Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione

Curriculum: Elettronica, Elettromagnetismo ed Elettrotecnica

Detection and Classification of Power Quality Disturbances using Machine Learning Algorithms

Candidate
Carlos Iturrino-García

Supervisor
Prof. Francesco Grasso

PhD Coordinator
Prof. Fabio Schoen

ciclo XXXV, 2019-2022

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione (DINFO).

Acknowledgments

I would like to acknowledge the efforts and input of my supervisor, Prof. Francesco Grasso, and all my colleagues of Smart Energy Lab who were of great help during my research. In particular my thanks go to Libero Paolucci, Prof. Antonio Luchetta, Marco Bindi, Alessandro Bartolini and Fabio Corti who collaborated on the main parts of my research work.

Contents

Chapter 1

Introduction

Society has become highly dependent on electric energy for school, work, transportation, communication and even for leisure in the modern world. Technology has evolved in such a way that more and more devices and instruments used in everyday life have become energy dependent. These include: phones, computers, washing machines, water pumps, televisions, etc. Also, transportation in the near future will be completely electric. This has brought more generation with the available resources i.e. Coal, Oil and Gas. Ideally, grid voltages and currents should have a purely sinusoidal waveform, but in reality they come in a distorted manner. These distortions vary in form and magnitude and can sometimes be a combination of different factors. These disturbances can have multiple sources that can affect multiple users in a community. These can be caused by the use of nonlinear electronic loads or by the generation of power using renewable sources. Due to the ongoing global pollution, energy generation has taken steps to mitigate it's negative effects on climate. This has brought an increasing interest in technology for renewable generation of energy. Recently, renewable energy sources have engrossed great tendency because of their potential to solve problems like increasing the need for electrical power, air pollution, and global warming. Both the wind energy and solar photovoltaic energy, among others systems, are combined to form the hybrid power system network to provide future energy demand. The properties of these sources, like wind variations and solar insolation changes, have a significant influence on power quality (PQ), reliability, and safety. Consequently, the low power quality leads to motor failure, overheating of the lines, inaccurate metering, prema-

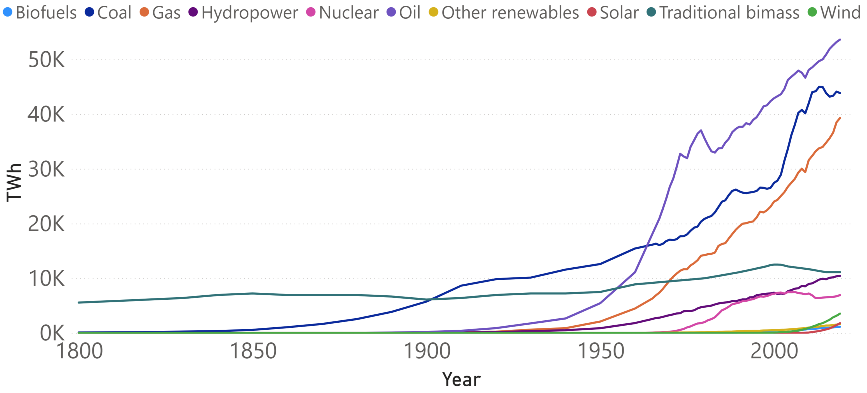


Figure 1.1: Global Energy Production by Source

ture aging of devices, and disturbances in communication circuits [1]. As shown in Figure 1.1, energy production using Oil, Coal and Gas dominate the current energy market. In recent years, an increase in renewable energy generation has increased which should mitigate climate change. As said before, the problem with these technologies is that they are highly non-linear and brings non-linearities into the power grid. In order to keep up with the proper functioning of these instruments, energy suppliers and consumers must keep up with the energy requirements for the proper functioning of these devices.

Other than renewable energy sources, PQ problems can arise from the use of electronic devices and appliances that bring severe problems to grid voltages and currents in the form of Power Quality Disturbances (PQDś). In recent years, a large number of nonlinear loads and distributed generations with random characteristics are connected to the power grid [2]. Although their extensive use in industrial, household, commercial and public sectors have improved many aspects of everyday life, they have brought negative effects on the power grid. The use of nonlinear electronic loads has increased the eventuality of unbalanced currents, unacceptable harmonic levels, and poor power factor in three-phase distribution systems [3]. In other words, power electronics technologies and/or nonlinear loads have made life easier and more comfortable but due to their nonlinear behavior it disturbs the power grid through voltage and current waveform distortions. As a consequence to the extensive use of nonlinear electronic devices, the purely sinusoidal waveform gets injected with distorting components in an increasing rate which have degraded PQ levels. If distorting components are injected, power losses and malfunctioning of electric devices can occur. Common effects of a degradation of power quality in the industrial sector include: loss of production, manufacturing interruption, loss of revenue, productivity cost, decrease competitiveness, lost opportunity, wasted energy, and the decrease of equipment life [4]. Similar effects can occur in the residential sector as overheating of AC appliances, TV screens displaying flicker and data loss, malfunctioning communication equipment and computer failures [5]. Nonlinear electronic loads can also cause disturbance to other consumers and interference in nearby communication networks [6].

Power Quality (PQ) is defined by the IEEE as " The concept of powering and grounding sensitive equipment in a matter that is suitable to the operation of that equipment" [7]. Any deviation, in voltage and current, from its nominal values in a certain period of time is considered a PQD. PQDs are classified as a deviation from its nominal magnitude and/or frequency components for a certain duration in time. Table 1.1 shows the PQDs characteristics as shown in the IEEE Recommended Practice for Monitoring Electric Power Quality [8].

In the last decades, PQ research has grown. Figure 4.6 shows the number of articles published by year as indexed by scopus. It shows a growing trend and it is expected to grow in the upcoming years. These statistics show the importance, usefulness, and vast quantities of research opportunities within this field of study.

Categories	Typical Spectral Content	Typical Durations	Typical Voltage Magnitudes
1 Transients			
1.1 Impulsive			
1.1.1 Nanoseconds	5ns rise	50ns	
1.1.2 Microseconds	1 s rise	50ns→1ms	
1.1.3 Miliseconds	0.1ms rise	1ms	
1.2 Oscillatory			

1.2.1 Low Frequency	5kHz	0.3→50ms	0→4
1.2.2 Medium Frequency	5→500 kHz	20 μs	0→8
1.2.3 High Frequency	0.5→ 5 MHz	5 μs	0→4
2 Short→duration rms variations			
2.1 Instantaneous			
2.1.1 sag		0.5→30 cycles	0.1→0.9
2.1.2 swell		0.5→30 cycles	1.1→1.8
2.2 Momentary			
2.2.1 Interruption		0.5 cycles→3 s	<0.1
2.2.2 Sag		0.5 cycles→3s	0.1→0.9
2.2.3 swell		0.5 cycles→3 s	1.1→1.4
2.2.4 Voltage Imbalance		0.5 cycles→3 s	2%→15%
2.3 Temporary			
2.3.1 Interruption		3 s → 1 min	<0.1
2.3.2 Sag		3 s → 1 min	0.1→0.9
2.3.3 swell		3 s → 1 min	1.1→1.2
2.3.4 Voltage Imbalance		3 s → 1 min	2%→15%
3 Long duration rms variations			
3.1 Interruption Sustained		1 min	0
3.2 Undervoltages		1 min	0.8→0.9
3.3 Overvoltages		1 min	1.1→1.2
3.4 Current Overload		1 min	
4 Imbalance			
4.1 Voltage		Steady State	0.5%→5%
4.2 Current		Steady State	1.0%→3.0%
5 Wave Distortion			
5.1 DC Offset		Steady State	0→0.1%
5.2 Harmonics	0→9 kHz	Steady State	0→20%
5.3 Interharmonics	0→9 kHz	Steady State	0→2%
5.4 Notching		Steady State	
5.5 Noise	broadband	Steady State	0→1%
6 Voltage Fluctuations	25 Hz	intermitent	0.1→7% 0.2→2
7 Power frequency variations		10 s	0.10 Hz

Table 1.1: IEEE 1159.

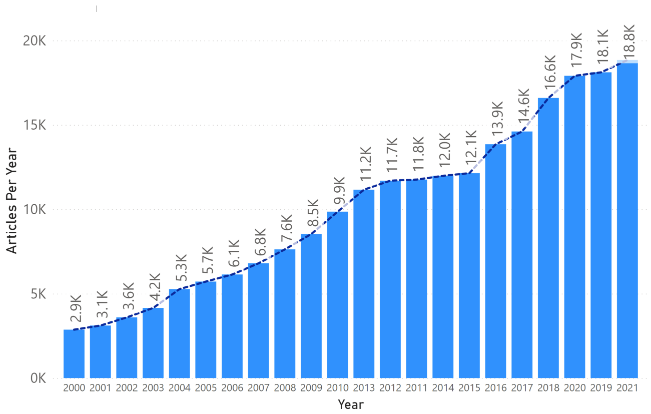


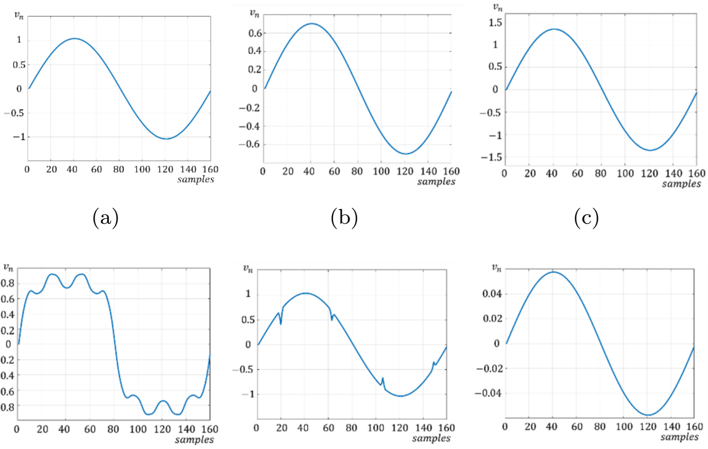
Figure 1.2: Articles published on a 21 year span

NOTE→These terms and categories apply to power quality measurements and are not to be confused with similar terms defined in IEEE Std 1366→2012 [B30] and other reliability→related standards, recommended practices, and guides.
a The quantity pu refers to per unit, which is dimensionless. The quantity 1.0 pu corresponds to 100%. The nominal condition is often considered to be 1.0 pu. In this table, the nominal peak value is used as the base for transients and the nominal rms value is used as the base for rms variations.
b Flicker severity index Pst as defined in IEC 61000→4→15:2010 [B17] and IEEE Std 1453 [B31].

Many instruments exist today that measure Power Quality (PQ). A comparison is made with the most common instruments used today shown in in Table 1.2. All these instruments have there pros and cons.

Table 1.2 Power Quality Measuring Instruments.

Manufacturer	Model	Cost	Sampling Freq	Country
Fluke	434-II	6.949,00 eu	200 kHz	USA
Dewesoft	Sirius XHS	100	15 MHz	Slovenia
HT Instruments	PQA820	2.059,00 eu	6.4 kHz	Italy
Hioki	PQ3198	6.897,00 eu	200kHz	Japan



(a) (b) (c)

(d) (e) (f)

Figure 1.3: (a) Normal (b) Sag (c) Swell (d) Harmonics (e) Notch (f) Interruption

These instruments are very effective at measuring voltage and current signals and usually have a high sampling rate making them effective at capturing high frequency components injected into the grid. The problem is that these instruments come at a high cost and, with it, they require trained professional to operate them. They also fail at recognizing PQDs.

1.1 Power Quality Disturbance

Power Quality Disturbances are deviations of voltage and current levels from their nominal values which can be considered as a wide spectrum of anomalies that can be considered as PQD. These deviations can be classified as magnitude deviations and frequency deviations.

The magnitude deviations are changes in there voltage levels from the accepted values. These deviations are the Sag, Swell and the interruption which are shown in Figure 1.3 b, c and f. On the other hand, frequency deviations are frequency components that are added to the supply waveforms that alter the functioning of given equipment. These include the Harmonics and the Notch which are shown in Figure 1.3 d and e.

1.1.1 Magnitude Deviations

Magnitude deviations includes deviations from the nominal working voltage levels. These include over voltage and under voltage. Over voltages include voltages over 10% of its nominal working value. Under voltages include voltage levels under 10% of its nominal values. Figure 1.4 shows a plot with a 50 Hz voltage signal where the red shaded area depicts where it is considered under voltage and the blue regions are considered over voltage.

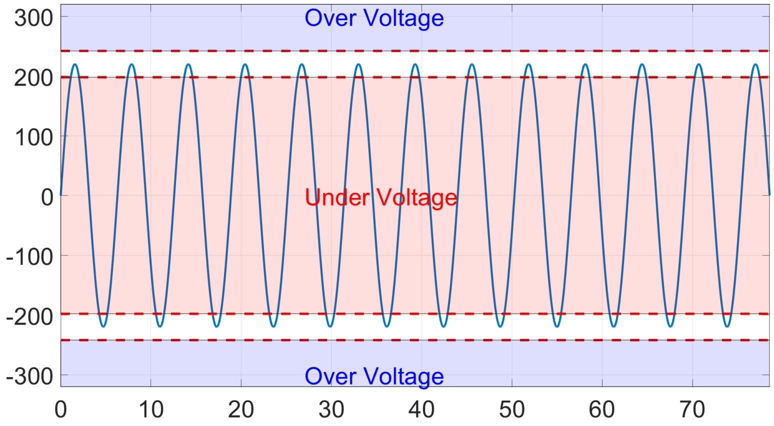


Figure 1.4: Magnitud deviation, under and over voltage

Voltages are 90% under its nominal value they are considered interruptions. Figure 1.5 shows a plot with a 50 Hz voltage signal where the red shaded area depicts the area where an interruption is considered.

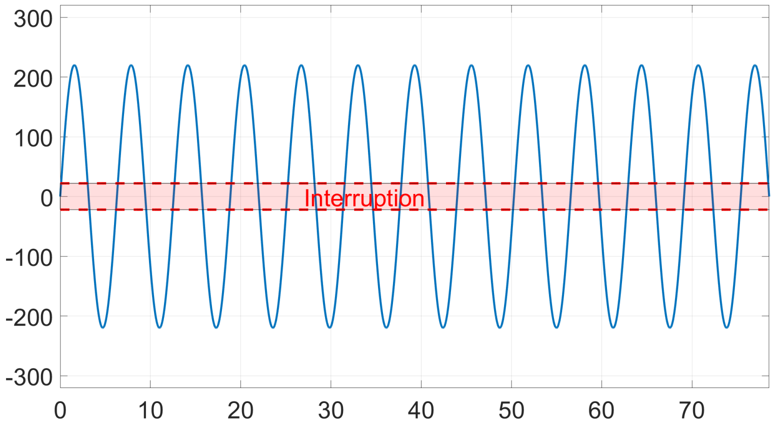


Figure 1.5: Magnitud deviation, interruption

1.1.2 Frequency Deviation

Frequency deviations on a voltage and current supply involve the injection of harmonic components injected to the grid. Harmonics are defined as waveforms at multiples of the fundamental input signals frequency. For the 50Hz AC waveform the 2nd harmonic is at 100Hz the 3rd harmonic is at 150Hz and so on. The magnitude of each frequency component has to be in such a way that the Total Harmonic Distortion (THD) does not surpass 8%. The THD of the voltage supply can be calculated using Equation 1.1. Similarly the THD for the current supply can be calculated using the same formula.

$$THD = \frac{\sqrt{\sum_{n=2}^{\inf} V_{nrms}^2}}{V_{1rms}} \tag{1.1}$$

As said in [9] The harmonics in which a signal can be decomposed, as shown before, are entire multiples of the fundamental. A part from the distinction between Odd (symmetrical) and Even (asymmetrical), they can be classified according its phase rotation with the fundamental:

- Positive sequence harmonics (4th, 7th, 10th,...): they do have the same phase rotation than the fundamental, and circulate between phases.

Table 1.3 Voltage Distortion Limits.

Bus Voltage V at PCC	**Individual harmonic (%)**	**Total Harmonic Distortion THD (%)**
$V \leq 1.0kV$	5.0	8.0
$1.0kV < V \leq 69kV$	3.0	5.0
$69kV < V \leq 161kV$	1.5	2.5
$161kV \leq V$	1.0	1.5

- Negative sequence harmonics (2nd, 5th, 8th,...): they have the opposite phase rotation than the fundamental, and circulate between phases.

- Zero sequence harmonics (3rd, 6th, 9th, ...), also known as Triplen harmonics: these harmonics are on phase with the fundamental, and circulate between phases and neutral. What is the same, they do not cancel and add up directly in the Neutral conductor. For that reason, in the presence of significant components of such harmonics in a 3 phase installation, Neutral conductor will need to be oversized (compared to phase conductors) to carry out these extra-current. For example, in the presence of around 10 Arms 3rd order harmonic in each of the 3 phases, will mean an extra current of around 30 Arms in Neutral conductor (at 150Hz in a 50Hz installation), like in Fig.8. If no triplen harmonics were present, Neutral current will carry no current (no 50Hz component present in Fig.8).

In theory, the supply voltage should be in a purely sinosoidal manner. In practice harmonic components will always be existent in the supply voltage due to many factors. Limits are drawn in order to ensure the proper functioning of electric and electronic loads. This limits are shown in Table 1.3. Table 1.3 shows the individual harmonic amplitude percentage with respect to the nominal working voltage of the voltage supply and the total harmonic distortion limit calculated using Equation 1.1. Table 1.3 is the IEEE 519 Voltage Distortion Limit standard [10].

The standard also shows the current limits for proper functioning of electric and electronic loads. As also shown in [10], the limits in this subclause apply to users connected to systems where the rated voltage at

Table 1.4 Current distortion limits for systems rated 120 V through 69 kV.

Maximum harmonic limits for systems rated 120V through 69kV						
	Individual harmonic order (odd harmonics)					
I_{SC}/I_L	$3 \leq 11$	$11 \leq 17$	$17 \leq 23$	$23 \leq 35$	$35 \leq 50$	TDD
$\leq 20^c$	4.0	2.0	1.5	0.6	0.3	5.0
$20 \leq 50$	7.0	3.5	2.5	1.0	0.5	8.0
$50 \leq 100$	10.0	4.5	4.0	1.5	0.7	12.0
$100 \leq 1000$	12.0	5.5	5.0	2.0	1.0	15.0
≥ 1000	15.0	7.0	6.0	2.5	1.4	20.0

the PCC is 120 V to 69 kV. At the PCC, users should limit their harmonic currents as follows:

- Daily 99th percentile very short time (3 s) harmonic currents should be less than 2.0 times the values given in 1.4.
- Weekly 99th percentile short time (10 min) harmonic currents should be less than 1.5 times the values given in 1.4.
- Weekly 95th percentile short time (10 min) harmonic currents should be less than the values given in 1.4.

1.2 The objective

PQDs are mainly expressed in magnitude and or frequency deviation from its nominal values in voltage and current signals. Do to the fact of the nature of the voltage signals, Machine Learning and Deep Learning algorithms are able to solve this kind of problems. The objective of this work is to explore, develop and implement machine learning and or deep learning algorithms for detection and classification of PQDs. A dataset needs to be created in order to train and compare different Deep Learning algorithms. The dataset is to be generated using Matlab Simulink. A comparison is necessary in order to determine the best one for the task. This comparison includes well→known deep learning architectures porven to be effective at other tasks. After the comparison, a new state of the art architecture called the Single Shot Power Quality Disturbance Detector SSPQDD is to be developed and tested. This architecture is designed specifically to tackle PQDs detection and classification in a real time environment. The experimental procedure and results are to be presented in order to show the effectiveness and performance of this new architecture. This algorithm is to be implemented using embedded electronics in a real time environment.

1.3 Contributions

The main contribution of this research is to design test and implement a deep learning architecture that can detect and classify PQDs using measured data from embedded electronics. This work presents a comparison of state of the art deep learning architectures for classification of PQDs. Also, this work presents the design procedure that for a state of the art deep learning architecture specifically design for detection and classification of PQDs in a real time environment using embedded electronics. Whith that, the architectures limitations and performance at detection and classification of PQDs. That is, its precision, recall, F1-score and the AUC of the ROC and compare them with previous architectures. In addition to designing and testing a state of the art architecture for detection and classification of PQDs, the architecture is implemented using embedded electronics. The measuring device created measures voltages and currents at 16kHz and detects and classifies PQDs in a real time environment. This state of the art architecture has higher precision and lower computational resource requirements than traditional architectures used for PQDs classification. Above all it is able to identify the position and duration disturbances and classify them.

Chapter 2

Literature review

This chapter gives a survey of related work on Power Quality Disturbances and Power Quality event detection and classification using feature extraction algorithms, machine learning algorithms and deep learning algorithms. The first part of the chapter introduces the term power quality and power quality disturbances, while the second part summarizes the power quality disturbances detection algorithms. The third section of the chapter describes the machine learning algorithms used for detection and classification. Going further still, the fourth section explains the deep learning algorithms used in recent work. Finally, object detection algorithms are presented and studied in the final section due to the importance of these algorithms in this thesis.

2.1 Power Quality

As said in Chapter 1, the definition PQ given by the IEEE is: the concept of powering and grounding of sensitive equipment in a matter that is suitable to the operation of that equipment. In other words, PQ involves power levels supplied and consumed at its nominal values. [11] defines PQ as the following.

- *Voltage quality* is concerned with deviations of the voltage from the ideal. The ideal voltage is a single-frequency sine wave of constant amplitude and frequency.

- *Current quality* is the complementary term to voltage quality: it is concerned with the deviation of the current from the ideal. The ideal current is again a single-frequency sine wave of constant amplitude and frequency, with the additional requirement that the current sine wave is in phase with the voltage sine wave.
- *Power quality* is the combination of voltage quality and current quality.
- *Quality of supply* is a combination of voltage quality and the non-technical aspects of the interaction from the power network to its customers
- *Quality of consumption* is the complementary term to quality of supply.

This can me summarize as a deviation from magnitud and or frequency in voltage and current measurements.

2.2 Power Quality Disturbance Detection Algorithms

Many attempts have been made to detect and classify PQDs in a real time environment. Fourier transform (FT) [12] [13], short-time Fourier transform (STFT) [14] [15], wavelet transform (WT) [16] [17] [18], s- transform (ST) [19] [20] [21] [22] [23], Hilbert Huang transform (HHT) [24] [25].

2.2.1 Fourier Transform

The Fourier Transform is a frequency analysis done to a time signal in order to extract its frequency characteristics. Naturally a frequency analysis can be done to a voltage and current signal which shows magnitude and frequency deviations hidden to the naked eye. The Fourier Transform is shown in Equation 2.1

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \tag{2.1}$$

Due to the nature of the processing, it is necessary for discretization of the signal. To gather the spectral content of the discrete signal. In order to be successful the Discrete Fourier Transform is performed. The discrete Fourier transform (DFT) is a transform in its own right such as the Fourier integral transform or the Fourier series transform. It is a powerful reversible mapping operation for time series. As the name implies, it has mathematical properties that are entirely analogous to those of the Fourier integral transform. In particular, it defines a spectrum of a time series; multiplication of the transform of two time series corresponds to convolving the time series [26]. The DFTis shown in Equation 2.2.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi}{N}kn} \tag{2.2}$$

In order to accelerate the computation of the DFT, an optimized version was developed called the Fast Fourier Transform (FFT). The FFT iterates on the array of given complex Fourier amplitudes and yields the result in less than $2NIog_2N$ operations without requiring more data storage than is required for the given array A [27].

In [28] the objective was to present unique features that characterize power quality events and methodologies to extract them from recorded voltage and/or current waveforms using Fourier and wavelet transforms. They utilized these identifying features to build an event identification module. The identification module extracts relevant identifying features.

[29] presents fractional Fourier transform (FRFT) based feature extraction a technique for classification of PQDs. FRFT is a generalized version of Fourier transform (FT) with an additional order control and can give time, frequency and intermediate time-frequency representations for a signal. The proposed technique shows better performance in most of the cases, when compared with Stockwell transform based classification under similar conditions. Further, a validation using real PQDs obtained from an experimental setup is shown. The corresponding results closely resemble the simulation outcomes.

