. 13.

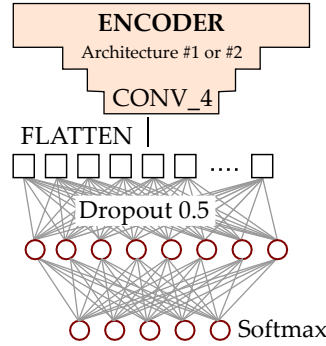


Fig. 13. Full model: encoder and classifier.

This second training allows verification that the previous architecture has learned meaningful information to achieve a good classification. The dataset consisted of 17654 scalogram images corresponding to PDS1-2-3-4 and NOTPDS (detailed in Table III). 80% of these images are used for training, 10% for validation and the remaining 10% for testing.

In order to compare performance of previous architectures (#1 and #2) for the classification task, 8 neurons are chosen in the hidden layer of Fig. 13. The performance of the networks was measured using Accuracy metric during training and validation over 10 epochs. The loss function used is the *Categorical Cross Entropy* and *Adam* optimization. Fig. 14 shows the synchronized training and validation loss evolutions during one simulation.

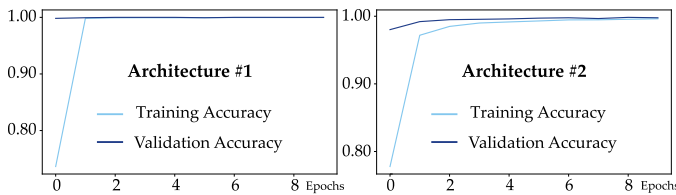


Fig. 14. Training and validation curve for fully connected layer with 8 neurons combined with architectures #1 and #2.

Table V summarizes the results for both architectures. Even though Architecture#1 is slightly more precise, it takes 6 times longer to train than Architecture#2. Therefore, the Architecture#2 is used to investigate the optimal number of neurons in the hidden layer.

The optimal number of neurons in the hidden layer was obtained by repeating the training 10 times for each number of hidden neurons. Multiple repetitions were performed in order to minimize the uncertainty rising from the random initialization of the network weights.

Results are shown as boxplots in Fig. 15, where zero neurons correspond to the case of the classifier without a hidden layer. It is clear that having a hidden layer does not significantly improve the result of using only the output layer,

and so to avoid overfitting, the chosen architecture should be the simplest.

TABLE V  
Results for fully connected layer with 8 neurons.

	Architecture#1	Architecture#2
Loss Test	0.00163	0.02106
Accuracy Test	100%	99.72%
Trainable parameters	125445	62725
Training time	60 min	10 min

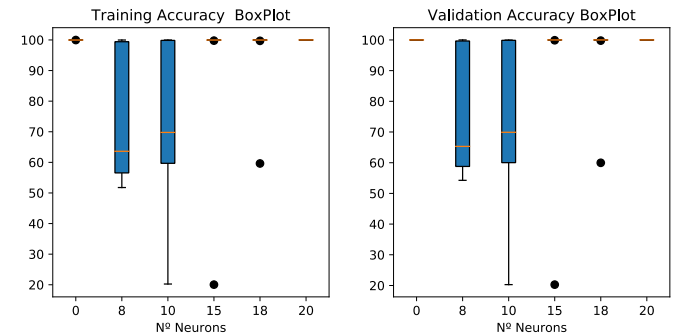


Fig. 15. Boxplot for training and validation Accuracy calculated for 10 repetitions for each number of neurons in the hidden layer.

The final model is tested with 10% of the dataset that was unseen for the training and validation process. Results are shown as a confusion matrix in Fig. 16, where none of the classes were misclassified, which validates the high performance of the chosen CAE architecture.

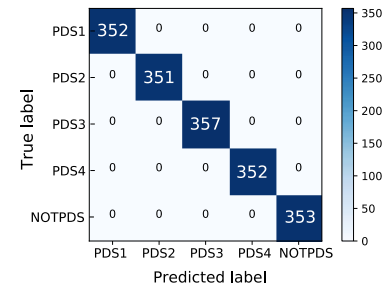


Fig. 16. Confusion matrix of final model.