The paper [30] discusses the application of the windowed fast Fourier transform to electric power quality assessment. The windowed FFT is a time windowed version of the discrete time Fourier transform. The window width may be adjusted and shifted to scan through large volumes of power quality data. The value of the WFFT is the rapid evaluation of data when no problem is detected, yet the ability to focus on detected power quality problems. There is an error introduced by windowing: this error is governed by a sinc function multiplier of the desired signal spectrum. It is possible to adjust parameters of the sinc function multiplier to control calculated spectrum error.

2.2.2 Short-Time Fourier Transform

The Short-Time Fourier Transform (STFT) is characterized by a Fourier transform executed in a fixed windowed interval. The window function $g(n)$ is called Blackmans window function and it is used to multiply a short segment of the signal by the window function. This avoids sharp sections and redundant information. A Fourier transform of this small windowed section $X_n(j\omega)$ is calculated and stacked up to form a matrix. The STFT equation is shown in Equation 2.3 and the Blackman Window Function is shown in Equation 2.4.

$$X(j\omega) = \sum x[n]g[n - mR]e^{-2\pi fn} \tag{2.3}$$

$$w[n] = a_0 + a_1 cos\left[\frac{2\pi n}{N}\right] + a_2 cos\left[\frac{4\pi n}{N}\right] \tag{2.4}$$

The paper [31] deals with the comparison of signal processing tools for power quality analysis. Two signal processing techniques are considered: the wavelet filters and the discrete short-time Fourier transforms. Then, examples of the two most frequent disturbances met in the power system are chosen. It is designed an adjustable speed drive with a six-pulse converter using EMTP/ATP and it is presented normal energizing of utility capacitors. The analysis is tested on a system consists of 13 buses and is representative of a medium-sized industrial plant. Finally, each kind of electrical disturbance is analyzed with example representing each tool. A qualitative comparison of results shows the advantages and drawbacks of each signal processing technique applied to power quality analysis. Since the center frequencies of the band-pass filters associated with the discrete STFT can be freely chosen, e.g. at the harmonics of 60 Hz, discrete STFT is more suitable for harmonic analysis. By selecting a small window length, STFT is able to detect transient positions in disturbance data.

In [32] the advantages and disadvantages of STFT and wavelet transform are compared which are used to analyze signal of transient harmonic in power system, combining their merits, a novel STFT based on the special frequency band is proposed in this paper. Multi-resolution analysis is used to detect the fault time and determine the main frequency range of transient signal. Then the central point and proper width of window of STFT can be obtained.

Consequently the accuracy and efficiency of STFT can be improved greatly, frequency components and its amplitudes of transient signal can be obtained accurately and quickly by this method.

2.2.3 Wavelet Transform

Wavelet analysis is an exciting new method for solving difficult problems in mathematics, physics, and engineering, with modern applications as diverse as wave propagation, data compression, signal processing, image processing, pattern recognition, computer graphics, the detection of aircraft and submarines and other medical image technology. Wavelets allow complex information such as music, speech, images and patterns to be decomposed into elementary forms at different positions and scales and subsequently reconstructed with high precision [33]. Equation 2.5 shows the Continuous Wavelet Transform (CWT) where the $x(t)$ is the input signal and the $\psi\left(\frac{t-b}{a}\right)$ is the wavelet function.

$$T(a,b) = \frac{1}{\sqrt{a}} \int x(t)\psi\Big(\frac{t-b}{a}\Big)dt \tag{2.5}$$

The wavelet function can be many predefined function like:Mexican Hat Wavelet 2.6, Morlet Wavelet 2.7, Complex Morlet wavelet 2.8, Gaussian Derivative Wavelet 2.9, Complex Gaussian Derivative Wavelet 2.10, Shannon Wavelet 2.11 and Frequency B-spline Wavelet 2.12.

$$\psi(t) = \frac{2}{\sqrt{3}\sqrt[4]{\pi}} e^{-\frac{t^2}{2}}(1-t^2) \tag{2.6}$$

$$\psi(t) = e^{-\frac{t^2}{2}} cos(5t) \tag{2.7}$$

$$\psi(t) = \frac{1}{\sqrt{\pi B}} e^{-\frac{t^2}{B}} e^{j2\pi Ct} \tag{2.8}$$

$$\psi(t) = Ce^{-t^2} \tag{2.9}$$

$$\psi(t) = Ce^{-jt}e^{-t^2} \tag{2.10}$$

$$\psi(t) = \sqrt{B}\frac{sin(\pi Bt)}{\pi Bt} e^{j2\pi Ct} \tag{2.11}$$

$$\psi(t) = \sqrt{B}\Big[\frac{sin(\pi B\frac{t}{M})}{\pi B\frac{t}{M}}\Big]^{M} e^{j2\pi Ct} \tag{2.12}$$

In [34], the paper presents a new approach to detect, localize, and investigate the feasibility of classifying various types of power quality disturbances. The approach is based on wavelet transform analysis, particularly the dyadic-orthonormal wavelet transform. The key idea underlying the approach is to decompose a given disturbance signal into other signals which represent a smoothed version and a detailed version of the original signal. The decomposition is performed using multiresolution signal decomposition techniques. They demonstrate and test their proposed technique to detect and localize disturbances with actual power line disturbances. In order to enhance the detection outcomes, they utilize the squared wavelet transform coefficients of the analyzed power line signal. Based on the results of the detection and localization, they carry out an initial investigation of the ability to uniquely characterize various types of power quality disturbances. This investigation is based on characterizing the uniqueness of the squared wavelet transform coefficients for each power quality disturbance.

2.2.4 S-Transform

S-transform (ST) is an effective method to analyze power quality in time and frequency domains. The S transform [6] is an extension to the ideas of wavelet transform, and is based on a moving and scalable localizing Gaussian window and has characteristics superior to either of the transforms. The S transform is fully convertible from the time domain to two-dimensional 2D frequency translation domain and to then familiar Fourier frequency domain. The amplitude frequency time spectrum and the phase frequency time spectrum are both useful in defining local spectral characteristics. The superior properties of the S transform are due to the fact that the modulating sinusoids are fixed with respect to the time axis while the localizing scalable Gaussian window dilates and translates. As a result, the phase spectrum is absolute in the sense that it is always referred to the origin of the time axis, the fixed reference point. The real and imaginary spectrum can be localized independently with a resolution in time, corresponding to the basis function in question and the changes in the absolute phase of a constituent frequency can be followed along the time axis and useful information can be extracted. The phase correction of the wavelet transform in the form of S transform can provide significant improvement in the detection and localization of power quality disturbance transients [35].

As said by [35], the S Transform is an extension of the Wavelet Transform as show in Equation 2.13 with a Gaussian window with $\mu = 0$ and $\sigma^2 = 1$ as shown in Equation 2.14.

$$WT_x(t,\sigma) = \frac{1}{\sqrt{\sigma}} \int_{-\infty}^{\infty} x(\tau)\psi\Big(\frac{\tau - t}{\sigma}\Big) d\tau \tag{2.13}$$

$$\psi(t) = \frac{1}{\sqrt{2\pi}} e^{\frac{t^2}{2}} \tag{2.14}$$

Substituting Equation 2.14 in Equation 2.13, Equation 2.15 is obtained. the Gaussian window function directly affects the time frequency resolution of ST. Substituting the time shift factor shown in Equation 2.16 in Equation 2.15, the S Transform is obtained as shown in Equation 2.17.

$$WT_x(t,\sigma) = \frac{1}{\sqrt{\sigma}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} x(\tau) e^{-\frac{(\tau - t)^2}{2\sigma^2}} d\tau \tag{2.15}$$

$$\sigma = \frac{1}{|f|} \tag{2.16}$$

$$S(t,f) = \int_{-\infty}^{\infty} \frac{|f|}{\sqrt{2\pi}} x(\tau) e^{-\frac{f^2(\tau - t)^2}{2}} e^{-i2\pi f\tau} d\tau \tag{2.17}$$

In [36] the authors use a modified version of the S Transform using an improved window function of energy concentration in time-frequency distribution to optimize the shape of each window function. This method determines the parameters of Gaussian window to maximize the product of energy concentration in a time-frequency domain within a given time and frequency interval, so as to improve the energy concentration. The result shows that compared with the SST with Gaussian window, ST based on the optimally concentrated window proposed in this paper has better energy concentration in time-frequency distribution.

2.2.5 Hilbert Huang Transform

The HHT is derived from the principals of empirical mode decomposition EMD and the Hilbert Transform. When applying the HHT, first, the EMD will decompose the acquired signal into a collection of intrinsic mode functions IMF. The IMF is a kind of complete, adaptive and almost orthogonal representation for the analysed signal. Since the IMF is almost monocomponent, it can determine all the instantaneous frequencies from the nonlinear or non stationary signal. Second, the local energy of each instantaneous frequency can be derived through the Hilbert Transform. Hence, the result is an energy frequency time distribution of the signal. Since applying the process of HHT is not computational intensive, the HHT becomes a promising method to extract the properties of nonlinear and non stationary signal [37].

Empirical Mode Decomposition

The essence of the method is to identify the intrinsic oscillatory modes by their characteristic time scales in the data empirically, and then decompose the data accordingly [38]. An intrinsic mode function (IMF) is a function that satisfies two conditions as said in [38]:

1. in the whole data set, the number of extrema and the number of zero crossings musteither equal or differ at most by one.

2. At any point, the mean value of theenvelope defined by the local maxima and the envelope defined by the local minimais zero.

To obtain the EMD given the requisites of a IMF an algorithm must be followed. As said in [39], given a signal $x(t)$, the effective algorithm of EMD can be summarized as follows:

1. Identify all extrema of $x(t)$.

2. Interpolate between minima (resp. maxima), ending up with some ”envelope” $e_{min}(t)$(resp. $e_{max}(t)$.)

3. Compute the average $m(t) = \big(e_{min}(t) + e_{max}(t)\big)/2$.

4. Extract the detail $d(t) = x(t) - m(t)$.

5. Iterate on the residual $m(t)$.

In practice, the above procedure has to be refined by a sifting process which amounts to first iterating steps 1)$-$4) upon the detail signal , until this latter can be considered as zero-mean according to some stopping criterion. Once this is achieved, the detail is considered as the effective IMF, the corresponding residual is computed and step 5) applies [39].

Hilbert Transform

The HT is one of the integral transforms (like Laplace and Fourier); it is named after David Hilbert, who first introduced it to solve a special case of the integral equations in the area of mathematical physics. The HT of the function $x(t)$ is defined by an integral transform [40]. The Hilbert Transform is shown in Equation 2.18.

$$H[x(t)] = \tilde{x}(t) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau \tag{2.18}$$

In [41], the Hilbert-Huang Transform and the Multilayer Perceptron Neural Network model are implemented in order to detect and classify disturbances in power quality. Eight common types of disturbances were analyzed based on the parameters stated in the IEEE 1159 standard. By means of instantaneous frequencies and intrinsic mode functions of each disturbance, the neural network is trained for the classification of these disturbances. The implemented method reached a precision percentage of 94.6, demonstrating the versatility and great potential that this method provides when detecting disturbances in power quality.

2.3 Machine Learning Algorithms

2.3.1 Support Vector Machines

An SVM decision function is more precisely an optimal hyperplane that serves to separate classify observations belonging to one class from another based on patterns of information about those observations called features. That hyperplane can then be used to determine the most probable label for unseen data. The features used to infer the hyperplane are not typically raw data; rather, they are most often derivative data resulting from some kind of interpolation during the feature selection stage Features are further referenced by coordinates based on their relationships to each other and form the support vectors. As with other forms of machine learning, workin with SVM involves balancing two complementary aimsd(1) maximizing the percentage of correct labels assigned to new examples by the classifier (optimizing its accuracy) and (2) ensuring that the classifier is generalizable to new data (optimizing its reproducibility). While the former is bound by the informativeness of the features used (i.e., feature importance), the latter is bound by the number of unique examples used to train the model [42].

The authors in [43] aim at automatic classification of power quality events using Wavelet Packet Transform (WPT) and Support Vector Machines (SVM). The features of the disturbance signals are extracted using WPT and given to the SVM for effective classification. Recent literature dealing with power quality establishes that support vector machine methods generally outperform traditional statistical and neural methods in classification problems involving power disturbance signals. However, the two vital issues namely the determination of the most appropriate feature subset and the model selection, if suitably addressed, could pave way for further improvement of their performances in terms of classification accuracy and computation time. This paper addresses these issues through a classification system using two optimization techniques, the genetic algorithms and simulated annealing. This system detects the best discriminative features and estimates the best SVM kernel parameters in a fully automatic way. Effectiveness of the proposed detection method is shown in comparison with the conventional parameter optimization methods discussed in literature like grid search method, neural classifiers like Probabilistic Neural Network (PNN), fuzzy k-nearest neighbor classifier (FkNN) and hence proved that the proposed method is reliable as it produces consistently better results. Results show a 98.33% accuracy of the proposed method compared to 96.25% and 97.92% accuracy of other methods.

2.3.2 K-Nearest Neighbor

The k-nearest neighbor decision rule (k-NN) is a commonly used classification algorithm in statistical pattern recognition. Each class is given as a set of sample prototypes, a training set of pattern vectors from that class. When an unknown vector is to be classified, its k closest neighbors are found from among all the prototype vectors, and the class label is decided based on a majority rule. To avoid ties on class overlap regions, the value of k should be odd. This rule is simple and elegant, and yet the error rate is small in practice. In theory, it is known that the asymptotic error rate as the number of prototype samples gets very large is close to the optimal Bayes error rate and actually tends to it when k is increased. For this reason, the k-NN rule has become the standard comparison method against which any new classifiers, e.g. neural networks, are compared [44].

In [45],power quality abnormality present in power supply was detected and classified using S Transform and k-nearest neighbors Classifier (KNN). In this work results show a 96.35% overall accuracy.

2.3.3 Artificial Neural Networks

An artificial neural network (or simply neural network) consists of an input layer of neurons (or nodes, units), one or two (or even three) hidden layers of neurons, and a final layer of output neurons. Each connection is associated with a numeric number called weight. The output, hi, of neuron i in the hidden layer is as shown in Equation 2.19

$$h_i = \sigma\Big(\sum_{j=1}^{N} V_{ij}x_j + T_i^{hid}\Big) \tag{2.19}$$

where $\sigma()$ is called activation (or transfer) function, N the number of input neurons, V_{ij} the weights, x_j inputs to the input neurons, and T_i^{hid} the threshold terms of the hidden neurons. The purpose of the activation function is, besides introducing nonlinearity into the neural network, to bound the value of the neuron so that the neural network is not paralyzed by divergent neurons [46].

In [47], a new dual neural-network-based methodology to detect and classify single and combined PQ disturbances is proposed, consisting, on the one hand, of an adaptive linear network for harmonic and interharmonic estimation that allows computing the root-mean-square voltage and total harmonic distortion indices. With these indices, it is possible to detect and classify sags, swells, outages, and harmonics interharmonics. On the other hand, a feedforward neural network for pattern recognition using the horizontal and vertical histograms of a specific voltage waveform can classify spikes, notching, flicker, and oscillatory transients. The combination of the aforementioned neural networks allows the detection and classification of all the aforementioned disturbances even when they appear simultaneously. An experiment under real operating conditions is carried out in order to test the proposed methodology. Results show an 96.31% overall accuracy containing combined disturbances.

2.4 Deep Learning Algorithms

Over the years research has brought artificial neural networks ,presented in section 2.3.3, with feature extraction capabilities, some explained in Section 2.2.1 to 2.2.5. The idea of deep learning algorithms are to train the feature extraction capabilities and the classification in one single learnable algorithm. Having this in contex, many different types of algorithms have been developed for different purposes, for example, the reigning algorithm for image classification is the Convolutional Neural Network (CNN). The CNN is able to the parameters for filters in order to extract the most important parts of an image. On the other hand, the most widely use algorithm for time series data is the Long-Short Term Memory (LSTM) because of its ability to eliminate long term, irrelevant data. These algorithms will be explained in the following sections in more detail.

2.4.1 Long-Short Term Memory

The Long-Short Term Memory (LSTM) architecture is a deep neural network that uses a series of gates to extract characteristic part of time series data. The LSTM is a recurrent network architecture in conjunction with an appropriate gradient-based learning algorithm. LSTM is designed to overcome error backflow problems. It can learn to bridge time intervals in excess of 1000 steps even in case of noisy, incompressible input sequences, without loss of short-time lag capabilities. This is achieved by an efficient, gradient-based algorithm for an architecture enforcing constant (thus, neither exploding nor vanishing) error flow through internal states of special units (provided the gradient computation is truncated at certain architecture-specific points; this does not affect long-term error flow, though) [48].

In [49] and [50] uses the LSTM network, to detect and classifies the PQ events in one step. This technique extracts amplitude, disturbance duration and total harmonic distortion from the captured waveform, and the LSTM classify the PQ events. Many simple PQ events such as interruption, sag, flicker, swell and surge or complex PQ events such as sag plus harmonics and swell plus harmonics are generated using MATLAB programming environment to evaluate the performance of LSTM. Also, real-time measurements are collected from an industrial substation and are used to ensure the effectiveness of the proposed LSTM technique. A comparison with other techniques is conducted and the results verify the good performance of LSTM in classifying the PQ problems.

2.4.2 Convolutional Neural Network

CNNs are comprised of three types of layers. These are convolutional layers, pooling layers and fully-connected layers. When these layers are stacked, a CNNarchitecture has been formed.The basic functionality of the example CNN above can be broken down into four key areas as enumerated in [51]:

1. As found in other forms of ANN, the input layer will hold the pixel values of the image.

2. The convolutional layer will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume.The rectified linear unit(commonly shortened to ReLu) aims to apply an elementwise activation function such as sigmoid to the output of the activation produced by the previous layer.

3. The pooling layer will then simply perform downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation.

4. The fully-connected layers will then perform the same duties found in standard ANNs and attempt to produce class scores from the activations, to be used for classification. It is also suggested that ReLu may be used between these layers, as to improve performance.

In [52], the authors propose a novel full closed-loop approach to detect and classify power quality disturbances based on a deep convolutional neural network. Considering the characteristics of power quality disturbances problem, a unit construction which consists of 1-D convolutional, pooling, and batch-normalization layers is designed to capture multi-scale features and reduce overfitting. In the proposed deep convolutional neural network, multiple units are stacked to extract features from massive disturbance samples automatically. Comparisons with other state of-the-art deep neural networks and traditional methods proves that the proposed method can overcome defects of traditional signal process and artificial feature selection. Considering microgrid is an important development form of multi-energy system and an essential part of smart grid, a typical simulation system is constructed to analyze the causes of power quality problems in microgrid and the field data from a multi-microgrid system are used to further prove the validity of the proposed method.

2.4.3 Gated Recurrent Unit

Gated Recurrent Neural Networks (Gated RNNs) have shown success in several applications involving sequential or temporal data [53]. For example, they have been applied extensively in speech recognition, music synthesis, natural language processing, machine translation, etc. Long Short-Term Memory (LSTM) RNNs and the recently introduced Gated Recurrent Unit (GRU) RNNs have been successfully shown to perform well with long sequence applications. Gated RNNs success is primarily due to the gating network signaling that control how the present input and previous memory are used to update the current activation and produce the current state. These gates have their own sets of weights that are adaptively updated in the learning phase (i.e., the training and evaluation process). While these models empower successful learning in RNNs, they introduce an increase in parameterization through their gate networks. Consequently, there is an added computational expense vis-a-vis the simple RNN model. It is noted that the LSTM RNN employs 3 distinct gate networks while the GRU RNN reduces the gate networks to two [54].

In [55], The authors present a new concise deep learning based sequence model to detect the power quality disturbances (PQD), which only uses original signals and does not require preprocessing and complex artificial feature extraction process. A simple gated recurrent network (SGRN) with a new recurrent cell structure is developed, which consists of only two gates : forget gate and input gate, and two weight matrices. Compared with the standard Recurrent Neural Network (RNN) model, the training process of the proposed method is more stable and the prediction accuracy is higher. In addition, this special structure retains basic nonlinearity and long term memory, while enabling the simple gated recurrent network model to be superior to Long Short Term Memory (LSTM) Network and Gated Recurrent Unit (GRU) Network in terms of the number of parameters (i.e. memory cost) and detection speed. In the light of the experimental results, the simple gated recurrent network algorithm can achieve 99.07% detection accuracy, and contains only 18,959 parameters, which indicates that our proposed method is easier to deploy in resource constrained internet of things (IoT) micro-controllers.

2.4.4 Transformers

Most competitive neural sequence transduction models have an encoder-decoder structure. Here, the encoder maps an input sequence of symbol representations $(x1, ..., xn)$ to a sequence of continuous representations $z = (z1, ..., zn)$. Given z, the decoder then generates an output sequence $(y1, ..., ym)$ of symbols one element at a time. At each step the model is auto-regressive, consuming the previously generated symbols as additional input when generating the next. The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 1, respectively [56].

In [57], the authors proposed a novel method based on visual attention mechanism and feed-forward neural network to classify single and combined power quality disturbances caused by non-balanced, nonlinear loads and distributed generations in the power grid. In the first step of the proposed method, visual attention mechanism is utilized to extract the disturbance features of power quality disturbances, through performing disturbance region selection, multi-scale spatial rarity analysis, and disturbance feature fusion on the binary image converted from the original voltage signal successively. Then, four disturbance feature indexes are selected for the characterization of power quality disturbances. Finally, a classifier using feed-forward neural network is constructed to distinguish various single and combined power quality disturbances. The classification accuracy of the proposed method is compared with that of several existing methods for the classification of power quality disturbances from two types of datasources. The power quality disturbances from the simulation operating conditions include eight kinds of single and thirty-eight kinds of combined power quality disturbances. The power quality disturbances from the IEEE Work Group P1159.3 and P1159.2 Datasets include seven kinds of single and eleven kinds of combined power quality disturbances. Comparison results demonstrate that the proposed method can classify single and combined power quality disturbances more accurate than the compared classification methods, which verifies the effectiveness of the proposed method.

2.5 Object Detection Algorithms

Object detection algorithms are algorithms that are specifically design to detect and classify different objects or anomalies from input signals. Usually, object detection algorithms are used to detect and classify objects in an image [58] [59] [60]. The goal of object detection is to detect and classify objects leading to many specialized fields and applications such as face detection and face recognition [61]. Over the years, these types of algorithms have proven useful to efficiently detect and classify object in images. These algorithms have evolved and have included deep learning algorithms. The three most used deep learning algorithms are the Regional Convolutional Neural Network (R-CNN), the You Only Look Once (YOLO), and the Single Shot Detector (SSD).

2.5.1 R-CNN

Recently, advanced deep learning techniques, especially region-based CNN (R-CNN), have attained remarkable successes in a diversity of tasks in machine learning, statistics, and computer vision, for example, object detection, image categorization, image segmentation, etc. This is because CNN only identifies the objects class, not the location of object in an image. Especially when multiple objects are in the image then CNN cannot work well due to interference [62].The problem the R-CNN system tries to solve is to locate objects in an image (object detection). The R-CNN does this by proposing regions of interest (ROI) and classifying these regions using a CNN architecture.

In [63], the authors propoes a method, called Mask R-CNN, that extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing to estimate human poses in the same framework. the authors show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding-box object detection, and person keypoint detection. Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners.

2.5.2 YOLO

The YOLO algorithm separates components of object detection into a single neural network. This network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This system divides the input image into an $S \times S$ grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. Formally, the confidence is defined as $Pr(Object) \times IOU^{truth}_{pred}$. If no object exists in that cell, the confidence scores should be zero. Otherwise we want the confidence score to equal the intersection over union (IOU) between the predicted box and the ground truth. Each bounding box consists of 5 predictions: $x, y, w, h, and confidence.$ The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell. The width and height are predicted relative to the whole image. Finally the confidence prediction represents the IOU between the predicted box and any ground truth box. Each grid cell also predicts C conditional class probabilities, $Pr(Classi|Object)$. These probabilities are conditioned on the grid cell containing an object [64].

In [65], the authors propose an improved YOLO-V3 model for detecting apples during different growth stages in orchards with fluctuating illumination, complex backgrounds, overlapping apples, and branches and leaves. Images of young apples, expanding apples, and ripe apples are initially collected. These images are subsequently augmented using rotation transformation, colour balance transformation, brightness transformation, and blur processing. The augmented images are used to create training sets. The DenseNet method is used to process feature layers with low resolution in the YOLO-V3 network. This effectively enhances feature propagation, promotes feature reuse, and improves network performance. After training the model, the performance of the trained model is tested on a test dataset. The test results show that the proposed YOLOV3-dense model is superior to the original YOLO-V3 model and the Faster R-CNN with VGG16 net model, which is the state-of-art fruit detection model. The average detection time of the model is per frame at 3000 × 3000 resolution, which can provide real-time detection of apples in orchards. Moreover, the YOLOV3-dense model can effectively provide apple detection under overlapping apples and occlusion conditions, and can be applied in the actual environment of orchards.

2.5.3 Single Shot Detector

As described in [66], the SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections. The early network layers are based on a standard architecture used for high quality image classification (truncated before any classification layers), which it is called the base network. Auxiliary structure to the network are added to produce detections with the following key features:

Multi-scale feature maps for detection Convolutional feature layers are added to the end of the truncated base network. These layers decrease in size progressively and allow predictions of detections at multiple scales.

Convolutional predictors for detection Each added feature layer (or optionally an ex- isting feature layer from the base network) can produce a fixed set of detection predic- tions using a set of convolutional filters. For a feature layer of size $m \times n$ with p channels, the basic element for predicting parameters of a potential detection is a $3 \times 3 \times p$ small kernel that produces either a score for a category, or a shape offset relative to the default box coordinates. At each of the $m \times n$ locations where the kernel is applied, it produces an output value.

Default boxes and aspect ratios A set of default bounding boxes are associated with each feature map cell, for multiple feature maps at the top of the network. The default boxes tile the feature map in a convolutional manner, so that the position of each box relative to its corresponding cell is fixed. At each feature map cell, the offsets relative to the default box shapes in the cell is predicted, as well as the per-class scores that indicate the presence of a class instance in each of those boxes. Specifically, for each box out of k at a given location, c class scores and the 4 offsets relative to the original default box shape are computed. This results in a total of $(c + 4)k$ filters that are applied around each location in the feature map, yielding $(c + 4)kmn$ outputs for a $m \times n$ feature map.

In [67], a set of benchmarks for object tracking with motion parameters (OTMP) was designed. The sample images were matched with the spatial depth of the camera, the pose of the camera, and other spatial parameters for the training of the detection model. Then, a Fast Depth-Assisted Single-

Shot MultiBox Detector (FDA-SSD) algorithm suitable for 3D target tracking was proposed by combining the depth information of the sample into the original Single-Shot MultiBox Detector (SSD). Finally, an FDA-SSD-based monocular motion platform target detection and tracking algorithm framework were established. Specifically, the spatial geometric constraints of the target were adapted to solve the target depth information, which was fed back to the detection model. Then, the normalized depth information of the target was employed to select the feature window of the convolutional layer for the detector at a specific scale. This significantly reduces the computational power for simultaneously calculating detectors of all scales. This framework effectively combines the two-dimensional detection model and the three-dimensional positioning algorithm. Compared with the original SSD method, the network model designed in this study has fewer actual operating parameters; the measured detection operation speed was increased by about 18.1% on average; the recognition rate was maintained at a high level consistent with that of the original SSD. Furthermore, several groups of experiments were conducted on target detection and target space tracking based on monocular motion platforms indoors. The root mean square error (RMSE) of the spatial tracking trajectory was less than 4.72 cm. The experimental results verified that the algorithm framework in this study can effectively realize tasks such as visual detection, classification, and spatial tracking based on a monocular motion platform.

Chapter 3

Dataset Generation

In this chapter the dataset generation is described. The first section is a brief introduction that explains the disturbances that are to be generated. In the second section a Matlab/Simulink model is presented which simulates the generation, transmission, distribution, and the load. The Matlab/Simulink model in this chapter contains a disturbance block which simulates a Sag, Swell, Harmonics, Transient, Notch and Interruption disturbances. Section 3 presents a data augmentation procedure that helps generalized the dataset acquired by the Matlab/Simulink model. Section 4, using the generated dataset, the STFT is implemented in order to study and obtain the disturbances in the time-frequency domain. Finally, a conclusion in section 5 is presented. [1]

3.1 Introduction

As a preliminary stage of this work, the voltage and current time signals are simulated using MATLAB Simulink. By doing this, it is possible to recreate the disturbances on the line and see how they interact with the appliances connected to the grid. Thanks to the simulation results, it is possible to acquire a current and voltage dataset with which the identification algorithms are trained, validated and tested. This dataset includes simulations

[1] This chapter has been published in part as “A Comparison of Power Quality Disturbance Detection and Classification Methods Using CNN, LSTM and CNN-LSTM” in *MDPI Applied Sciences, 2020* [15].

of several disturbances such as Sag, Swell, Harmonics, Transient, Notch and Interruption. After that, the networks are fed with an experimental dataset of voltage and current field measurements containing the disturbances mentioned above.

For the generation of the PQD dataset a MATLAB / Simulink model of a micro grid has been implemented. The model is shown in Figure 3.1 and it includes several different industrial loads. It is possible to identify a three-phase dynamic load which could be associated to an electrical motor with variable load, a linear load and a nonlinear load which injects disturbances on the net [68]. These disturbances include:

- Sag
- Swell
- Harmonics
- Transient
- Notch
- Interruption

3.2 Simulink Models

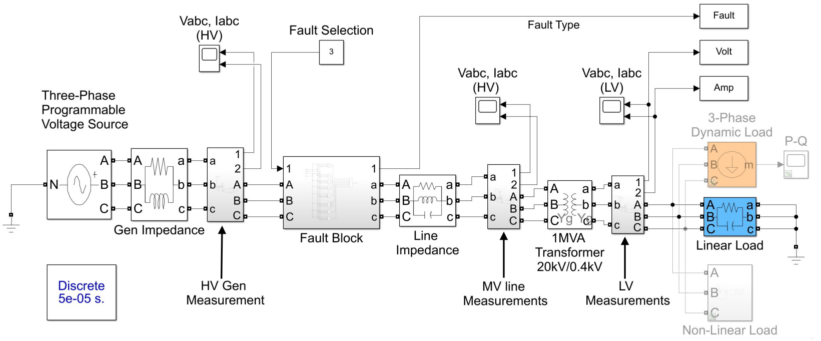
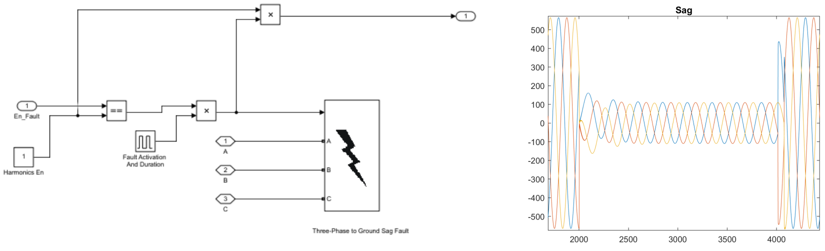


Figure 3.1: Simulink Model that generates voltage disturbance in a distribution network.

The simulink model simulates a power grid beginning from the generation of power all the way to the load. The simulink models the generation of power using the three-phase programmable voltage source. The transmission line is simulated using the General Impedance block. The disturbances are then generated using the fault block. The fault block is a programmable block in which the fault is chosen using the Fault Selection block. The Fault Selection block spans through 1-6 in which 1 is for sag, 2 is for swell, 3 is for harmonics, 4 is for transient, 5 is for notch and 6 for interruption. The distribution line impedance are simulated using the line impedance block which is connected to a 1kVA transformer which ends in the simulated loads.

The Fault block for each disturbance is shown in figures 3.2 to 3.7. In Figure 3.2-a, the functioning of a sag disturbance is show along with the three phase voltage signal simulation shown in Figure 3.2-b. The sag simulation is a block that contains a duty cycle block that activates the block. The three phase voltage signal is then reduced using the three phase to ground sag fault block.

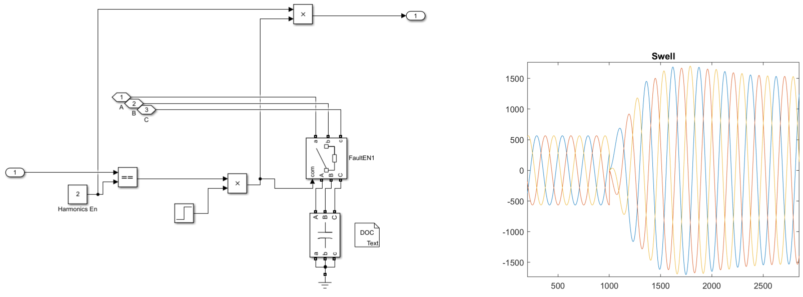


(a)

(b)

Figure 3.2: (a) Precision comparison of deep learning architectures (b) Recall comparison of deep learning architectures

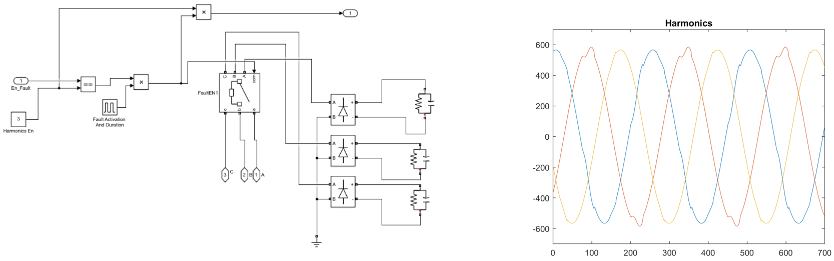
In Figure 3.3-a, the functioning of a swell disturbance is show along with the three phase voltage signal simulation shown in Figure 3.3-b. The swell disturbance in a voltage line is activated using the disturbance control block and with a step function block. When the swell block is activated, the fault enable block is then turned on which it is connected to a capacitor with initial conditions. The effects show a voltage rise measured in the simulated distribution line.



(a) (b)

Figure 3.3: (a) Precision comparison of deep learning architectures (b) Recall comparison of deep learning architectures

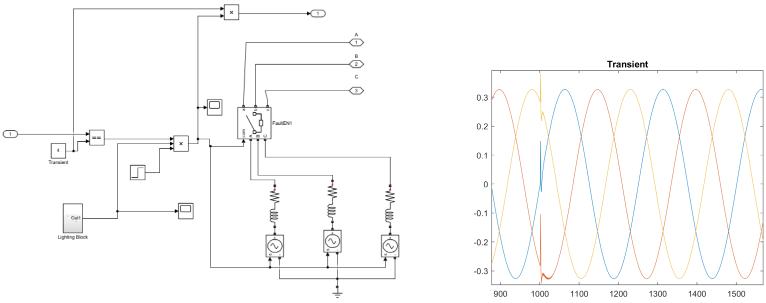
In Figure 3.4-a, the functioning of a harmonic disturbance is show along with the three phase voltage signal simulation shown in Figure 3.4-b. Similar to the other disturbances, the harmonics disturbance is generated by choosing the 3 in the disturbance control block. A duty cycle then activates the Fault enable block. The fault enable block is connected to three diodes connected in parallel with capacitors and resistances. The diodes, which are non-linear components, inject harmonic components with the help of the duty cycle block and the capacitors with initial conditions.



(a) (b)

Figure 3.4: (a) Precision comparison of deep learning architectures (b) Recall comparison of deep learning architectures

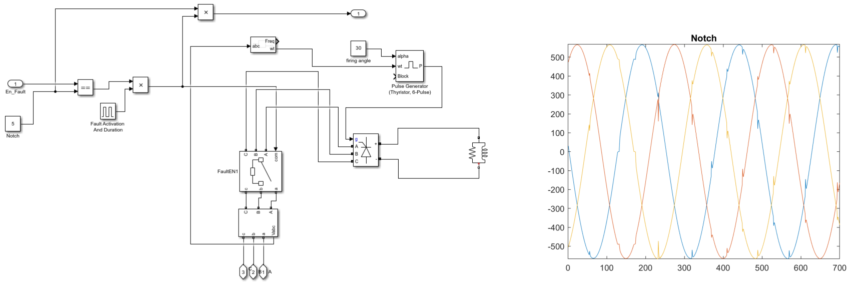
In Figure 3.5-a, the functioning of a Transient disturbance is show along with the three phase voltage signal simulation shown in Figure 3.5-b. The transient disturbance is also controlled by the disturbance control block which is connected to a step function block. The difference with this disturbance is that it also has a lighting block. The lighting block injects a short time duration with a linear increment with a high slope. These blocks activate another generator connected to ac generators in series with a resistance and an inductive element with initial conditions. This simulink configuration generates a spike seen clearly in the current signal. This spike is shown in Figure 3.5.



(a) (b)

Figure 3.5: (a) Precision comparison of deep learning architectures (b) Recall comparison of deep learning architectures

In Figure 3.6-a, the functioning of a Notch disturbance is show along with the three phase voltage signal simulation shown in Figure 3.6-b. Works by connecting the disturbance control block to with the duty cycle block to a fault enable block or switch. The fault enable block is connected to a silicon controlled rectifier in parallel with a resistance and an inductor. The silicon controlled rectifier is activated using a pulse generator block or thyristor. These simulink model generate the notching effect shown in Figure 3.6-b.

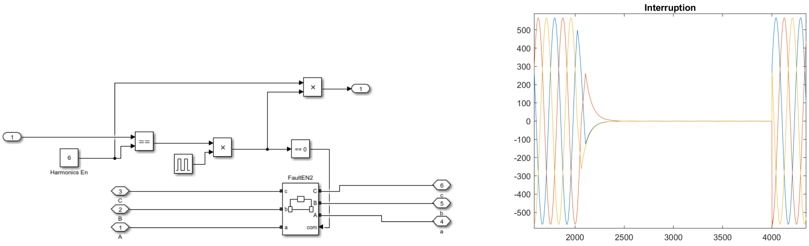


(a)

(b)

Figure 3.6: (a) Precision comparison of deep learning architectures (b) Recall comparison of deep learning architectures

In Figure 3.7-a, the functioning of a interruption disturbance is show along with the three phase voltage signal simulation shown in Figure 3.7-b. The interruption is preformed by adding a fault enable block or switch to the duty cycle block and to the disturbance control block. This combination enables the block to cut the voltage supply as shown in Figure 3.7-b.



(a)

(b)

Figure 3.7: (a) Precision comparison of deep learning architectures (b) Recall comparison of deep learning architectures

3.3 Data Augmentation

In order to obtain a more generalized result for training and validation data augmentation is necessary. Data augmentation is the art of manipulating the training and validation set to obtain more data with small variations. In our problem, the network is trained with all three phases, the negative is calculated from all phases and oversampled by 2 and 4. That is, from 1 simulation result, 12 new samples are obtained.

$$\mathbf{Voltage_{3\phi}^{T}} = \begin{bmatrix} \mathbf{V_{\phi_1}^{T}} \\ \mathbf{V_{\phi_2}^{T}} \\ \mathbf{V_{\phi_3}^{T}} \end{bmatrix} \qquad \mathbf{Voltage_{3\phi}^{\frac{T}{2}}} = \begin{bmatrix} \mathbf{V_{\phi_1}^{\frac{T}{2}}} \\ \mathbf{V_{\phi_2}^{\frac{T}{2}}} \\ \mathbf{V_{\phi_3}^{\frac{T}{2}}} \end{bmatrix} \qquad \mathbf{Voltage_{3\phi}^{\frac{T}{4}}} = \begin{bmatrix} \mathbf{V_{\phi_1}^{\frac{T}{4}}} \\ \mathbf{V_{\phi_2}^{\frac{T}{4}}} \\ \mathbf{V_{\phi_3}^{\frac{T}{4}}} \end{bmatrix} \tag{3.1}$$

$$\mathbf{Class}_d = \begin{bmatrix} \mathbf{Voltage_{3\phi}^{T}} \\ \mathbf{Voltage_{3\phi}^{\frac{T}{2}}} \\ \mathbf{Voltage_{3\phi}^{\frac{T}{4}}} \\ -\mathbf{Voltage_{3\phi}^{T}} \\ -\mathbf{Voltage_{3\phi}^{\frac{T}{2}}} \\ -\mathbf{Voltage_{3\phi}^{\frac{T}{4}}} \end{bmatrix} \tag{3.2}$$

For this work the steps to do are as follows. First, finish the simulink model and adjust it to gather the fault simulation for every disturbance and organize it in a cell array. The disturbances that are implemented in the simulink model are the sag, voltage rise, interruption and no fault. The harmonic distortion, the transient and the notch were added using the examples in [69]. After the simulation is completed in simulink, a script was made that gathers the simulink disturbance data of each fault into a structured array. After that, each fault was labeled with a number as seen in the following:

- 0 for no fault
- 1 for sag
- 2 for voltage rise
- 3 for harmonics
- 4 for transient
- 5 for notch
- 6 for interruption

The data for training and the targets should look like in Equation 3.3. To train the neural network, the data for the **Class** and **Target** are shuffled together in order to obtain a generalized solution for the network.

$$Class = \begin{bmatrix} \textbf{Normal} \\ \textbf{Sag} \\ \textbf{Swell} \\ \textbf{Harmonics} \\ \textbf{Transient} \\ \textbf{Notch} \\ \textbf{Interruption} \end{bmatrix} \qquad Target = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \\ \mathbf{6} \end{bmatrix} \tag{3.3}$$

After the time series signals are stored in a structured array and shuffled, the next step is to train the network, validate and test it. For this, the dataset containing the time series signals are separated into 75% of the signals for training and the other 25% for validation. This network trains with a batch size of 20 data points with a maximum of 30 epoch making a total of 15 iterations per epoch. After the training is completed, the performance of the network is evaluated using a confusion matrix.

3.4 Short Time Fourier Transform (STFT)

The Short Time Fourier Transform is characterized by a fourier transform executed in a fixed windowed interval. The window function $g(n)$ is called a blackman's window function that is used to multiply a short segment of the signal with the window function. This avoids sharp sections and redundant information. A fourier transform of this small windowed section $X_n(j\omega)$ is calculated and stacked up to form a matrix. The STFT equation is shown in Equation 3.4. An illustration of the Blackman's window and the algorithm can be seen in Figure 3.8 and 3.9 respectively. This matrix can now be treated as an image to train a Convolutional Neural Network which is capable of feature extraction of the frequency component of the signal.

$$X(j\omega) = \sum x(n)g(n - mR)e^{-2j\pi fn} \tag{3.4}$$

window function blackmann

$$w[n] = a_0 + a_1 cos\Big(\frac{2\pi n}{N}\Big) + a_2 cos\Big(\frac{4\pi n}{N}\Big) \tag{3.5}$$

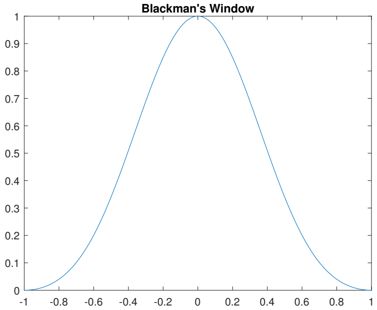


Figure 3.8: Blackman's Window

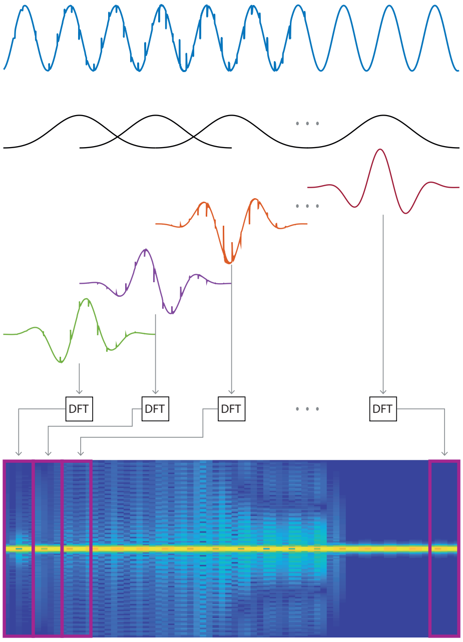
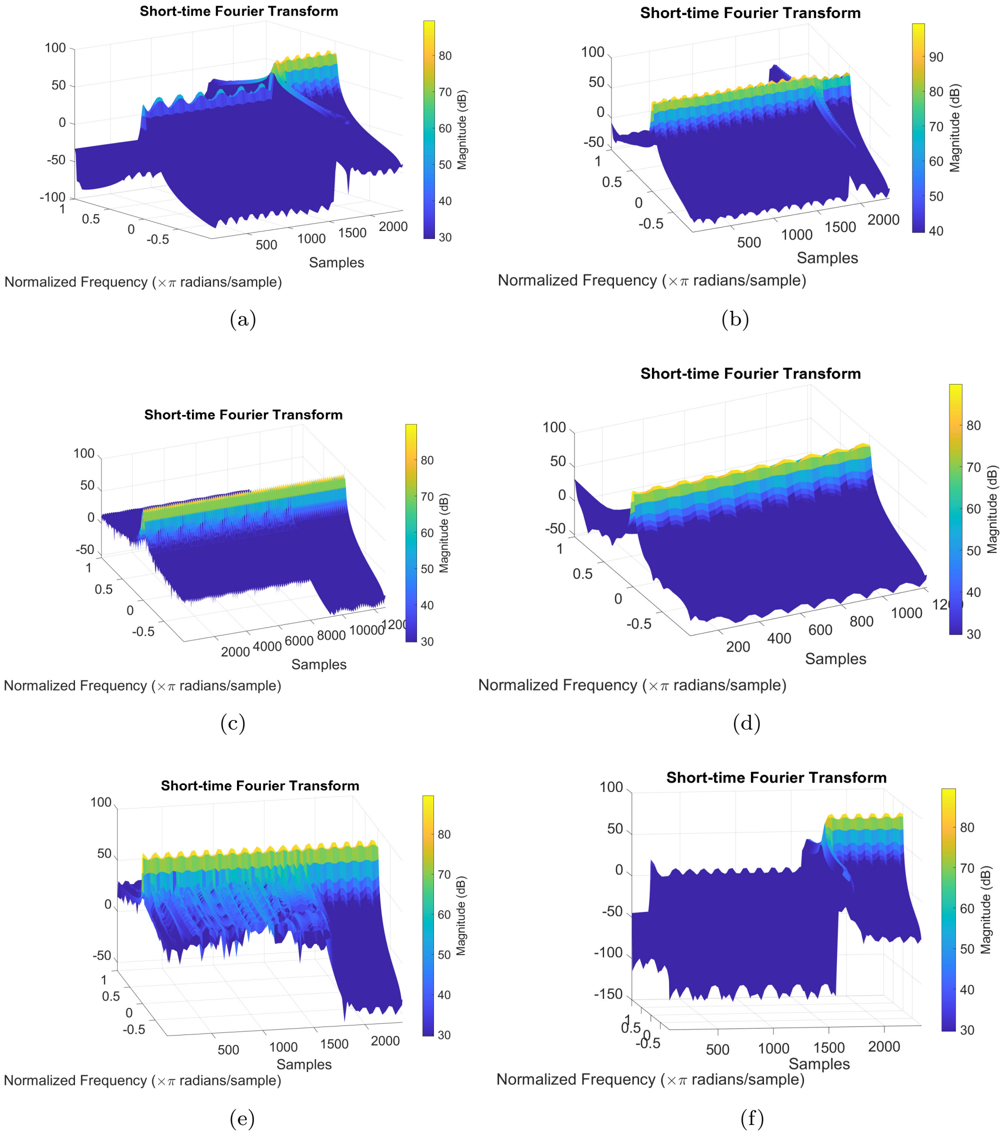


Figure 3.9: Short Time Fourier Transform algorithm illustration

Figure 3.10 shows the STFT of different PQDs where the x,y and z dimensions represents the samples, frequency and magnitude respectively. Figure 3.10 (a) shows the STFT of a Sag where it clearly shows a decrease in magnitude. On the ohter hands, Figure 3.10 (b) shows the STFT of a swell where it shows an increase in magnitude. Figure 3.10 (c) shows the STFT of a harmonic distortion where and increase in magnitude in frequencies higher than the fundamental are shown. Figure 3.10 (d) shows a constant value because there in no disturbance that causes irregularities in magitude or frequency. In Figure 3.10 (e) also shows an increase in magnitude in frequencies higher than the nominal 50Hz due to the injection of high frequency components in the notch PQD. Finally, the interruption that drops the magnitude to zero as shown in Figure 3.10 (f).