Fig. 17 shows the comparison of two images of PDS2. The one on the right is noisier than the one on the left, as can be seen in the scalogram. As it can be noticed, despite the background noise, the model can predict the defect with an accuracy of approximately 87%.

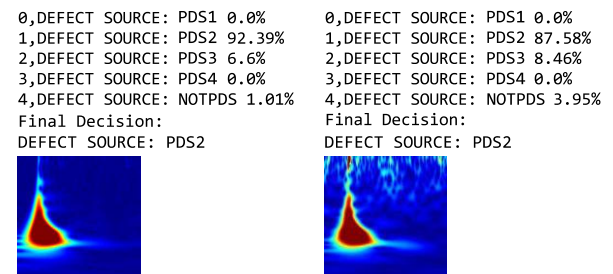


Fig. 17. Predictions with Architecture#2 and a fully connected layer.

## VII. CONCLUSIONS

In this paper, a new methodology for the automatic recognition of partial discharges was presented. The method has consisted of training a Deep Neural Network (DNN) model to predict the sources of partial discharge that are commonly found in MV switchgear. PD measurements were made in a real MV electrical network and the data were collected from four different PD sources, along with typical noise sources found in these types of installations. The database was built with scalogram representations, which are images obtained using the CWT of the PD signal.

The chosen DNN model was a Convolutional Autoencoder (CAE), which combines the benefits of convolutional filtering in Convolutional Neural Networks (CNN) with unsupervised pre-training of Autoencoders (AE). The optimal architecture was investigated through simulations summarized on section 6. It has been found that the use of deconvolutional layers instead of pooling layers in the decoder part reduces the training time and gives better performance results. Results have also shown that no hidden layer is need for the classifier.

The high prediction accuracy of the model has proven that this methodology is an effective tool to automate the recognition of PDs. However, its implementation in a large condition monitoring system has still to be developed.

## VIII. ACKNOWLEDGMENT

The authors would like to thanks to Ormazabal's UDEX team for all the support during the experiments.

## IX. REFERENCES

- [1] E. Gulski and F. H. Kreuger, "Recognition of discharge sources using statistical tools", in *[1991] Proceedings of the 3rd International Conference on Properties and Applications of Dielectric Materials*, Jul. 1991, pp. 39–42 vol.1.
- [2] N. C. Sahoo, M. M. A. Salama, and R. Bartnikas, "Trends in partial discharge pattern classification: a survey", *IEEE Trans. Dielectr. Electr. Insul.*, vol. 12, no. 2, pp. 248–264, Apr. 2005.
- [3] W. J. K. Raymond, H. A. Iliias, A. H. A. Bakar, and H. Mokhlis, "Partial discharge classifications: Review of recent progress", *Measurement*, vol. 68, pp. 164–181, May 2015.
- [4] S. Barrios, D. Buldain, M. P. Comech, I. Gilbert, and I. Orue, "Partial Discharge Classification Using Deep Learning Methods—Survey of Recent Progress", *Energies*, vol. 12, no. 13, p. 2485, Jun. 2019.
- [5] X. Peng, F. Yang, G. Wang, Y. Wu, L. Li, Z. Li, A. A. Bhatti, C. Zhou, D. M. Hepburn, A. J. Reid, M. D. Judd, W. H. Siew, "A Convolutional Neural Network Based Deep Learning Methodology for Recognition of Partial Discharge Patterns from High Voltage Cables", *IEEE Trans. Power Deliv.*, pp. 1–1, 2019.
- [6] L. Duan, J. Hu, G. Zhao, K. Chen, J. He, and S. Wang, "Identification of Partial Discharge Defects Based on Deep Learning Method", *IEEE Trans. Power Deliv.*, pp. 1–1, 2019.
- [7] H. Song, J. Dai, G. Sheng, and X. Jiang, "GIS partial discharge pattern recognition via deep convolutional neural network under complex data source", *IEEE Trans. Dielectr. Electr. Insul.*, vol. 25, no. 2, pp. 678–685, Apr. 2018.
- [8] M.-T. Nguyen, V.-H. Nguyen, S.-J. Yun, and Y.-H. Kim, "Recurrent Neural Network for Partial Discharge Diagnosis in Gas-Insulated Switchgear", *Energies*, vol. 11, no. 5, p. 1202, May 2018.
- [9] G. Li, M. Rong, X. Wang, X. Li, and Y. Li, "Partial discharge patterns recognition with deep Convolutional Neural Networks", in *2016 International Conference on Condition Monitoring and Diagnosis (CMD)*, Sep. 2016, pp. 324–327.
- [10] G. Li, X. Wang, X. Li, A. Yang, and M. Rong, "Partial Discharge Recognition with a Multi-Resolution Convolutional Neural Network", *Sensors*, vol. 18, no. 10, p. 3512, Oct. 2018.