(a)

(b)

(c)

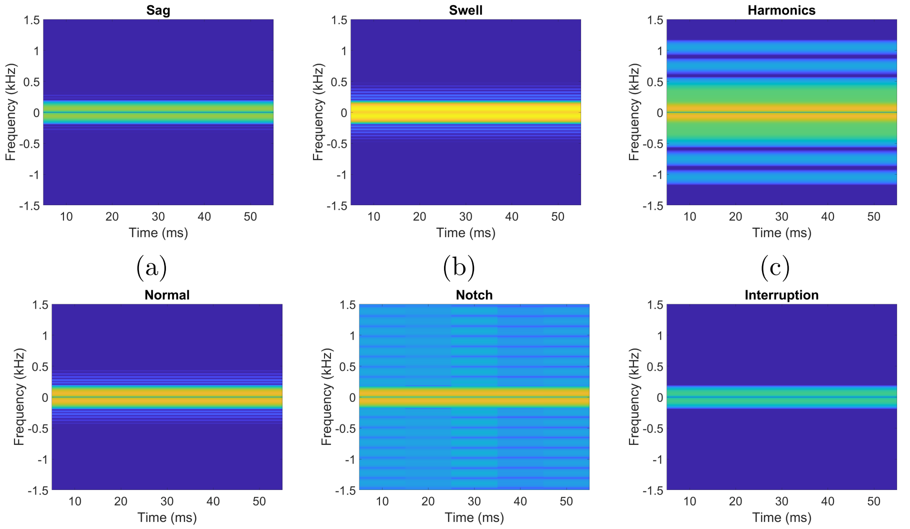
(d)

(e)

(f)

Figure 3.10: (a)STFT Sag (b) STFT Swell (c) STFT Harmonics (d) STFT Transient (e) STFT Notch (f) STFT Interruption

Figure 3.11 shows the STFT as a 2D image for classification using the Convolutional Neural Network explained in Chapter 5. Figure 3.11 (a) shows the sag where it clearly shows a reduction in magnitude. 3.11 (b) shows an increment in magnitude which corresponds to a swell disturbance. 3.11 (c) represents the harmonic distortion which clearly shows the injection of high frequency components. The Normal STFT is shown in Figure 3.11 (d). The notch is shown in Figure 3.11 (e) which also injects high frequency components to the voltage signal. Finally, 3.11 (f) shows the interruption which the magnitude is almost zero.



(a) (b) (c)

(d) (e) (f)

Figure 3.11: (a) STFT Sag (b) STFT Swell (c) STFT Harmonics (d) STFT Normal (e) STFT Notch (f) STFT Interruption

3.5 Conclusion

This chapter describes the dataset generation used for training machine learning and deep learning architectures in this thesis. The disturbances obtained for training machine learning algorithms include the: Sag, Swell, Harmonics, Transient, Notch, and Interruption. This disturbances were generated using a Matlab/Simulink model shown in Section 3.2. The dataset generated by the Simulink models were augmented in order to obtain a more generalized dataset so the dataset contains the most cases possible as shown in Section 3.3.

This chapter also presents a frequency analysis of different possible disturbance in a low voltage distribution lines. The frequency analysis used in this chapter is the Short-Time Fourier Transform (STFT). The STFT allows the extraction of the frequency components along its time instances. To do this, a Blackmann window was used and the discrete fourier transform (DFT) was then calculated to the window sections. This is shown in Section 3.4.

Chapter 4

Comparison of Different Deep Learning Architectures for Detection and Classification of Power Quality Disturbances

In this chapter a brief introduction on the machine learning algorithms commonly used for PQD classifications. Section 2, the Long-Short Term Memory (LSTM) is explained in detail including the feature extraction cells, the architecture and finally the classification with results. In Section 3, the Convolutional Neural Network (CNN) is presented. The feature extraction layer is explained in detail followed by the architecture used and finally the classification with results. Section 4 the CNN-LSTM is presented along with its architecture and the results with the hyperparameter tuning. Section 5 summarizes the comparison of the results obtained by the LSTM, CNN, CNN-LSTM and the CNN-LSTM with hyperparameter tuning. The comparison is compared using the Precision and Recall. The architectures were also compared with experimental measurements. Section 6 finalizes with a conclusion. [1]

[1] The results of this work were published in "A Comparison of Power Quality Disturbance Detection and Classification Methods Using CNN, LSTM and CNN-LSTM"in *MDPI Applied Sciences, 2020* [15].

4.1 Introduction

The most convenient way to classify PQD using classification algorithms is in the time domain. It is the most convenient way because no preprocessing is required and thus less computational effort and more classification speed. In theory, the signal is obtain by a measuring device and directly fed to a classification algorithm in order to detect and classify PQD. In this chapter the use of different deep learning algorithms is evaluated in order to detect and classify PQD. No preprocessing is done in order to evaluate effectiveness of the algorithms in the time domain. Several approaches have been explored in literature using different algorithms. In [70], a Kalman Filter is used in an UPQC to extract the state components of the distorted supply voltage and load current. The algorithm can classify PQD internally enabling the conditioning of the PQ signals for power factor correction. The technique seems to work well with the detection of sag, swell and harmonic distortion, however it shows a certain lag between the disturbance starting condition and the detection [71]; furthermore, the algorithm is usually applied to a restricted number of disturbances. On the other hand, the wavelet transform is used as a tool for analyzing PQD as shown in [72]. The tool is very useful for the extraction of the signals features for learning algorithms like the SVM as shown in [73], [74]. However, it does not perform disturbances detection by itself. The SVM showed interesting performances for the detection of a wide range of PQ disturbances and it is often used as a benchmark to assess the performances of other algorithms. The main disadvantage of the PQD detection techniques mentioned above is that, once the voltage and current waveforms are acquired, a preprocessing of the signal must be performed before feeding it to the algorithm. This usually consists of a signal features extraction. Deep learning algorithms solves this problem by implicitly applying a feature extraction for the classification of the signal. In other words, these algorithms could be fed with raw data and still make accurate classifications. This can help to speed up the identification and classification process especially in real time applications. For the training and the validation of deep learning algorithms, it is necessary an extensive dataset in order to avoid overfitting and obtain generalization. Unfortunately, it is not easy to obtain such datasets with experimental data. One reason is that performing on-field data sampling through measurement campaigns is time consuming, many of these disturbances indeed are sporadic, and it is not always possible to record an event with a desired amplitude and duration.

For that reason, simulated voltage disturbances are used in order to create the dataset for training and validation as shown in Chapter 3. For further generalizing the dataset, data augmentation is used, since it has proven to be efficient in improving accuracy by reducing overfitting [75], [76]. This chapter explores different deep learning architectures which are trained and validated using simulated data and tested using experimental data. Once the simulated data is generated, it is then augmented, in order to obtain a generalized result and overcome any sampling discrepancy and phase difference between simulated data and measured data. The signals are pre-processed in order to compare the accuracy of each architecture in their proven classification tasks. With respect to other works in which the training, validation and testing steps are performed using purely simulated data or purely experimental data [77] [78] [79], in the present work the training and validation steps are performed with simulated datasets, while the testing one is performed with experimental datasets that were acquired on the field.

4.2 Long-Short Term Memory

The first deep learning algorithm tested is the Long-Short Term Memory (LSTM). The LSTM is a recurrent neural network that was specifically designed for time series data. A recurrent neural network (RNN) is a neural network that simulates a discrete-time dynamical system that has an input x_t, an output y_t and a hidden state h_t as defined in [80]. A set back of RNNs is that they suffer from vanishing or exploding gradient. Truncating the gradient where this does not harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete-time steps by enforcing constant error flow through constant error carousels within special units [48].

The LSTM has 3 states that help the network to reduce the long term dependency of the data. These states are called the Forget State, Input State and the Output State. The Forget State eliminates redundant or useless data. The Input State process the new data and finally the Output State process the input data with the cell state. This will be explained further on.

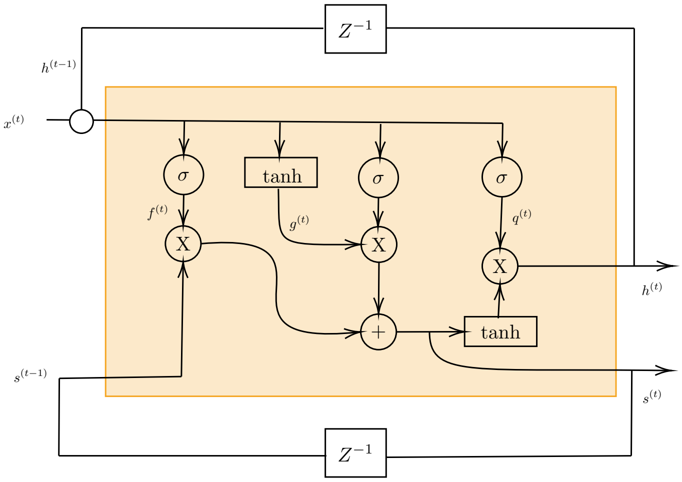


Figure 4.1: Block diagram of one cell of a Long Short Term Memory architecture.

4.2.1 Feature Extraction Cells

Forget State

The forget state controls the state parameter $s^{(t)}$ via a sigmoid function σ. This state controls what the cell should remember through time and what to forget. The equation of the forget state is seen in Equation 4.1. Where $f^{(t)}$ is the forget vector, x^t and $h^{(t-1)}$ are the input and previous output respectively. The input and the previous output are multiplied by trained weights U and W with bias b. This result is then truncated between 0 and 1 via a sigmoid function. Basically the idea is to have an input vector added with the previous output vector passed through a neural network which outputs the values to keep with a 1 and the values to forget with a 0.

$$f_i^{(t)} = \sigma\Big(b_i^f + \sum_j U_{i,j}^f x_j^t + \sum_j W_{i,j}^f h_j^{(t-1)}\Big) \tag{4.1}$$

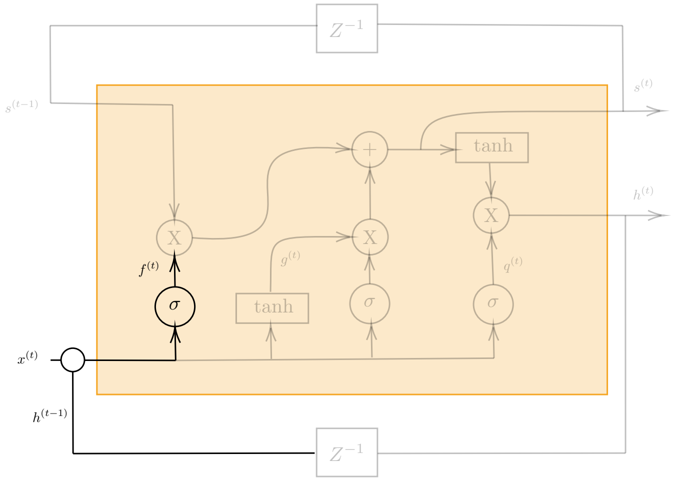


Figure 4.2: Block diagram of one cell of a Long Short Term Memory architecture.

Input State

The new state of the cell is defined in the input state where the previous state is multiplied by the forget state dropping off irrelevant information. This can be seen in $f_i^{(t)} s_i^{(t-1)}$ of Equation 4.2. Now the relevant information gets updated in $g_i^{(t)} \sigma\Big(b_i + \sum_j U_{i,j} x_j^t + \sum_j W_{i,j} h_j^{(t-1)}\Big)$ which is the multiplication of the input and previous output result from the neural network times $g^{(t)}$ which are the candidates for the next time step of the cell state. The equation that generates the vector that contains the candidates for the next cell state is shown in Equation 4.3.

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\Big(b_i + \sum_j U_{i,j} x_j^t + \sum_j W_{i,j} h_j^{(t-1)}\Big) \tag{4.2}$$

$$g_i^{(t)} = \tanh\Big(b_i^g + \sum_j U_{i,j}^g x_j^t + \sum_j W_{i,j}^g h_j^{(t-1)}\Big) \tag{4.3}$$

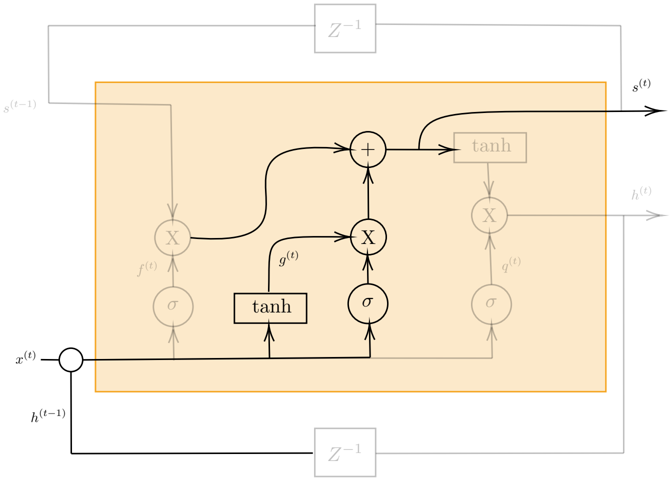


Figure 4.3: Block diagram of one cell of a Long Short Term Memory architecture.

Output State

The output state decides what should the output of the cell and the new cell state should be. The output of the cell is seen in Equation 4.4 where the cell state goes through a hyperbolic tangent and is then multiplied by the output of another hidden layer seen in Equation 4.5.

$$h_i^t = \tanh\left(s_i^{(t)}\right) q_i^{(t)} \tag{4.4}$$

$$q_i^{(t)} = \sigma\Big(b_i^o + \sum_j U_{i,j}^o x_j^t + \sum_j W_{i,j}^o h_j^{(t-1)}\Big) \tag{4.5}$$

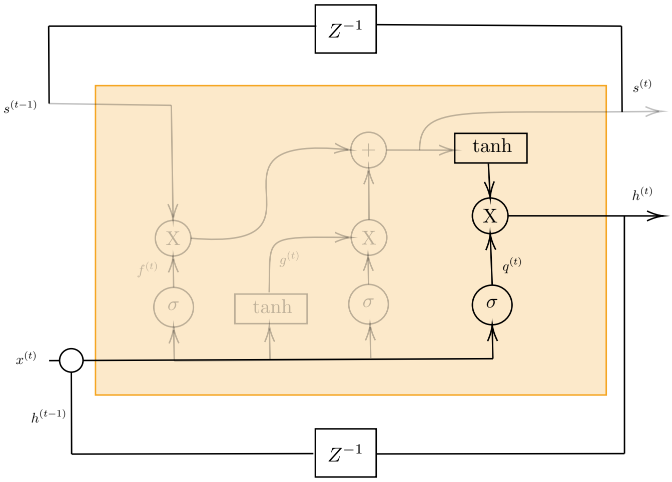


Figure 4.4: Block diagram of one cell of a Long Short Term Memory architecture.

4.2.2 Architecture

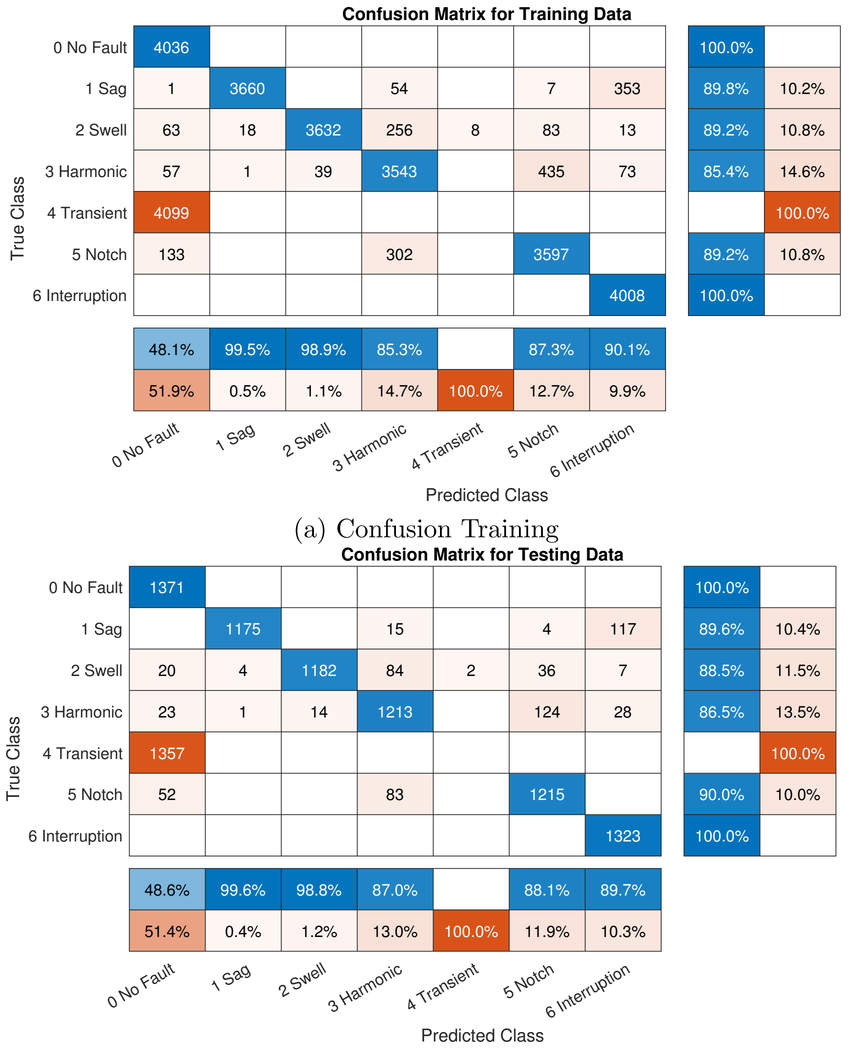
The LSTM architecture is fairly simple and short. The input size corresponds to the current datapoint and the 9 previously measured datapoints. That is, $\mathbf{X} = (x[n], x[n-1], x[n-2], \cdots, x[n-9])$. The LSTM is then composed of 100 cells, or hidden layers and then a fully connected layer with 6 outputs as mentioned earlier. Table 4.1 summarizes the LSTM architecture.

Table 4.1 LSTM architecture

Description	**Output Shape**
Input	[1 10 1]
LSTM	[1 100 1]
Fully Connected Layer	[1 1 6]

4.2.3 Classification

The LSTM training gave an accuracy of 79.14% where most of the problems were found in the Transient disturbance as shown in the precision and recall plots. The architecture was not able to detect the transient disturbance either in the training or the validation signals, that is, the LSTM classified the Transient signal as a No Fault in 100% of the cases. Concerning the other classes it resulted in 10.2%to 14.6% misclassification. Results are shown in Figure 4.5.



(a) Confusion Training

(b)Confusion Validation

Figure 4.5: LSTM Confusion Matrix for Training (a) and Validation (b)

4.3 Convolutional Neural Network

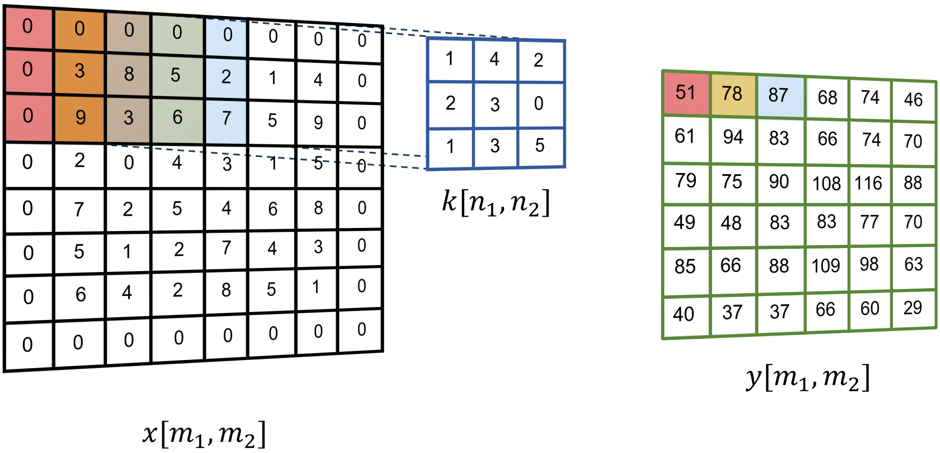
Convolutional neural networks or CNN, are a particular type of neural network for data processing that has a grid-like topology. Convolutional networks proved to be successful in several practical applications. They essentially consist of neural networks that use convolution in place of general matrix multiplication in at least one of their layers [81]. The convolutional layer is accompanied by a pooling layer which is a type of under sampling that helps with processing speed. Since the signals of interest are 1-D signals and the CNN processes a 2-D signal, a pre-processing of each signals is necessary. Hence, the Short Time Fourier Transform is performed on each signal before feeding it to the CNN; by doing this, an image containing the spectral components and amplitude of the signal of interest is generated.

4.3.1 Feature Extraction Layers

Convolutional Layer

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivariant representations. Moreover, convolution provides a means for working with inputs of variable size [81]. The convolution layer of a convolutional neural network operates by applying a convolution to each dataset. Since the hallmark of the CNN is image classification, the 2 dimensional version of the discrete convolution is used. To serve as a reminder, the 2 dimensional discrete convolution operation is shown in Equation 4.6.

$$y[n_1, n_2] = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} x[m_1, m_2]k[n_1 - m_1, n_2 - m_2] \tag{4.6}$$



0	0	0	0	0	0	0	0
0	3	8	5	2	1	4	0
0	9	3	6	7	5	9	0
0	2	0	4	3	1	5	0
0	7	2	5	4	6	8	0
0	5	1	2	7	4	3	0
0	6	4	2	8	5	1	0
0	0	0	0	0	0	0	0

$x[m_1, m_2]$

1	4	2
2	3	0
1	3	5

$k[n_1, n_2]$

51	78	87	68	74	46
61	94	83	66	74	70
79	75	90	108	116	88
49	48	83	83	77	70
85	66	88	109	98	63
40	37	37	66	60	29

$y[m_1, m_2]$

Figure 4.6: 2D Convolution

The convolutional layer can also work with 1D siganls. The convolutional layer in Figure 4.7 and Figure 4.8 shows how the 1 dimensional convolutional layer works. In Figure 4.7, The input (black outline) multiplies each element by each of the kernel elements (blue outline). The result is then added to obtain the result of the first element. This is done moving the kernel 1 step to the right until it reaches the last element of the input signal. In order to preserve the dimension of the output signal with respect to the input signal, zero padding is done. The results of these multiplications and additions are then concatenated to create the output of the layer.

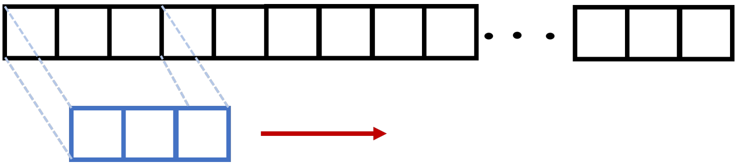


Figure 4.7: 1D Convolution

In order to reduce the dimension of the output of the convolution, 1 dimensional convolution is done without zero padding. The result is the convolved input with the kernel which also reduces the dimensionality and thus reduces the training time and computational effort. An example of the process is shown in Figure 4.8 where the input (Black Outline) is convolved with the kernel (Blue Outline). The first element of the output is the element-wise multiplication of the kernel with the first 3 elements of the input. The kernel then shifts one space and multiplies with the second, third and fourth element of the input. This is done until the kernel multiplies the last element of the input signal. The result of the element wise multiplications are then added and concatenated.

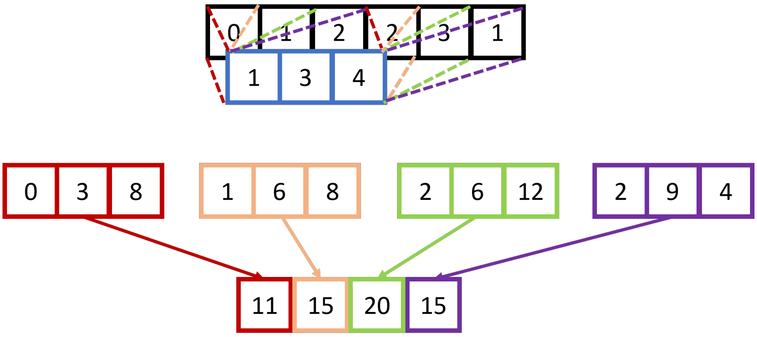


Figure 4.8: 1D Convolution

MaxPooling Layer

A pooling function replaces the output of the layer with a summary statistic of the previous layer outputs [81]. The most popular pooling functions include the max of a rectangular neighborhood, the average, the L2 norm, or a weighted average based on the distance from the central datum. This layer in the architecture speeds up the training and classification since it undersamples the dataset and helps the network to obtain a more generalized result. An illustration of the pooling function is shown in Figure 4.9. The matematical representation is shown in Equation 4.7 through Equation 4.10. Figure 4.7 represents the pooling function where X_n is the vector containing the pooled data of the dataset.

$$\mathcal{X}_n = \{x_j, ..., x_N\} \tag{4.7}$$

For each set of $\mathcal{X}_n$ the pooling function is performed. The summary statistic of the group of data represented by $\hat{x}$ is shown in Equation 4.8.

$$\begin{aligned}\hat{x}_1 &= f(x_1, x_2, x_3) = f(\mathcal{X}_1)\\ \hat{x}_2 &= f(x_2, x_3, x_4) = f(\mathcal{X}_2)\\ \hat{x}_3 &= f(x_3, x_4, x_5) = f(\mathcal{X}_3)\\ \hat{x}_4 &= f(x_4, x_5, x_6) = f(\mathcal{X}_4)\\ \hat{x}_5 &= f(x_5, x_6, x_7) = f(\mathcal{X}_5)\end{aligned} \tag{4.8}$$

Formaly, Equation 4.8 can be represented as shown in and Equation 4.9. Where, in this example, the pooling number is 3, the values obtained from the pooling are $n, n+1, n+2$. The pooling layer can contain either the maximum of the set, the average, the L2 norm or the weighted average of the pool. The most common to use is the maxpooling function. In this work the max pooling is used as shown in Equation 4.10.

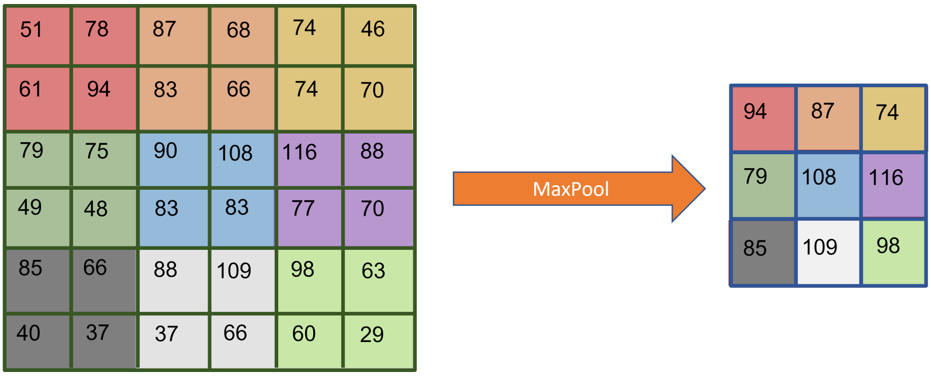
$$\hat{x}_n = f(x_n, x_{n+1}, x_{n+2}) = f(\mathcal{X}_n) \tag{4.9}$$

Figure 4.9 shows a representation of the maxpooling function in 2 dimensions. The representation basically divides the 6 × 6 matrix in a 3 × 3 cells. Each cell contains 4 elements in a 2 × 2 matrix. The Maxpooling layer then takes the largest value of these 4 elements and places it in the corresponding cell. For example, if in the first cell the largest element is 94, the the maxpooling layer places the value in the $(1, 1)$ position in the output matrix. This operation reduces the training time and computational effort.

$$f(\mathcal{X}_n) = argmax(\mathcal{X}_n) \tag{4.10}$$

4.3.2 Architecture

Table 4.2 shows the architecture used for this experiment. Using a sampling frequency of 8kHz, the signal takes as input 160 samples which corresponds to one period of a voltage signal at 50Hz. The architecture developed for this experiment is inspired on the VGG16 architecture which is frequently used in image classification tasks and has good classification rate. The VGG16 architecture reduces the dimension of the of the input signal along the architecture using the maxpooling function. A the number of samples is reduced by layer, the number of filters increases. This is done in order to maximize



51	78	87	68	74	46
61	94	83	66	74	70
79	75	90	108	116	88
49	48	83	83	77	70
85	66	88	109	98	63
40	37	37	66	60	29

94	87	74
79	108	116
85	109	98

Figure 4.9: 2D Max-Pooling

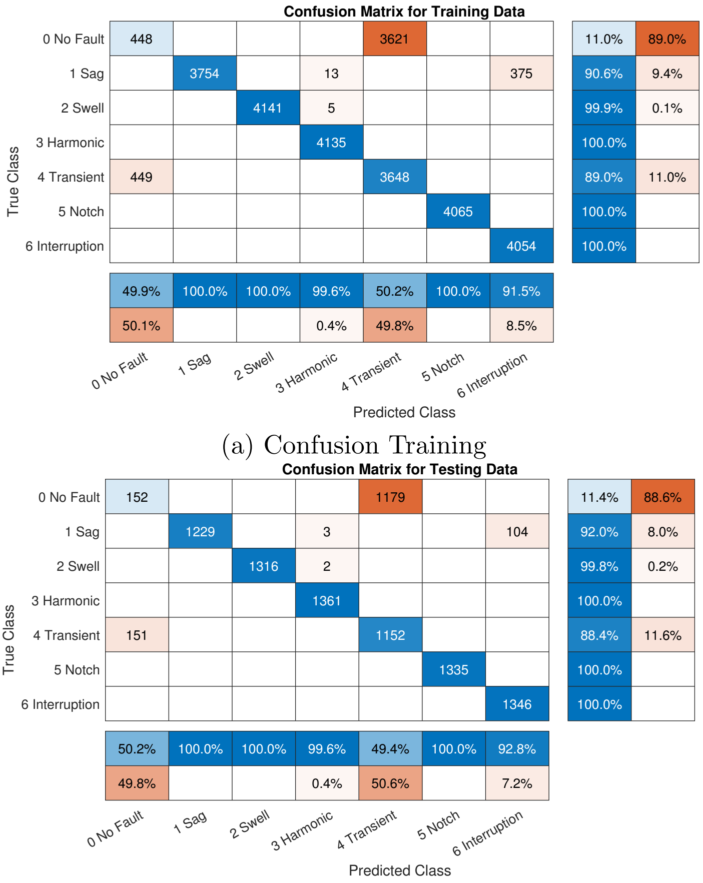
the feature extraction capabilities while reducing the processing time during training and during the inferencing. The architecture is divided by blocks. In each block the algorithm computes the convolution in the convolutional layer, then the batch is normalized using the batch normalization layer, an activation function is applied using the Relu Layer and finally the Maxpooling layer which outputs a summary statistics of the results. The architecture starts in the first block with 64 filters in the convolutional layer and reduces the input signal by half i.e. from 160 to 80. The second block, has an input dimension of 80 data point and the filter number increases by 2 i.e. 128 filters. Skipping the maxpooling layer, the third block starts with 80 data point but increases the number of filters to 256. Again ,skipping the maxpooling layer, the fourth block starts with 80 data point but increases the number of filters to 512. The fith block reduces the input size by half i.e. to 40 datapoints and increases the number of filters to 1024. Finally, before the classification section, the input dimension is decreased to 20, or by half, and the number of filters is increased to 2048. Finally the classification layer with 6 outputs. For the classification, since it is a multiclass problem, a softmax function is used. The softmax function turns the outputs to a Probability Density Function (PDF) in which the sum of all the outputs equals to 1. The class with higher probability is the class that is classified in that specific forward pass. The classification process is shown in Section 4.3.3.

Table 4.2 CNN Architecture

Description	**Output Shape**
Input	[1 160 1]
[1 3] Convolution 64 filters	[1 160 64]
Batch Normalization	[1 160 64]
Relu	[1 160 64]
[1 2] Max-Pooling	[1 80 64]
[1 3] Convolution 128 filters	[1 80 128]
Batch Normalization	[1 80 128]
Relu	[1 80 128]
[1 3] Convolution 256 filters	[1 80 256]
Batch Normalization	[1 80 256]
Relu	[1 80 256]
[1 3] Convolution 512 filters	[1 80 512]
Batch Normalization	[1 80 512]
Relu	[1 80 512]
[1 2] Max-Pooling	[1 40 512]
[1 3] Convolution 1024 filters	[1 40 1024]
Batch Normalization	[1 40 1024]
Relu	[1 40 1024]
[1 2] Max-Pooling	[1 20 1024]
[1 3] Convolution 2048 filters	[1 20 2048]
Batch Normalization	[1 20 2048]
Relu	[1 20 2048]
Fully Connected Layer	[1 1 6]

4.3.3 Classification

The classification is done by a feedforward neural network. After the batch-normalization a Rectifing linear unit **(ReLu)** is added in order to eliminate the negative values. Then, the feedforward neural network is added with 6 outputs corresponding to the 6 classes or disturbances studied in this work. The output of the neural network is then passed through the softmax function in order to convert the output to a probability distribution. The maximum probaility will determine the class or disturbance found in that window frame. The training of the CNN gave an accuracy of 84.58% which is an improvement with respect to the LSTM. The problem with this architecture is that it classified 89% of the No Fault as a Transient from the training dataset and an 88.6% from the validation. Again, it showed confusion between the two classes as shown in the precision and recall plots. Results are shown in Figure 4.10.



(a) Confusion Training

(b) Confusion Validation

Figure 4.10: CNN Confusion Matrix for Training (a) and Validation (b)

4.4 CNN-LSTM

This architecture mixes the CNN with the LSTM. In order to do this a sequence folding layer right after the input layer is added. The sequence folding layer converts a batch of data sequences to a batch of data. After this layer, the CNN comes into play. After the CNN, there is a sequence unfolding layer used to convert the batch of data in a batch of sequenced data. The sequence data are the input to the LSTM. Before the LSTM layer, there is a flattening layer that reshapes the input data to the input of the LSTM layer. Then a fully connected layer, a soft max and finally a classification layer are added respectively.By adjusting the parameters of the above mentioned architecture the CNN-LSTM with adjusted hyperparameters is obtained.

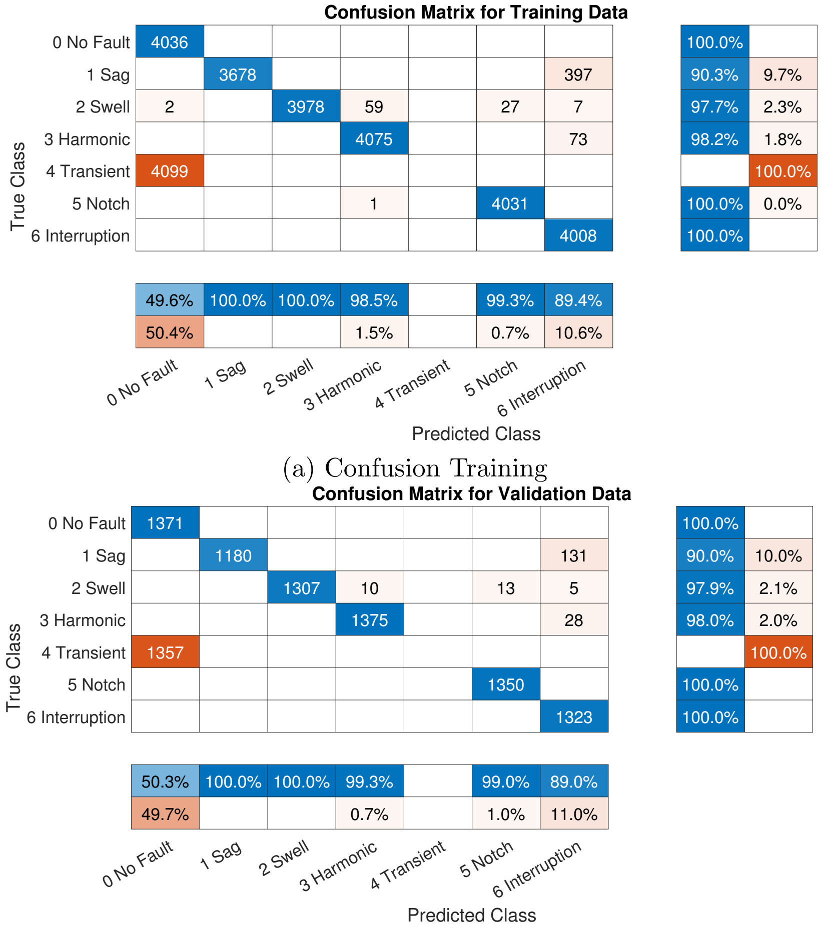
4.4.1 Architecture

Table 4.3 CNN LSTM architecture

Description	**Output Shape**
Input	[1 160 1]
Folding	[1 160 1]
[1 3] Convolution 20 filters	[1 160 20]
Batch Normalization	[1 160 20]
Relu	[1 160 20]
Unfolding	[1 160 20]
Flatten	[1 3200 1]
LSTM	[1 100 1]
Fully Connected Layer	[1 1 6]

4.4.2 Classification

As regards to the the hybrid CNNLSTM, two similar strategies were tested. The first strategy consisted in joining the LSTM and CNN that were used in the previous experiment and the second was using the combined architectures while adjusting the hyperparameters. The first strategy, exploiting the CNN-LSTM, resulted in an improvement of the classification performances of almost all of the the disturbances except for the Transient which resulted in 100% misclassification as with the LSTM. The other disturbances misclassifications ranged between 2% to 10%, which was an improvement. Concerning the second strategy exploiting the hybrid architecture, a significant improvement on the Transient response recognition was reached resulting in a 51.1% misclassification with the No Fault condition. In both, precision and recall, it showed more or less a 50% chance of miss-classification. For the other disturbances the misclassification ranges between 1.3% and 4.8% which is also an improvement. Results are shown in Figure 4.11. The hybrid CNN-LSTM with adjusted hyperparameters was able to detect the Transient disturbance in 48.9% of the signals where the transient was present. The other architectures failed completely in this task. Furthermore, this architecture obtained better results on the other disturbance classifications. Results are shown in Figure 4.12.

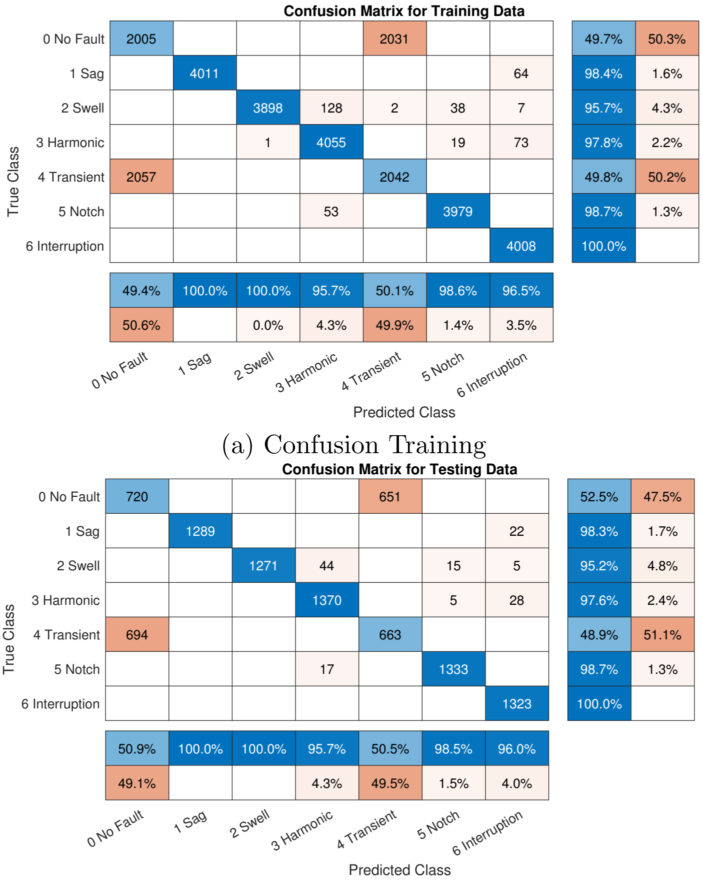


(a) Confusion Training

(b) Confusion Validation

Figure 4.11: CNN-LSTM Confusion Matrix for Training (a) and Validation (b)

LSTM CNN with adjusted hyperparameters



True Class	0 No Fault	1 Sag	2 Swell	3 Harmonic	4 Transient	5 Notch	6 Interruption		
0 No Fault	2005				2031			49.7%	50.3%
1 Sag		4011					64	98.4%	1.6%
2 Swell			3898	128	2	38	7	95.7%	4.3%
3 Harmonic			1	4055		19	73	97.8%	2.2%
4 Transient	2057				2042			49.8%	50.2%
5 Notch				53		3979		98.7%	1.3%
6 Interruption							4008	100.0%	
	49.4%	100.0%	100.0%	95.7%	50.1%	98.6%	96.5%		
	50.6%		0.0%	4.3%	49.9%	1.4%	3.5%		

(a) Confusion Training

True Class	0 No Fault	1 Sag	2 Swell	3 Harmonic	4 Transient	5 Notch	6 Interruption		
0 No Fault	720				651			52.5%	47.5%
1 Sag		1289					22	98.3%	1.7%
2 Swell			1271	44		15	5	95.2%	4.8%
3 Harmonic				1370		5	28	97.6%	2.4%
4 Transient	694				663			48.9%	51.1%
5 Notch				17		1333		98.7%	1.3%
6 Interruption							1323	100.0%	
	50.9%	100.0%	100.0%	95.7%	50.5%	98.5%	96.0%		
	49.1%			4.3%	49.5%	1.5%	4.0%		

(b) Confusion Validation

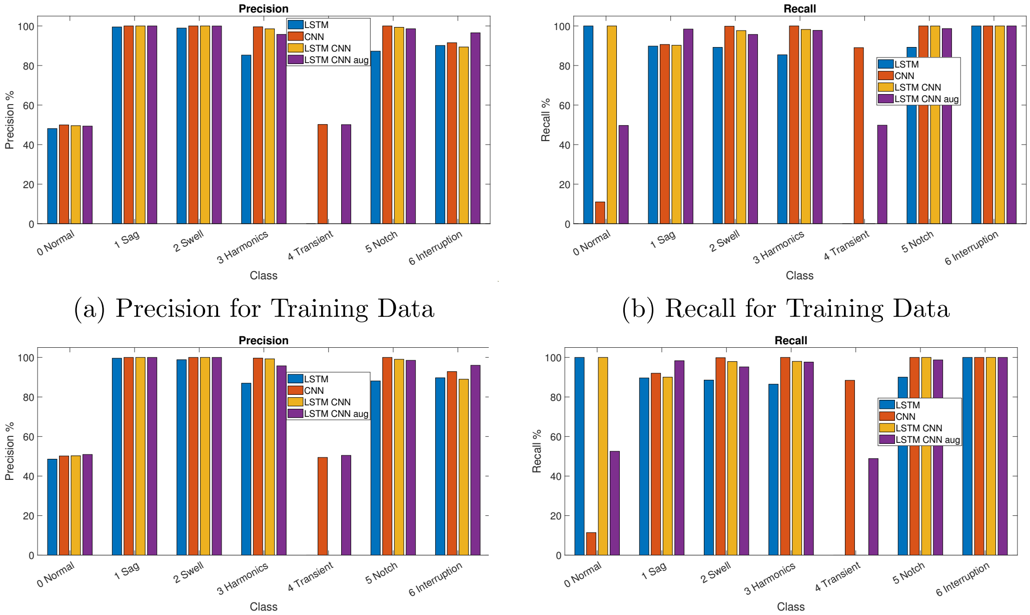
Figure 4.12: CNN-LSTM with hyperparameter adjustment Confusion Matrix for Training (a) and Validation (b)

4.5 Comparison

The different deep learning architectures were trained, validated and tested. A comparison was made using the training and validation results by calculating the confusion matrix and extracting the precision and recall values. Tests were made using an experimental test bench to determine the effectiveness of the deep learning architectures using experimental measurements. An event signal was generated containing a signal with no disturbance followed by a sag and an interruption and finally a reestablishment of the voltage. This signal comes from the power source containing harmonic some components. Two other signals were generated one containing a sudden interruption and the other containing a sudden voltage drop or sag. These tests were conducted by classifying the signal by parts using a sliding window method. This comparison determines the most suitable architecture for a real time detection and classification of power quality disturbances using a sliding window method. The parameters used in this comparison is the Precision and Recall for the training and validation dataset and the a positive classification rate in the testing phase using experimental data.

4.5.1 Precision Recall

On each of the confusion matrix of the compared architectures, the precision and the recall was calculated. The precision of a classifier is defined as the number of retrieved relevant items as a proportion of the number of retrieved items for any given class [82]. In other words, it is the ratio between the positive identifications that are actually correct and the entire set of positive identifications of any given class in the dataset. Recall, on the other hand, is defined as the number of retrieved relevant items as a proportion of all the relevant items, for any given retrieved set [82]. In other words, the proportion of the actual positives that where identified correctly. The comparison of precision and recall of different architectures are shown in Figure 4.13. The precision and recall of the LSTM-CNN with adjusted hyperparameters had superior results with respect to the other architectures due to the fact that it had better scores for classifying the transient in both training and testing. From this experiment it can be said the LSTM CNN had the better results and thus can be said it is the most suitable for this type of application.



(a) Precision for Training Data

(b) Recall for Training Data

(c) Precision for Testing Data

(d) Recall for Testing Data

Figure 4.13: Bar chart of the comparison of the precision and recall of the LSTM (blue), CNN (red), LSTM-CNN(yellow) and LSTM-CNN with adjusted hyperparameters(purple). (a) Precision Training Data, (b) Recall Training Data, (c) Precision Testing Data, (d) Recall Testing Data.

4.5.2 Classification

The event signal, shown in Figure 4.14, Figure 4.15, Figure 4.16 and Figure 4.17 , is a voltage signal with a harmonic distortion, sag and an interruption. Each architecture had good results when tested using this signal. However, the LSTM classified the harmonics as a no fault and misclassified a section as a notch. The LSTM-CNN also showed the same problem. On the other hand the CNN misclassified the harmonic disturbance as a transient disturbance. While all architectures successfully classified only the Interruption and the Sag in the testing, the LSTM-CNN with adjusted hyperparameters was the one that had better results because it classified the harmonics, sag, interruption correctly without misclasification.

The LSTM had 4 missclassifications where it classified a sag where there was no disturbance. It also classified a notch where there was a sag. It classified a harmonic distortion along with a sag when there was no disturbance. The transition between the normal voltage signal to the disturbance injects high frequency components to the classification window that the LSTM classifies that section of the signal as a notch disturbance or a harmonic distortion. The overall classification rate for this test is 94.7%.