- [11] B. Adam and S. Tenbohlen, "Classification of multiple PD Sources by Signal Features and LSTM Networks", in *2018 IEEE International Conference on High Voltage Engineering and Application (ICHVE)*, Athens, Greece, Sep. 2018, pp. 1–4.
- [12] K. Banno, Y. Nakamura, Y. Fujii, and T. Takano, "Partial Discharge Source Classification for Switchgears with Transient Earth Voltage Sensor Using Convolutional Neural Network", in *2018 Condition Monitoring and Diagnosis (CMD)*, Sep. 2018, pp. 1–5.
- [13] X. Wan, H. Song, L. Luo, Z. Li, G. Sheng, and X. Jiang, "Pattern Recognition of Partial Discharge Image Based on One-dimensional Convolutional Neural Network", in *2018 Condition Monitoring and Diagnosis (CMD)*, Sep. 2018, pp. 1–4.
- [14] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis", *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.
- [15] I. Gilbert, P. Mulroy, A. Hurtado, N. Akroud, and I. Orue, "Practical experience of a partial discharge monitoring application on an experimentation mv distribution network", in *Proceedings International Conf. on Electricity Distribution (CIRED)*, 2015, p. 4.
- [16] P. Wagenaars, P. A. A. F. Wouters, P. C. J. M. van der Wielen, and E. F. Steennis, "Influence of Ring Main Units and Substations on Online Partial-Discharge Detection and Location in Medium-Voltage Cable Networks", *IEEE Trans. Power Deliv.*, vol. 26, no. 2, Apr. 2011.
- [17] W. Yijiang, L. Chen, W. Ganjun, P. Xiaosheng, L. Taiwei, and Z. Yunzheng, "Partial Discharge Data Augmentation of High Voltage Cables based on the Variable Noise Superposition and Generative Adversarial Network", in *2018 International Conference on Power System Technology (POWERCON)*, Nov. 2018, pp. 3855–3859.
- [18] X. Wang, H. Huang, Y. Hu, and Y. Yang, "Partial Discharge Pattern Recognition with Data Augmentation based on Generative Adversarial Networks", in *2018 Condition Monitoring and Diagnosis (CMD)*, Sep. 2018, pp. 1–4.
- [19] J. A. Ardila-Rey, J. E. Ortiz, W. Creixell, F. Muhammad-Sukki, and N. A. Bani, "Artificial Generation of Partial Discharge Sources Through an Algorithm Based on Deep Convolutional Generative Adversarial Networks", *IEEE Access*, vol. 8, pp. 24561–24575, 2020.
- [20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative Adversarial Nets", in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.
- [21] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision", in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, France, May 2010, pp. 253–256.
- [22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders", in *Proceedings of the 25th international conference on Machine learning - ICML '08*, Helsinki, Finland, 2008.
- [23] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction", in *Artificial Neural Networks and Machine Learning – ICANN 2011*, Berlin, Heidelberg, 2011, pp. 52–59.
- [24] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does Unsupervised Pre-training Help Deep Learning?", *J. Mach. Learn. Res.*, pp. 625–660, 2010.
- [25] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", *ArXiv14126980 Cs*, Jan. 2017, Accessed: Mar. 09, 2020. [Online]. Available: <http://arxiv.org/abs/1412.6980>.
- [26] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and Checkerboard Artifacts", *Distill*, vol. 1, no. 10, p. e3, Oct. 2016.