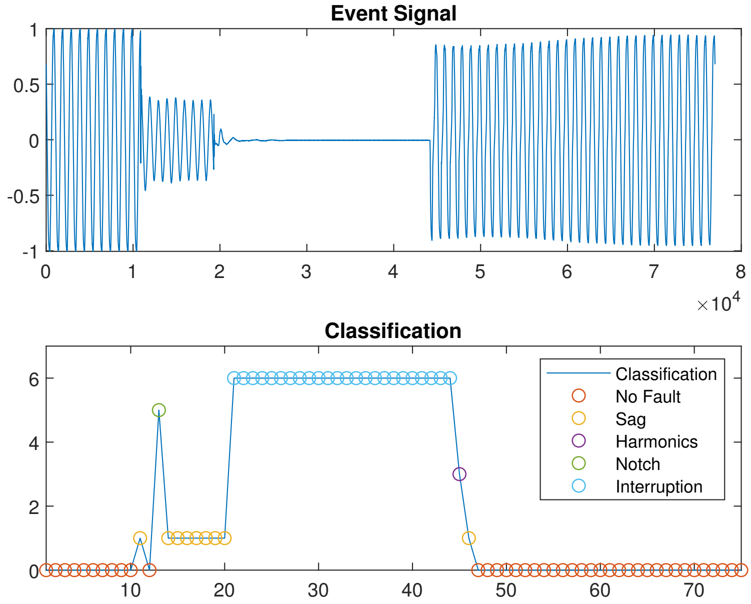


Figure 4.14: LSTM Testing Results

The CNN had only 2 missclassifications. During a Sag, it classified a harmonic distortion and it classified a transient when there was no disturbance at the end. The window length for the CNN was larger than the other architectures. This is due in part because the transition between the normal functioning of the voltage supply and a disturbance. The transition between the normal voltage signal to the disturbance injects high frequency components to the classification window that the CNN classifies that section of the signal as a harmonic disturbance. The overall classification rate of the CNN for this test is 93.3%.

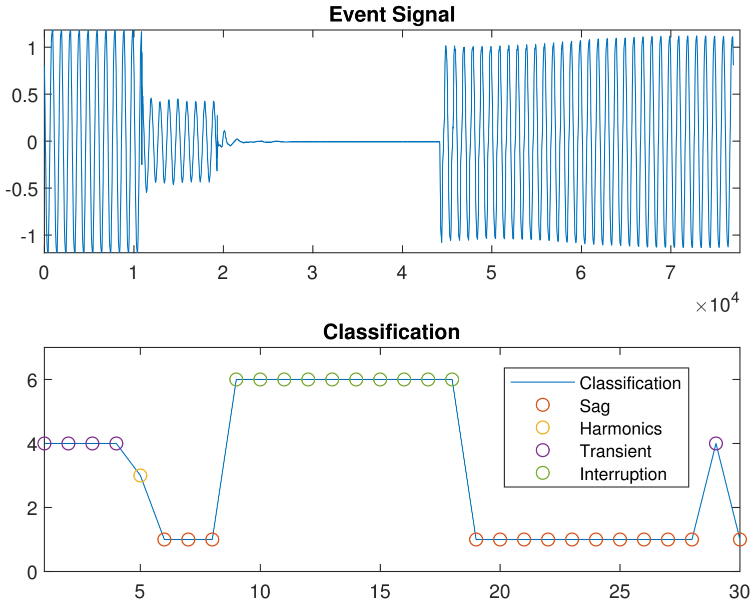


Figure 4.15: CNN Testing Results

For the hybrid CNN-LSTM, results gave 4 missclassifications. During a sag, it classified a notch. It classified a notch, swell and then another notch during no disturbance. The CNN-LSTM suffers the same missclassifications as the LSTM and the CNN. The transition between the normal voltage signal to the disturbance injects high frequency components to the classification window that the CNN-LSTM classifies that section of the signal as a notch disturbance. The overall classification rate of the CNN-LSTM for this test is 93.3% which are good results if the spectral leakage is not considered and this architecture can be considered for a real time application due to its simplicity in implementation and low number of layers and parameters.

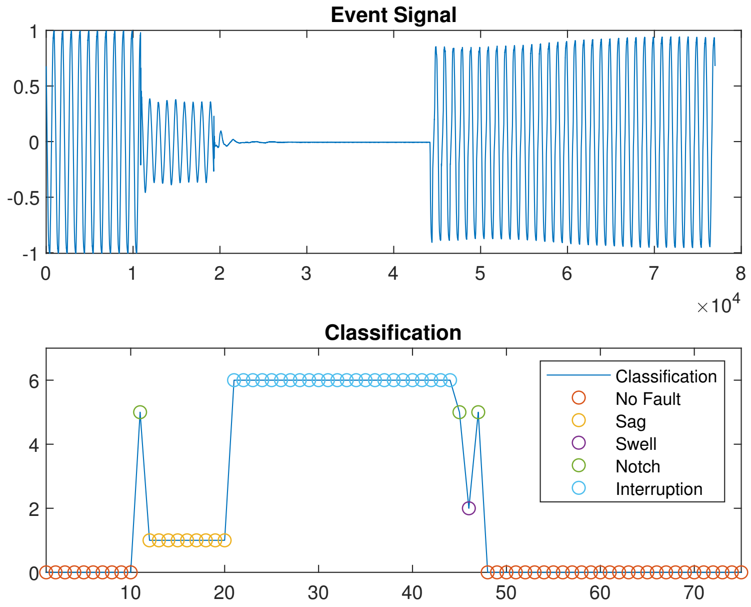


Figure 4.16: CNNLSTM Testing Results

The final test done in this comparison is the hybrid CNN-LSTM with adjusted hyperparameters. For the hybrid CNN-LSTM with adjusted hyperparameters, results gave just 2 missclassifications. During a sag, it classified a swell and during an interruption it classified a transient. This is due to the transition between normal to disturbance and transition between disturbances. These transitions inject high frequency components in the signal that leads the deep learning algorithm to a missclassification. Clearly this was the best results making it the most precise architecture. It also identified the nartural harmonics found in common voltage supply. Which is totally normal due to the natural harmonic components injected in the power grid. The overall classification rate for this test is 97.3% if the harmonic distortions are seen as a normal voltage supply which makes it the most successful of the architectures compared in this chapter.

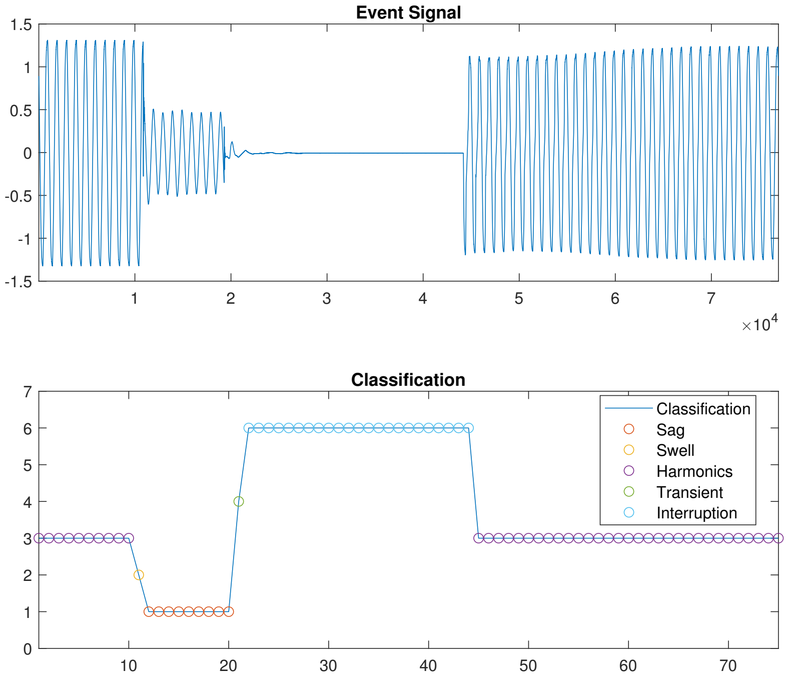


Figure 4.17: CNNLSTM with adjusted hyperparameters Testing Results

Other tests were conducted with experimental datasets using the test bench shown in Figure 8 in order to compare and prove the effectiveness of all the architectures previously mentioned. It has been possible to generate several experimental datasets of the interruption and of the sag disturbances. The experimental measurments are shown in Figure 4.19 and Figure 4.20 along with the plots of the classification results below. Each classification point consist of 1000 samples of the measured signal. Once again, the CNN-LSTM with adjusted hyperparameters was the most consistent in classifying all the disturbances without misclassification. As mentioned before, the identification algorithms were tested with exerimental datasets containing interruption and sag disturbances. Concerning the sag disturbance, all of the four architectures performed correct identification. Some misclassifications occurred when testing the interruption disturbance with the CNN-LSTM and with the LSTM. Combining these results with the ones previously mentioned, the CNN-LSTM with adjusted hyperparameters is the is the one which performed best.

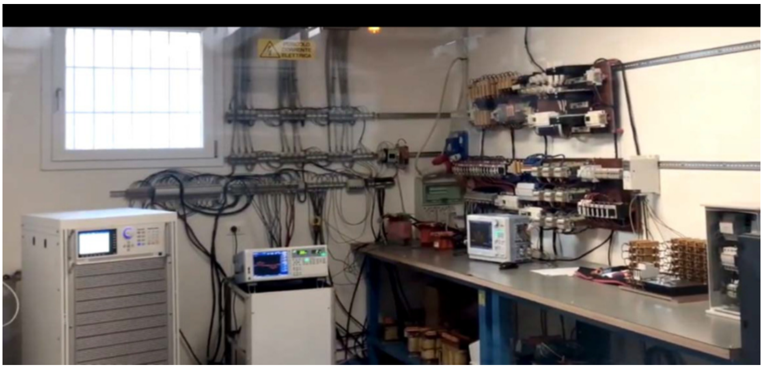


Figure 4.18: Test Bench

Other tests were conducted with experimental datasets using the test bench shown in Figure 4.18 in order to compare and prove the effectiveness of all the architectures previously mentioned. It has been possible to generate several experimental datasets of the interruption and of the sag disturbances. The experimental measurements are shown in Figure 4.19 and Figure 4.20 along with the plots of the classification results below. Each classification point consist of 1000 samples of the measured signal. Once again, the CNN-LSTM with adjusted hyperparameters was the most consistent in classifying all the disturbances without misclassification. As mentioned before, the identification algorithms were tested with exerimental datasets containing interruption and sag disturbances. Concerning the sag disturbance, all of the four architectures performed correct identification. Some misclassifications occurred when testing the interruption disturbance with the CNN-LSTM and with the LSTM. Combining these results with the ones previously mentioned, the CNN-LSTM with adjusted hyperparameters is the is the one which performed best.

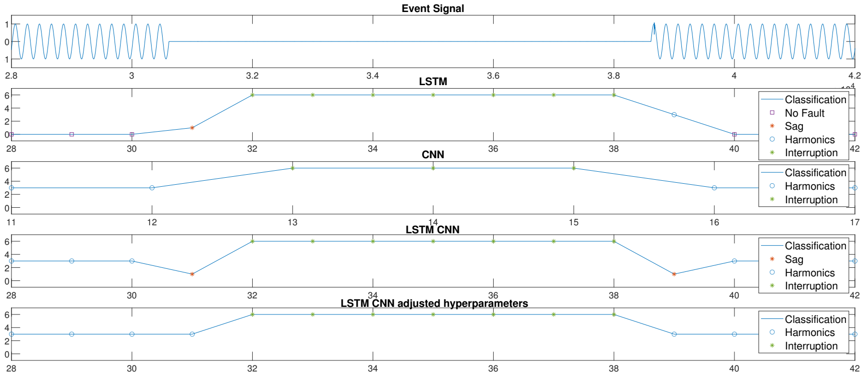


Figure 4.19: Voltage signal with an interruption measured on the test bench (top plot). From top to bottom, the classification performances of each architecture: LSTM, CNN, LSTM-CNN and the LSTM-CNN with adjusted hyperparameters.

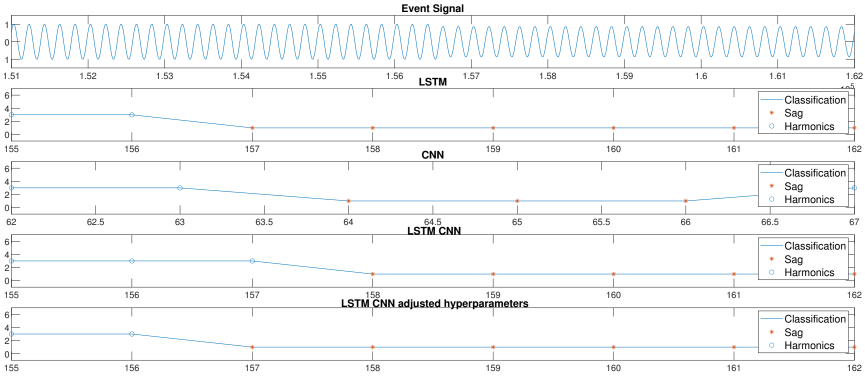


Figure 4.20: Voltage signal with an sag measured on the test bench (top plot). From top to bottom, the classification performances of each architecture: LSTM, CNN, LSTM-CNN and the LSTM-CNN with adjusted hyperparameters.

4.6 Conclusion

This work investigates the effectiveness of various deep learning architectures for Power Quality disturbances detection and classification. To do this, it is imperative to study the mechanism of these algorithms to extract the unique features of each disturbance and obtaining an efficient and accurate classification. The training and validation of deep learning architectures depend on a large number of data to better generalize the classification results. A Matlab/Simulink model has been designed and implemented in order to generate these disturbances. To improve the classification performances of the strategies under evaluation and converge to a generalized result, the data in the simulated dataset was augmented. Using the resulting datasets the authors have proposed a comparison among the LSTM, the CNN and a joint architecture that uses both the LSTM and CNN. All of the architectures were trained and validated using the augmented datasets and then tested using experimental data. Concerning the experimental validation of the algorithms, it has been possible to generate an experimental dataset of the interruption and of the sag disturbances. The two datasets were processed by exploiting the four previously mentioned architectures. The first signal contained a train of interruptions and the second signal a train of sags. All of the four architectures successfully classified the sag signal. There were some discrepancies between the architectures while classifying the signal containing interruptions. Again, the LSTM-CNN with adjusted hyperparameters proved to be superior in classifying the disturbances. These results show that it is possible to train deep learning architectures with simulated data and operate disturbance identification on experimental data. The transient disturbance appears to be hardly detectable for all of the architectures under evaluation, mainly due to the small duration of the disturbance. The architecture that best performed while classifying the transient disturbance was the LSTM-CNN with adjusted hyperparameters. Furthermore, concerning the classifications of other disturbances, the LSTM-CNN with adjusted hyperparameters was the most performing one, both considering the simulated and the experimental datasets.

Chapter 5

Comparison of Power Quality Detection and Classification Algorithms in the Frequency Domain

In this chapter a comparison of two machine learning algorithms are presented classifying the PQD in the frequency and time-frequency domain. These algorithms are the Multi-layer multi value network (MLMVN) and the Short-Time Fourier Transform Convolutional Neural Network (STFT-CNN). Section 1 is a brief introduction were the experiments and comparisons are briefly explained. Section 2 explains the MLMVNs functioning. Section 3 describes the CNN used in this chapter and how the classification is done. Section 4 explains data preparation for the MLMVN and the STFT-CNN and the training procedure. In Section 5, the training results are presented for the MLMVN and the CNN followed by the experimental results for both algorithms. Section 6 presents the comparison of both algorithms. Finally, a conclusion is presented in Section 7. [1]

[1] This chapter has been published as "Power Quality Analysis Based on Machine Learning Methods for Low Voltage Electrical Distribution Lines" in *Electric Power Systems Research, 2022* [83].

5.1 Introduction

Feature extraction methods have been extensively used as a means of obtaining the most significant parts of a dataset. The objective is to extract the frequency components of the dataset in order to maximize the perrformance of Machine Learning algorithms.
This chapter presents two classification methods capable of detecting electrical disturbances in low voltage grids. Both approaches rely on machine learning techniques that classify voltage signals in the frequency domain. The first technique here proposed uses the Fourier Transform (FT) of the voltage waveform and classifies the corresponding complex coefficients through a Multilayer neural network with Multi-Valued Neurons (MLMVN). In this case, the structure of the classifier has three layers and a small number of neurons in the hidden layer. Therefore, the computational effort is very low, the learning time is short, and no coding operations are necessary because the neural network can process complex-valued inputs. The second technique involves the use of the Short Time Fourier Transform (STFT) and a Convolutional Neural Network (CNN) with 2-D convolutions in each layer for feature extraction and the reduction of dimensionality. The five disturbances considered in this paper are: voltage sag, voltage swell, harmonic distortion, voltage notch and interruption. The performances of the two classifiers are compared during the training phase using simulated data and subsequently through experimental measurements, obtained from an artificial generator of disturbances and a variable load. Both techniques represent an innovative approach to this type of problem and guarantee excellent classification results.

5.2 Complex Neural Network

One of the most innovative aspects presented in this paper is the use of a MLMVN in the classification of electrical disturbances. This paper represents the first application of a neural classifier based on Multi-Valued neurons on the field of power quality evaluation. The MLMVN structure used in this work is the classic three-layer configuration presented in [84] while the use of binary neurons in the output layer, the introduction of the Winner Takes All rule and the choice of processing complex coefficients obtained through the FFT of the sampled voltage waveforms are specific aspects of this application. This type of neural network is based on a feed forward neural network structure and a derivative free backpropagation procedure during the training phase [84]. The absence of derivative terms makes the correction of the weights very fast compared to other machine learning techniques. Also, the complex nature of MLMVN makes it easily adaptable to electrical problems. In fact, the electrical quantities in power transmission and distribution grids are characterized by alternating waveforms and therefore are represented by phasors. Since each electrical standard has a single frequency value, line quantities can be expressed as complex numbers characterized by magnitude and phase. For these reasons, MLMVN has been used with good results in failure prevention for electrical infrastructures [85] and analog circuits [86]. From a general point of view, this classifier is a three-layer neural network in which the elementary unit is the Multi-Valued Neuron (MVN) described in [84] and the inputs and weights are complex numbers. Figure 5.1 shows the global structure of the MLMVN where, for example, $W_i^{(}k, m)$ is the $i - th$ complex-valued weight of the $k - th$ neuron belonging to the layer m, $N_{(}m - 1)$ is the number of the neurons belonging to the hidden layer, N_m is the number of the neurons belonging to the output layer and $(X_1, X_2, \cdots, X_n)$ are the complex-valued inputs.

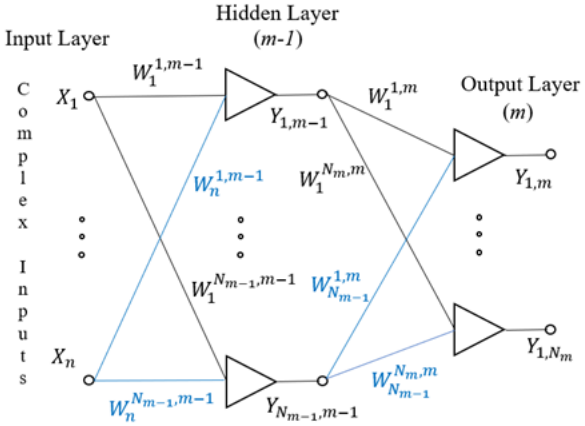


Figure 5.1: VGG-16 Architecture

5.3 Convolutional Neural Network

In this work, the use of CNNs was studied by means of a STFT. The STFT is used to extract the spectral component of the input voltage signal along with its temporal component. This is then used to classify the input voltage signal using the CNN. The STFT for classification using the CNNs has been previously used in other applications to improve CNN classification results as shown [87], [88], [89] . CNNs are feedforward neural networks that uses 2D convolutions in each layer for feature extraction and dimensionality reduction. The 2-D convolution is shown in Equation 5.1.

$$S[n_1, n_2] = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} x[m_1, m_2] k[n_1 - m_1, n_2 - m_2] \tag{5.1}$$

The CNN works by adjusting the kernel denoted by parameter k during training to find the optimum kernel weights for feature extraction of signal x for each corresponding filter in each convolutional layer. A max-pooling layer is then added to the convolutional layer to reduce the size of the image, extract the most important parts of the image, and reduce training time. Pooling layers of a CNN implement a spatial dimensionality reduction operation designed to reduce the number of trainable parameters for the next layers and allow them to focus on larger areas of the input pattern [90]. The max-pooling layer can be defined as the summary statistics of the output of the preceding convolutional layer. The max-pooling layer identifies the maximum of a given section and sets it as the reduced output of the convolutional layer.

CNNs were originally created for image classification tasks. In order to create a 2D, image-like signal, the use of a STFT was explored to convert a 1D signal into a 2D matrix. This is done to exploit the CNN image feature extraction capabilities. The STFT is a Discrete Fourier Transform in a windowed section of the signal. The STFT permits frequency analysis in the time domain using a sliding window. The STFT is an enhanced mathematical methodology, derived from the discrete Fourier transform (DFT), to explore the instantaneous frequency as well as the instantaneous amplitude of localized waves with time-varying characteristics [88]. This method allows the time signal to be converted into a time-frequency signal i.e., a 2D matrix. Some of the disturbances in power quality studied in this work involve the injection of undesired frequency components (harmonics distortion and notch). The other disturbances involve the deviation of voltage levels from their nominal values which can also be shown in the STFT shown in Chapter 3.4. The heatmap represents a yellow color for a high level of a frequency component and a blue color for a low level of a frequency component. All signals have a high level at 50Hz which corresponds to the supply voltage frequency or fundamental frequency. The harmonic distortion and the notch show other frequency components and the sag, swell and interruption show a decrease or an increment in intensity at 50Hz. The window function used in this work is the Blackman window. The equation to the Blackman window is shown in Figure 5.2.

$$X[m,n] = \sum_{k=0}^{L-1} x[k]w[k-m]e^{-j2\pi nk/L} \tag{5.2}$$

Since the classification task involves multiple classes, the CNN has the same number of classes as outputs. The output of the CNN involves a fully connected layer with 6 outputs. Each output represents each class. The output is then converted to a probabilistic density function by means of a softmax function. Softmax functions are most often used as the output of a classifier with the aim of representing the probability distribution over n different classes [81]. The softmax function converts the output of each neuron into a probability distribution as shown in Equation 5.3, where e^{x_i} is the exponential output of a given neuron and $\sum_{j=1}^{K} e^{x_j}$ is the sum of all the exponential outputs. The sum of all the outputs of the softmax equals 1. Figure 5.2 shows an example of the softmax function which classified a given voltage signal with a probability of harmonic distortion of 99% and a probability of other disturbances of 0.2%.

$$P\left(\hat{y}=j|x\right) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{5.3}$$

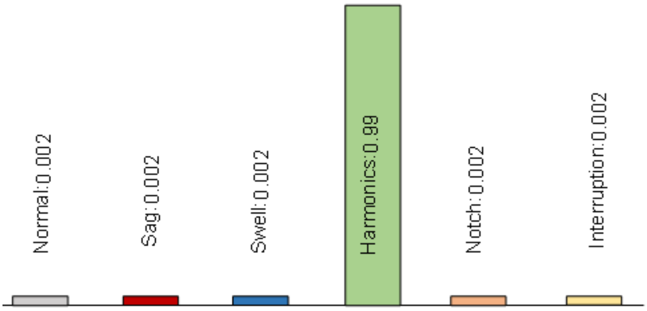


Figure 5.2: VGG-16 Architecture

Since this is a classification task and it is a multiclass problem, the loss function is a cross entropy loss shown in Equation 5.4. In this formula, $P\left(\hat{y}_i|x\right)$ is the output of the SoftMax function and y_i is the training label.

$$CE = \sum_{i=1}^{C} y_i log\left(P\left(\hat{y}_i|x\right)\right) \tag{5.4}$$

The gradient during training is calculated and the CNN is updated using the Adaptive Moment Optimizer (ADAM optimizer). The ADAM optimizer is an adaptive learning rate optimizer that uses first and second order moments of the gradients for updating the individual parameters. In this work, the input voltage signal is converted to a time-frequency matrix and then classified using the CNN. To do this, a dataset of voltage signals with disturbances is generated and transformed into its time-frequency counterpart using the STFT. The CNN is trained using the time-frequency dataset obtaining a probability of each disturbance with the SoftMax function. using the cross-entropy loss function the output is measured with the labeled classes and the weights are adjusted using the ADAM optimizer.

5.4 Results

This section presents the main results obtained during the training phase of the machine learning techniques described above. The data used during the training phase is generated by a simulation procedure on Matlab and Simulink environments. Therefore, a Matlab script is used to create a large variability of electrical disturbances in a very short time starting from the sinusoidal function of the line voltage, which is characterized by a frequency of 50 Hz and a root mean square value of 230 V. The amplitude and the frequency components of this signals are modified to create all the different disturbances following the formal definitions given in Section I. Starting from the normal sinusoidal signal shown in Fig. 1a, the value of the maximum amplitude is chosen randomly in the interval $(32.53 \div 420.02)$ V to simulate the presence of a voltage sag. This problem, in fact, causes a reduction of the phase voltage between 10% and 90% of the nominal value. Similarly, examples of voltage swell are created by considering increases in the maximum amplitude from 10% to 50% of the nominal value. As for the harmonic disturbances, signals with frequencies multiple of the fundamental (50 Hz) are generated up to the eleventh harmonic and then added to the line voltage. Notch is a condition when the magnitude of voltage decreases towards zero for a short period of time, usually microseconds. This condition is simulated in Matlab by adding impulsive components at specific instants of the nominal voltage waveform. Finally, interruptions are simulated by reducing the maximum voltage value below 10% of the nominal value. Furthermore, other examples of PQDs are generated through the Simulink model proposed in [29] by the Authors. In this way, it is possible to simulate with a high level of accuracy distortions caused by faults in low voltage distribution networks. A waveform with a duration of three periods (60 ms) is created for each signal and 250 examples are generated for each fault class (nominal condition, voltage sag, voltage swell, harmonic distortion, notch, interruption). 200 of these examples have been generated using the Matlab script and the remaining 50 using Simulink. Therefore, a set of 1500 simulated signals are used to train the neural classifiers described above. The simulated voltage waveforms are sampled with a frequency of 8 kHz resulting in 480 samples for each example. It should be noted that the training procedure is divided into two phases: the learning phase and validation phase. During the learning phase 80% of the dataset is chosen randomly and used for the correction of the weights. Subsequently, the remaining 20% is used in validation to verify the classification results. In both phases, the index used to evaluate the performance is called Classification Rate (CR) and it corresponds to the ratio between correctly classified samples and the total number of processed samples.

5.4.1 Complex Neural Network

The MLMVN training procedure requires a matrix like dataset shown in Equation 5.5, containing a large variability of electrical disturbances expressed in the frequency domain. Therefore, the Discrete Fourier Transform is applied using a Fast Fourier Transform algorithm to the samples of the voltage waveforms generated in Matlab. In this paper 256 points are considered for the DFT, and the corresponding complex values are used as inputs of the MLMVN.

In this paper, the complex-valued inputs $(X_1, X_2, \ldots,\ X_n)$ are obtained from the Discrete Fourier Transform (DFT) of the sampled line voltage with a frequency of 8 kHz. During the training phase, the time-domain samples of the waveforms are processed using a Fast Fourier Transform (FFT) algorithm and each complex term obtained is used as an input of the MLMVN. Figure 5.3 summarizes this procedure.

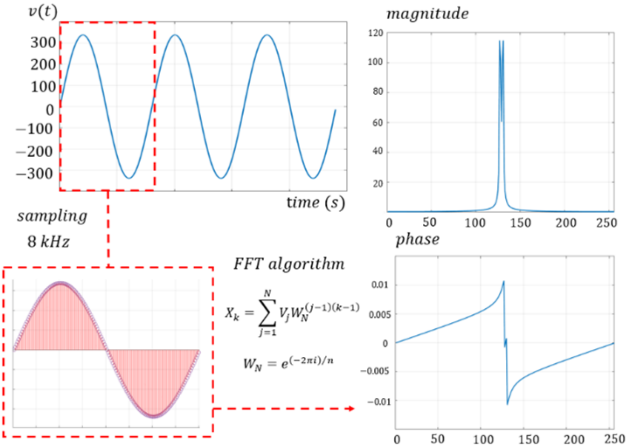


Figure 5.3: Fast Fourier Transform (FFT) algorithm

Since the correction of the weights is based on a supervised learning algorithm, many sample signals must be used with the corresponding desired classifications. Therefore, the structure of the dataset matrix used during the training phase is

$$\begin{bmatrix} X_1^{(1)} & X_2^{(1)} & \cdots & X_n^{(1)} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_1^{(N_S)} & X_2^{(N_S)} & \cdots & X_n^{(N_S)} & 5 \end{bmatrix} \tag{5.5}$$

where the last column contains the indexes of the fault classes, n is the number of points used in the fast Fourier transform algorithm and N_S is the total number of examples. Each termX_k $(k = 1, \ldots,\ n)$ is calculated as

$$X_k = \sum_{j=1}^{N} V_j W_N^{(j-1)(k-1)} \tag{5.6}$$

in which V_j is a voltage sample and W_N is obtained using

$$W_N = e^{(-2\pi i)/n} \tag{5.7}$$

Once the complex-valued inputs are calculated, all the weights are initialized to random values and the dataset matrix shown in Equation 5.5 is processed one row at a time. The element in the last column of each sample is used to calculate the corresponding desired output D, while the inputs $(X_1, X_2, \ldots,\ X_n)$ are processed through the two layers of neurons of the MLMVN. Neurons belonging to the hidden layer are characterized by a continuous activation function as shown in Equation 5.8

$$P\left(z\right) = e^{i\ Arg(z)} = \frac{z}{|z|} \tag{5.8}$$

where z is the weighted sum of the inputs in Equation 5.9,

$$z = W_0 X_0 + \sum_{i=1}^{n} W_i X_1 \tag{5.9}$$

W_i is the $i - th$ weight of the considered neuron and X_i represents the corresponding $i - th$ input.

On the other hand, the output layer of the MLMVN contains only discrete neurons, which have a finite number of possible outputs. Each of these neurons divides the complex plane into k equal sectors and the output corresponds to the lower border of the sector containing z. From a mathematical point of view, given the total number of sectors k, the output of the neuron is equal to the lower limit of the jth sector if the argument of the weighted sum is between $2\pi k$ and $2\pi(j+1)/k$

$$P\left(Z\right) = Y = \varepsilon_k^j = e^{i2\pi j/k} \quad if \ 2\pi/k \leq \arg\left(z\right) < 2\pi(j+1)/k \tag{5.10}$$

Therefore, the combination of the output neurons is used to define the global classification results. In this sense, it is necessary to mention that the output neurons used in this paper are binary. This is a specific solution chosen for this kind of application which involves the use of a neuron for each electrical disturbance in the output layer. Therefore, each neuron has only two possible outputs: $(1+j0)$ or $(-1+j0)$. The first value corresponds to the lower border of the sector $[0\pi)$ while the second term is that of the interval $[\pi, 2\pi)$. This setting allows the reduction of misclassifications between consequential sectors but requires the introduction of a specific method for selecting outputs. In fact, the single failure hypothesis is assumed, and this means that only one neuron can be activated by detecting the corresponding disturbance. If, during the training phase, more than one neuron is activated, the Winner Takes All rule is used. This means that only the neuron with the lowest error is kept in the activated state. Therefore, each output neuron is associated with a specific voltage disturbance and the upper half plane $[0\pi)$ is used to describe its absence while the lower half plane $[\pi, 2\pi)$ is used to indicate the problem. For example, the first neuron belonging to the output layer focuses on sensing the voltage sag, as shown in Fig. 8. Therefore, one neuron for each disturbance is used in the output layer. To facilitate the interpretation of the results, the first sector of each output neuron is encoded by the value 0 while the second sector by the value 1. Table 5.1 summarizes the organization of the output layer.

Table 5.1 Output Neurons and Disturbance Classes

Fault Class	Description	Output Combination					
0	No Disturbance	0	0	0	0	0	0
1	Voltage Sag	1	0	0	0	0	0
2	Voltage Swell	0	1	0	0	0	0
3	Harmonic Distortion	0	0	1	0	0	0
4	Voltage Notch	0	0	0	1	0	0
5	Interruption	0	0	0	0	0	1

As said before, MLMVN falls in the category of feedforward neural networks and the training is performed in a supervised manner. The first step in this procedure is to map the correct combination of desired outputs to each example belonging to the dataset. All the outputs equaled to 0 are converted into the complex number $(1 + j0)$ while the outputs equaled to 1 become $(1 - j0)$. These values are used to calculate output errors and initiate the backpropagation procedure. Given $D^s_{k,m}$ the desired output of the kth neuron belonging to layer m obtained by processing the sample s $(s = 1, \ldots,\ N_s)$, the corresponding error is the difference between $D^s_{k,m}$ and the current output $Y^s_{k,m}$. These values are normalized with respect to the number of neurons of the previous layer

$$\delta^s_{k,m} = \frac{D^s_{k,m} - Y^s_{k,m}}{N_{m-1} + 1} \tag{5.11}$$

These errors, calculated on the output neurons, are backpropagated from the last layer to the input one through the mathematical rule shown in Equation 13, as shown in [33].

$$\delta^s_{k,m-1} = \frac{1}{N_{m-1} + 1} \sum_{i=1}^{n_m} \delta^s_{i,m} (W^{i,m}_k)^{-1} \tag{5.12}$$

This standard correction procedure allows the adjustment of the weights by using Equation 5.13,

$$\Delta W^{k,m}_i = \frac{\alpha_{k,m}}{(Nm - 1 + 1)|z^s_{k,m}|} \delta^s_{k,m} \bar{Y}^s_{i,m-1} \tag{5.13}$$

where $\Delta W^{k,m}_i$ is the correction for the $i - th$ weight of the $k - th$ neuron belonging to the layer m, $\alpha_{k,m}$ is the corresponding learning rate, $nm - 1$ is the number of the inputs equal to the number of outputs of the previous layer, $\left|z^s_{k,m}\right|$ is the magnitude of the current weighted sum, $\delta^s_{k,m}$ is the output error and $\bar{Y}^s_{i,m-1}$ is the conjugate-transposed of the input for the output layer neurons and inner hidden layer neurons (if any) or a reciprocal input for the first hidden layer neurons. Equation 5.13 is used individually for each weight, and it represents the main difference between neural networks based on Multi-Valued Neurons (MVNs) and those based on real-valued neurons, because it doesnt contain derivative terms. This guarantees the low computational cost and very fast training phase of MLMVN compared to other algorithms. To obtain a further reduction of the training time the standard correction rules have been replaced with a batch algorithm [91]. In this case, the output errors are calculated as shown in Equation 5.11 and backpropagated as shown in Equation 5.12 for each row of the dataset without adjusting the weights. Once all the examples belonging to the dataset have been processed and the corresponding errors have been defined, i.e., at the end of each training epoch, the corrections of the weights are calculated through a batch algorithm such as the QR decomposition. Each error is then saved in a specific matrix

$$\begin{bmatrix} \delta^1_{1,m} & \delta^1_{2,m} & \cdots & \delta^1_{n,m} \\ \delta^2_{1,m} & \delta^2_{2,m} & \cdots & \delta^2_{n,m} \\ \vdots & \vdots & \vdots & \vdots \\ \delta^{N_s}_{1,m} & \delta^{N_s}_{2,m} & \cdots & \delta^{N_s}_{n,m} \end{bmatrix} \tag{5.14}$$

and a corresponding oversized system can be written as shown in Equation 5.15, because the number of samples is greater than the number of corrections representing the unknowns.

$$Y \Delta W k = \delta k \tag{5.15}$$

This system must be solved through a Linear Least Square (LLS) method obtaining the best corrections in order to meet the condition as shown in Equation 5.16.

$$\Delta W_k = argmin||Y \Delta W_k - \delta^k||2 = Y \delta^k \tag{5.16}$$

where the superscript k indicates the number of the neuron considered, $Y^* = (Y^* Y)^{-1} Y^T$ is the pseudo-inverse of the matrix Y and Y^T is its conjugate transpose. In this work, QR decomposition is used, and the error matrix shown in Equation 15 for the hidden layer consists of the backpropagated terms. To improve the classification performance of the MLMVN proposed in this paper, the soft margin rule is adopted [92]. In this case, the training phase is changed to bring the weighted sums as close as possible to the bisector of the desired sectors. This technique avoids misclassification of the z terms that fall close to the edge between two successive sectors. From the computational point of view, there are no differences compared to the standard procedure, because the only change is the use of bisectors as desired outputs $D^s_{k,m}$. Therefore, the goal of weight correction is not only the positioning of the output in the correct sector, but also the minimization of the distance with respect to the bisector of that sector.

5.4.2 Convolutional Neural Network

The CNN training procedure requires a dataset containing the electrical voltage disturbance in their time-frequency domain. Therefore, the simulated voltage disturbances were converted to their time-frequency domain via the STFT as shown in Section II.A. The used CNN architecture has an input of 500 rows and 5 columns. This means 500 frequency components and 5 cycles. The CNN reduces the input using the max-pooling layers and at the same time increases the filter size. The training was done, and the precision and recall were calculated for each class. The precision of the CNN resulted in 100% in all classes except the normal which had a classification rate of 99.3% with the training data and a 98.9% with the validation data. The results are shown in Table 5.4. The recall resulted in 100% classification rate in all except for the sag and swell which resulted in a 99.6% and 99.7% for the training for the sag and swell respectively and 99.7% and 99.2% for the validation for the sag and swell respectively. The results are shown in Table 5.5. The CNN has an overall accuracy of 99.89% for the training dataset and overall accuracy of 99.82% for the validation dataset.

5.5 Results

This section proposes further validation by using real voltage measurements to highlight the advantages and disadvantages of both techniques. Therefore, different examples of disturbances were generated by the two experimental setups described above, and the corresponding voltage waveforms were sampled with a frequency of 8 kHz. Finally, a comparison of a specific sequence of electrical disturbances is presented.

5.5.1 Complex Neural Network

Training Results

The MLMVN training procedure requires a matrix like dataset shown in Equation 5.5, containing a large variability of electrical disturbances expressed in the frequency domain. Therefore, the Discrete Fourier Transform is applied using a Fast Fourier Transform algorithm to the samples of the voltage waveforms generated in Matlab. In this paper 256 points are considered for the DFT, and the corresponding complex values are used as inputs of the MLMVN.

Table 5.2 summarizes the results obtained using 50 neurons in the hidden layer of the MLMVN. As said before, the output layer contains 5 binary neurons, one for each electrical disturbance, and the nominal conditions correspond to a combination of five zeros.

Table 5.2 Precision for the MLMVN

Fault Class	Training CR%	Validation CR%
0-Normal	100	100
1-Sag	100	100
2-Swell	100	100
3-Harmonics	100	100
4-Notch	100	99.19
5-Interruption	100	100

Experimental Results

The first voltage waveform used to validate the performance of the MLMVN based classifier is shown in Figure 5.4, where the voltage sags and nominal conditions alternate with different durations. The time window proposed in Figure 5.4 has a duration of five seconds and, therefore, it contains 250 sinusoidal periods each of which is made up of 8000 samples. One of the most important aspects in the evaluation of classification results is the time interval taken into consideration. For example, the signal shown in Figure 5.4 can be processed using five consecutive time intervals of one second duration. The proposed monitoring method assesses the DFT for each interval and classifies them. As shown in Figure 5.4, this procedure allows the perfect classification of the considered voltage waveform.However, there are some situations in which the MLMVN misclassifies, for example in the case of a brief perturbation. Figure 5.5 describes this condition: the time interval taken into consideration is classified as normal, but it presents a voltage sag of 60 ms. To overcome this limitation, it is possible to reduce the duration of the time interval used for the classification procedure. In this way, short disturbances are identified with a high classification accuracy and the exact moment they start is detected. Figure 5.6 shows a classification example in which a voltage waveform is processed using 60 ms time intervals. Note that the excellent classification results shown in Figure 5.6 can also be obtained by considering the other fault classes.

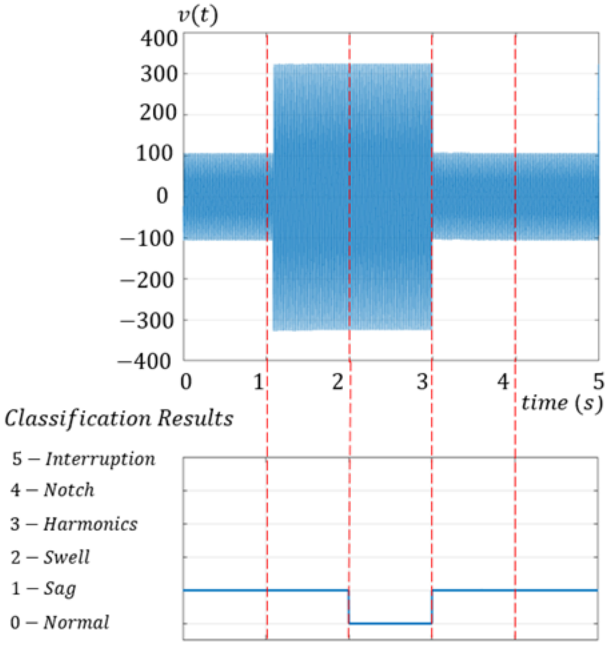


Figure 5.4: Voltage Waveform and classification

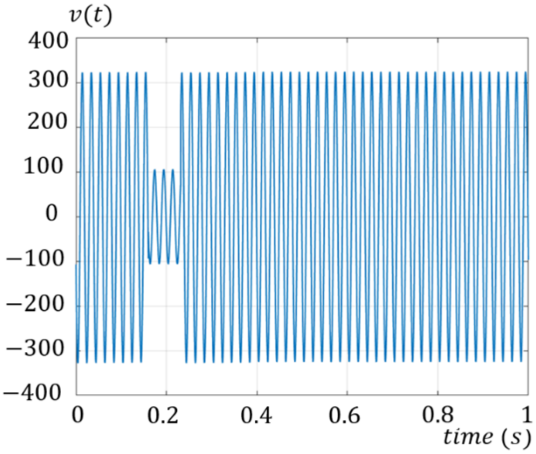


Figure 5.5: Voltage Waveform and classification

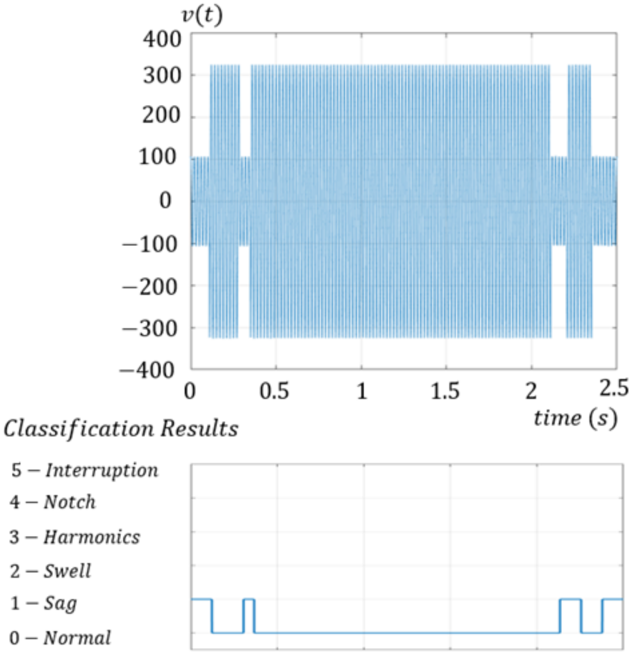


Figure 5.6: Voltage Waveform and classification

Table 5.3 summarizes the classification performances obtained using different time interval durations.

Table 5.3 Precision for the MLMVN

Disturbance	Time Interval			
	0.06s	0.6s	1s	2s
1-Sag	98.5%	90%	80%	66.6%
2-Swell	97%	87.5%	79.5%	66.6%
3-Harmonics	99.25%	90%	84%	75%
4-Notch	97%	90%	80%	68%
5-Interruption	98.5%	90%	80%	70%

These results were obtained considering real voltage measurements of 25 seconds and confirmed the excellent performance when the waveform is processed using a short time interval. On the other hand, the classification rate decreases as the number of periods processed simultaneously increases. It should be noted that the classification of the harmonic disturbance is slightly better than that of the other perturbations when using time intervals of 1 s and 2 s. The reason for this result is that the presence of a voltage component with frequency higher than 50 Hz introduces significant variation in the Fourier analysis. As shown in Figure 5.7, in the Magnitude representation several lines are introduced with respect to the normal condition, each of which corresponds to a frequency component. These contributions are also present in the case of a short duration harmonic perturbation and therefore make the classification slightly easier. As for the other voltage disturbances, they focus on the 50 Hz component, and this makes it difficult to recognize brief problems.

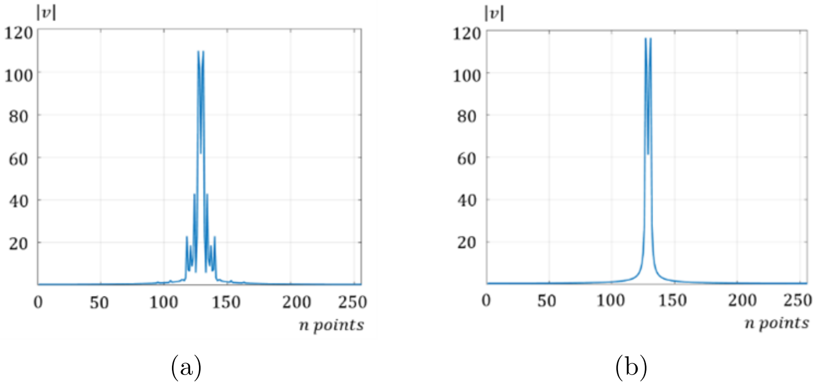


Figure 5.7: DFT results: a) magnitude in the case of a voltage waveform with harmonic disturbance of 0.15 s; b) magnitude of a normal voltage waveform.

In addition, it should be noted that some of the errors in detecting voltage sags and interruptions using a short time interval (0.006 s) correspond to class 4 misclassifications. In fact, in the instant in which the voltage drop begins, features very similar to those of a notch can occur. This means that the MLMVN can detect the starting point of these disturbances, but sometimes classifies it as a notch. Without considering these errors the classification rate would be over 99%.

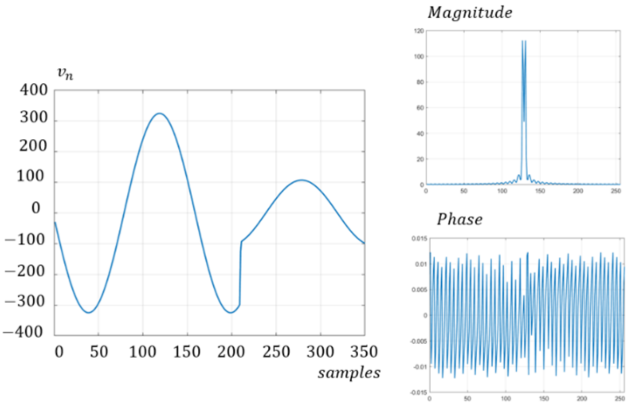


Figure 5.8: Voltage Waveform and classification

5.5.2 Convolutional Neural Network

The same voltage waveform described in the previous section was used to validate the performance of the CNN classifier, where the voltage sags and nominal conditions alternate with different durations using the voltage disturbance generator. For the validation of the CNN, the signal is converted to its time frequency domain using the STFT and then classified as a 500 × 5 image. The dimensions of the time frequency matrix represent 250 frequency components of 3 cycles, or 60 ms, with a sliding window of half a cycle which makes 5 DFT. The validation of the CNN STFT resulted in 4 misclassifications out of 479 classifications which makes it a 99.16% accuracy. Misclassifications in this experiment occurred in transitions between normal to sag or sag to normal. This is due to high-frequency components found at each transition, which led to a harmonic classification. Fig. 5.9 to 5.11 shows the classification results. The top plot shows the voltage signal to be transformed and classified. All figures show at least one sag that transitions to a normal or vice versa. As shown in the STFTs the transitions create high frequency components which often lead to a harmonic classification as shown in Figure 5.9 to 5.11.

Table 5.4 Precision for the CNN

Fault Class	Training CR%	Validation CR%
0-Normal	99.3	98.9
1-Sag	100	100
2-Swell	100	100
3-Harmonics	100	100
4-Notch	100	100
5-Interruption	100	100

Table 5.5 Recall for the CNN

Fault Class	Training CR%	Validation CR%
0-Normal	100	100
1-Sag	99.6	99.7
2-Swell	99.7	99.2
3-Harmonics	100	100
4-Notch	100	100
5-Interruption	100	100

5.6 Result Comparison

Finally, a comparison between CNN-STFT and MLMVN based classifiers is presented. For this comparison, a voltage signal containing all 5 categories of disturbances is generated. The rated voltage value is that of the Italian distribution network (Vrms = 230V, f = 50Hz). This signal is generated as shown in section IV and sampled with a frequency of 8kHz. The goal of the classification is to determine the power quality by studying 60 ms (3 cycles) at a time. Therefore, the sampled signal is divided into groups of 480 samples and each of them is assigned the corresponding classification. Figure 5.12a shows the overall signal and the correct classification of the 16 groups of the analyzed samples, while Figure 5.12b and Figure 5.12c present the classification results obtained through the two techniques. The MLMVN-based classifier misclassifies in the first sample. This is a very complex situation to recognize because the voltage sag situation occurs in the last half-period of the three taken into consideration. It can be said that

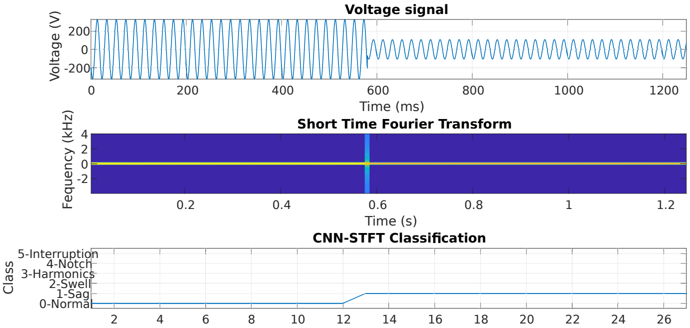


Figure 5.9: Experimental signal generated (Top). The STFT of the signal (Middle) and the classification results (Bottom).

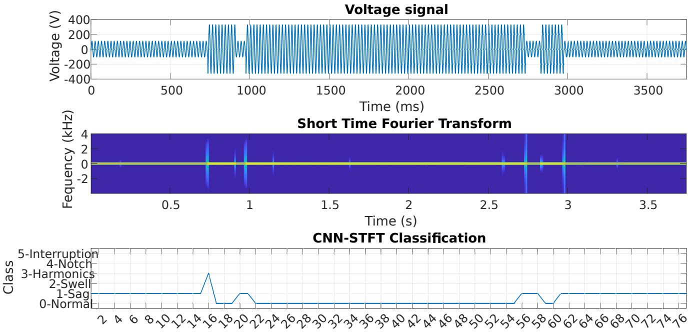


Figure 5.10: Experimental signal generated (Top). The STFT of the signal (Middle) and the classification results(Bottom).

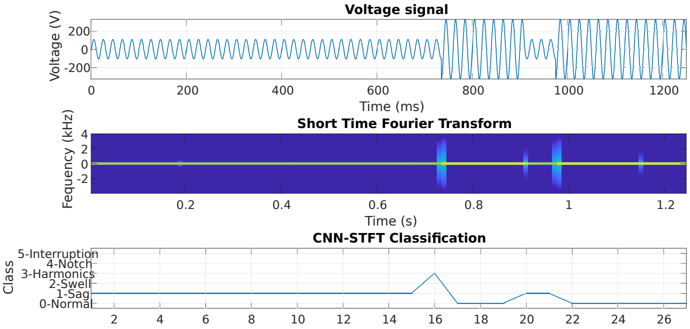
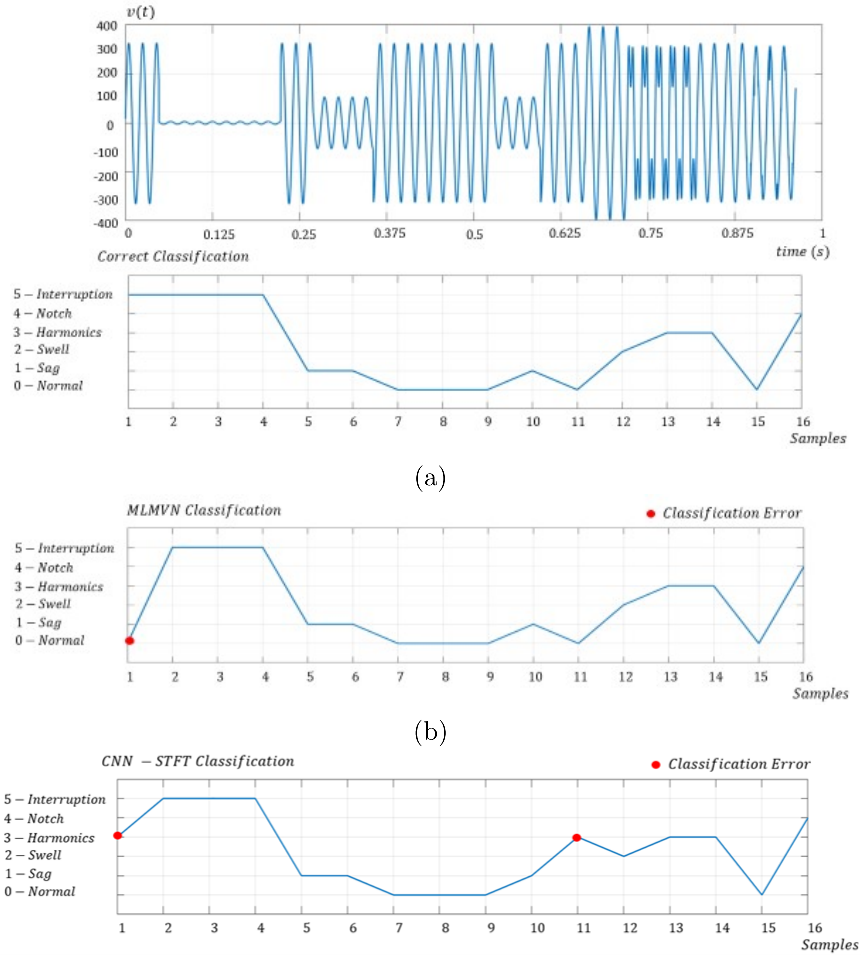


Figure 5.11: Experimental signal generated (Top). The STFT of the signal (Middle) and the classification results (Bottom).

by using one FT for each sine cycle, i.e. by analyzing one cycle at a time, this type of error can be eliminated by ensuring 100% accuracy.



(a)

(b)

(c)

Figure 5.12: DFT results: a) magnitude in the case of a voltage waveform with harmonic disturbance of 0.15 s; b) magnitude of a normal voltage waveform.

5.7 Conclusion

In conclusion, it can be stated that the two proposed techniques allow the monitoring of the power quality in a low voltage distribution network with an excellent level of accuracy. The low training time and the use of a common techniques, such as the Fourier transform, in the data processing phase make the two classifiers very versatile and easily adaptable for the recognition of other electrical disturbances. Compared to other techniques they allow the analysis and classification of a voltage signal in time and frequency. This can further enhance the feature extraction capabilities due to the addition of the frequency dimension. Furthermore, these are well known signal processing techniques applicable for a real-time environment. Future developments could be focused on improving performance when processing a larger number of cycles per classification and introducing additional types of disturbances that are very frequent in industry. Furthermore, the real-time applications of these two approaches will certainly be studied in the future to develop an effective monitoring tool for electric grids. In order to adapt the proposed classifiers to different acquisition devices in many other electrical systems, a measurement noise treatment will be introduced during the training phase. Finally, a very interesting future development will be the improvement of neural algorithms to work under multiple failure hypotheses to classify disturbances consisting of multiple distortions.

Chapter 6

Single Shot Power Quality Disturbance Detector

This chapter presents an innovative Single Shot Power Quality Disturbance Detector (SSPQDD). This algorithm presented in this chapter is an algorithm inspired in object detection algorithms. This algorithm is able to detect a disturbance a its duration along with a classification. Section 1 presents an introduction. Section 2 shows the mathematical formulation of the innovative algorithm showing the block diagram and the training data. Section 3 presents the training dataset and how it is supposed to be augmented. Section 4 presents the formal algorithm representation and how it should be implemented. It also presents the base network and how it should be modified in order to comply to the given needs of PQD classification. Section 5 briefly describes the training procedure and how it is implemented. Finally, Section 6 presents the conclusion of the chapter. [1]

[1] The development of the algorithm presented in this chapter has been published in "An Innovative Single Shot Power Quality Disturbance Detector Algorithm" in *in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-10, 2022, Art no. 2517210*, [93]

6.1 Introduction

The proposed SSPQDD is based on a pretrained VGG16 architecture. The VGG16 is a a deep CNN proposed by K. Simonyan and A. Zisserman from the University of Oxford in 2014. In the last years, VGG16 has been used for Image Recognition or Classification and for Image Detection and Localization, while it is not been used for PQD classification. In this work, the base architecture of the proposed SSPQDD is a VGG16 network because of its outstanding accuracy in classifying PQDs compared to other architectures presented in this work. The results of this comparison are shown in the following section. After the training of the base network, the fully connected layers are substituted by feature extraction layers. This is done in order to take advantage of the maxpooling layers typical of the VGG-16 architecture and to classify voltage signals in a grid like structure. The proposed approach classifies 16 samples of the voltage signal per grid. Figure 6.2 shows the block diagram of the proposed method in which the base network is the VGG16 architecture, and the feedforward layers are substituted by feature extraction layers for the grid like classification. The added stage uses a maxpooling layer for dimensionality reduction or, in other words, for classification of a larger number of samples. The final layer has 2 outputs. The first output, the confidence output, is a binary classification that classifies the signal as disturbance or no disturbance. The confidence output is used to mask the second output which is a multiclass classification.

St has 188 grids which are 16 samples per grid. This is fundamental to ensure that the algorithm will be able to find and classify Transient disturbances of few nanoseconds as well as Temporary disturbances lasting over 1 min. The confidence output is characterized by 2 outputs per grid, while the classification one has 6 output per grid due to the 6 types of disturbances taken into consideration. These include Sag, Swell, Harmonic, Transient, Notch, and Interruption. A confidence of class 0 means Normal.

6.2 Mathematical Formulation

The dataset used to train and validate the proposed network involves voltage signals containing Sag, Swell, Harmonics, Transient, Notch, and Interruption. All the PQDs have been generated randomly within the signal to ensure different time durations of the disturbances and to cover different locations in a window frame. The dataset was generated using Matlab Simulink considering a sampling frequency of 8 kHz for a total duration of 3.75 s. The target includes the confidence matrices and the classification matrices. The classification matrices are multiplied by a mask matrix for training stability purposes. The matrices are 2 x 188 for the confidence matrices and 6 x 188 for the classification matrices. The output is then confronted with the confidence and classification matrices. Figure 6.2 (a) shows an extract of the voltage signal used to train the algorithm including a swell disturbance instantaneously followed by a sag disturbance. Below the voltage signal there are the training matrices shown in Figure 6.2 (b) representing the classification matrices and the confidence matrices. Pretraining of the VGG16 is done classifying the disturbance as a multiclass classification problem using the softmax function. The fully connected layers are then removed, and the feature extraction layers and the output layers are then added to the base network. Training of the SSPQDD network is done with voltage signals and comparing the outputs with the target matrices shown in Figure 6.2 (b) The softmax function is used in a binary manner on each grid in order to determine if a PQD exists in that given grid. Thus, the probability to classify the jth class given a sample vector x and a weighting vector w is given by Equation 6.1.

$$P(\hat{y} = j|x) = \frac{e^{x^T w_j}}{\sum_1^K e^{x^T w_k}} \tag{6.1}$$

The multiclass softmax function is used in the classification output. The classification output, after the sofmax, is then multiplied by a mask matrix for training stability purposes and class imbalances. The Loss function is then calculated using the crossentropy loss. The crossentropy loss and the binary crossentropy loss are shown in Equation 6.2 and 6.3 respectively.

$$CE_{class} = \sum_{t=1}^{N} y_{class_t} log\big[St_{class}(x)_t\big] \tag{6.2}$$

$$CE_{conf} = \sum_{t=1}^{N} \big[y_{conf_t} log\big[St_{conf}(x)_t + (1 - y_{conf_t}) log\big[1 - St_{conf}(x)_t\big]\big] \tag{6.3}$$

The loss of the confidence and of the classifications are then added and the weigths of the proposed network are updated using the Adaptive moment optimization algorithm. The complete algorithm of the proposed SSPQDD is shown in Algoritm 1. After the algorithm is tested, the confidence matrices are then multiplied to the classification matrices in order to find the disturbance in the signal frame. Equation 6.4 to Equation 6.8 shows the formal mathematical representation of the procedure. Equation 6.4 and Equation 6.5 represent the output for the classification and the confidence respectively where they are convolution operations.

$$\begin{aligned} Class\colon l^2(\mathbb{Z}_T) &\to l^2(\mathbb{Z}_{NG}) \\ (x,\theta) \mapsto& Class\{(x,\theta)\} \triangleq x * \theta = F \end{aligned} \tag{6.4}$$

$$\begin{aligned} Conf\colon l^2(\mathbb{Z}_T) &\to l^2(\mathbb{Z}_{NG}) \\ (x,w) \mapsto& Conf\{(x,w)\} \triangleq x * w = Y \end{aligned} \tag{6.5}$$

Where N is the number of filters, G is the number of grids and, θ and w are the kernels convolved with the input x.

The softmax function is applied to the output in order to obtain a probability distribution on each grid for the class. The softmax function is defined in Equation 6.6 for the Class output and in Equation 6.7 for the confidence output.

$$\begin{aligned} H\colon l^2(\mathbb{Z}_N \times \mathbb{Z}_G) &\to l^2(\mathbb{Z}_N \times \mathbb{Z}_G) \\ Class(n,g) &\mapsto H\{Class(n,g)\} = h \end{aligned} \tag{6.6}$$

Where $h[n,g] = \Big\{h[n,g] \in \mathbb{R}; n \in \mathbb{Z}_N; g \in \mathbb{Z}_G\colon h[n,g] = \frac{e^{F_n}}{\sum_{i=1}^{K} e^{F_i}} \forall g\Big\}$.

$$\begin{aligned} H\colon l^2(\mathbb{Z}_N \times \mathbb{Z}_G) &\to l^2(\mathbb{Z}_N \times \mathbb{Z}_G) \\ Conf(n,g) &\mapsto H\{Conf(n,g)\} = St_{\eta_{conf}} \end{aligned} \tag{6.7}$$

Where $St_{\eta_{conf}}[n,g] = \Big\{St_{\eta_{conf}}[n,g] \in \mathbb{R}; n \in \mathbb{Z}_2; g \in \mathbb{Z}_G\colon St_{\eta_{conf}}[n,g] = \frac{e^{Y_n}}{\sum_{i=1}^{2} e^{Y_i}} \forall g\Big\}$.

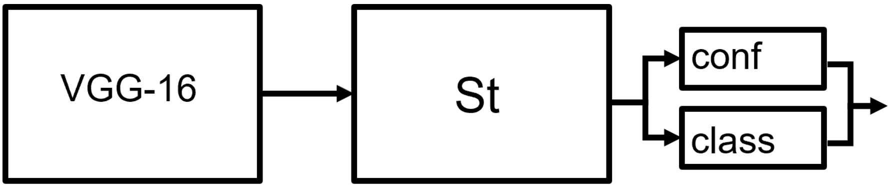
For the use of this algorithm, it is required to multiply the each element of the 1^{st} row of the confidence output with the each column of the class output. To do this a Hadamart product is defined in Equation 6.8 resulting in Equation 6.9.

$$\begin{aligned} \circ \colon l^2(\mathbb{Z}_N \times \mathbb{Z}_G) \times l^2(\mathbb{Z}_1 \times \mathbb{Z}_G) &\to l^2(\mathbb{Z}_N \times \mathbb{Z}_G) \\ (h, St_{\eta_{conf}}) &\mapsto \circ\{(h, St_{\eta_{conf}})\} \triangleq h \circ St_{\eta_{conf}} = St_{\eta_{class}} \end{aligned} \tag{6.8}$$

$$\text{where } St_{\eta_{class}} = \begin{bmatrix} h[1,1] * St_{\eta_{conf}}[1,1] & \dots & h[1,G] * St_{\eta_{conf}}[1,G] \\ \vdots & \ddots & \vdots \\ h[N,1] * St_{\eta_{conf}}[1,1] & \dots & h[N,G] * St_{\eta_{conf}}[1,G] \end{bmatrix}$$

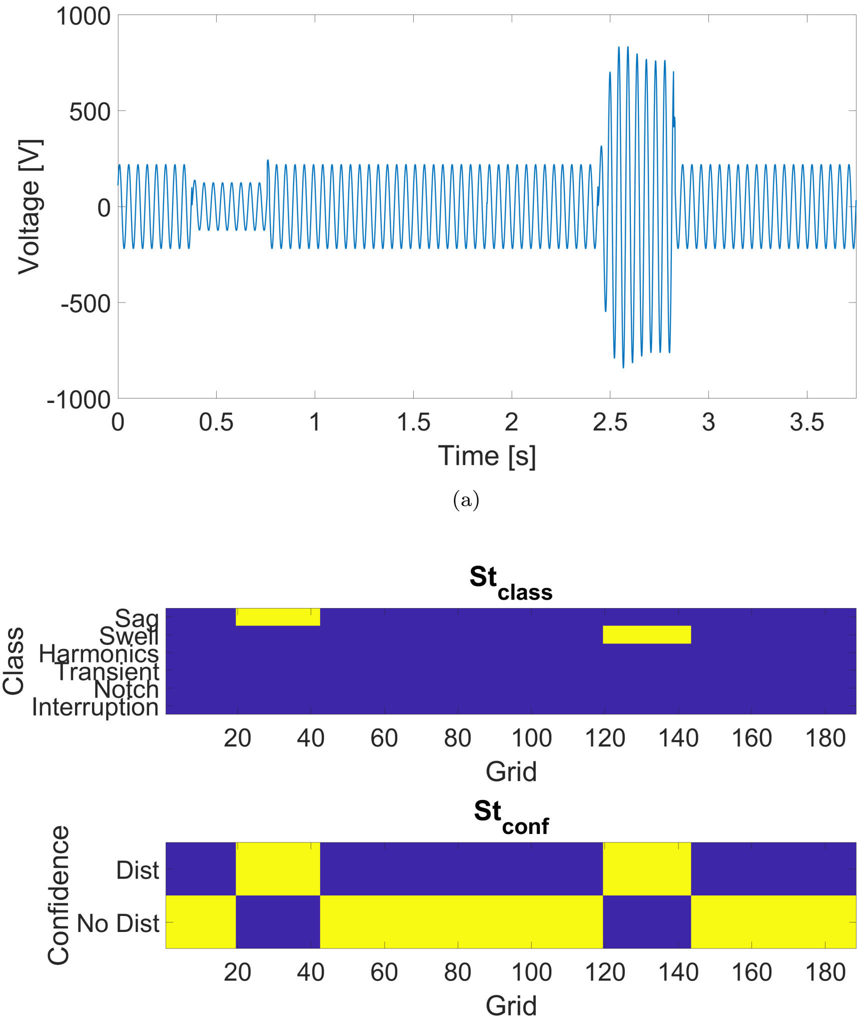
$$\begin{aligned} St_{\eta_{class}} \colon \mathbb{Z}_N \times \mathbb{Z}_G &\to \mathbb{Z}_{NG} \\ (n, g) &\mapsto St_{\eta_{class}}(n, g) \end{aligned} \tag{6.9}$$

An example of PQD classification performed using the proposed SSPQDD architecture is illustrated in Figure 6.2, where the output of the network in case of a transient disturbance is illustrated. The top subplot shows how, using the proposed architecture, even a small sliding window is able to identify the transient.



(a)

Figure 6.1: SSPQDD Block Diagram



(a)

(b)

Figure 6.2: SSPQDD Block Diagram

6.3 Training Dataset

The section describes the generation of the dataset using the simulink model shown in Section 3.2. The disturbances generated are then augmented in order to obtain a generalized dataset to obtain the best possible model.

6.3.1 Dataset Generation

The dataset is obtained as shown in Section 3.2. The dataset with all disturbances is generated with the localization of the disturbance inside a 3000 sample time frame. With the localization of the disturbances of the voltage time frame, the confidence training set and the class training set are generated. The confidence training set is a 2 × G matrix where the matrix is $\begin{bmatrix}0 & 1\end{bmatrix}^T$ and where the position of the disturbance on the voltage signal is $\begin{bmatrix}1 & 0\end{bmatrix}^T$. For the class training set matrices the dimensions are N × G. Where the classification, if there is any in that given grid, is $\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T$. The position of the 1 is the classification result for that given grid. In the 1^{st} position classifies as a sag, 2^{nd} position classifies as a swell, 3^{rd} position classifies as a harmonics, 4^{th} position classifies as a transient, 5^{th} position classifies as a notch and 6^{th} position classifies as an interruption.

6.3.2 Data Augmentation

The dataset generated was shifted in order to move the disturbances in different positions in the time frame. The 2 and 3 disturbances were also added and mixed in order to generalize the dataset and train the SSPQDD to be able to detect multiple disturbances in one time frame.

6.4 Algorithm

In this section, the SSPQDD algorithm is descried. Algorithm 1 shows the formal description of the SSPQDD algorithm. Since the algorithm uses the Adaptive moment optimization (Adam) algorithm, the parameters α, β_1 and β_2 are set to 0.00005, 0.5, and 0.999 respectively. N equals the number of classes which in our case it's 6 which include: sag, swell, harmonics, transient, notch and interruption. G represents the number of grids that the time signal is divided into which in this case, a voltage signal sampled at 8 kHz with 3000 samples, makes 188 grids or 16 samples per grid.

The SSPQDD algorithm requires an input signal x of 3000 samples, the batch size m for training which in this case was 256, the number of classes N and the number of grids G already mentioned and finally the Adam hyperparameters α, β_1 and β_2. Finally, a training mask M_{St} is required in order to stabilize training. The training mask is the the confidence dataset that filters the the classification output of the SSPQDD. This is required only for training purposes.

The SSPQDD samples m signals x from the dataset with its corresponding training sets y_{class} and y_{conf}. The training mask is then defined as $M_{St} = y_{conf}[1, G]$. While the SSPQDD weights has not converged, do a forward pass and obtain both outputs, *class* and *conf*. The softmax function is used on both outputs and the output of the *class* is then multiplied by the training mask finally obtaining St_{class} and St_{conf}. The loss function L as shown in Algorithm 1 is calculated and then the weights for the SSPQDD architecture are the updated.

Algorithm 1 SSPQDD, α =0.00005, β_1 =0.5, β_2 =0.999, $N = 6$, $G = 188$

```
Require: Input $x$ , target class $y_{class}$, target confidence $y_{conf}$, the training
    mask $M_{St}$, the batch size $m$, the number of classes $N$, the number of
    grids $G$, Adam hyperparameters $\alpha$, $\beta_1$, $\beta_2$.
while $\theta$ has not converged do
    Sample data batch $\{x^{(1)}, \cdots, x^{(m)}\}$ from dataset $x \sim P_{data}$
    Sample data batch $\{y^{(1)}_{class}, \cdots, y^{(m)}_{class}\}$ from $y_{class} \sim P_{class}$
    Sample data batch $\{y^{(1)}_{conf}, \cdots, y^{(m)}_{conf}\}$ from $y_{conf} \sim P_{conf}$
    for $i = 1, \cdots, m$ do
        $St_{class}(x^{(i)}) \leftarrow softmax(class(x^{(i)})) \circ M_{st}$
        $St_{conf}(x^{(i)}) \leftarrow softmax(conf(x^{(i)}))$
        $L^{(i)} = -\sum_{g=1}^{G}\Big[\sum_{t=1}^{N} y^{(i)}_{class_{(t,g)}} log\big[St_{class}(x^{(i)})_{(t,g)}\big]\Big] +$
    $\big[y^{(i)}_{conf_g} log\big[St_{conf_g}(x^{(i)})\big] + (1 - y^{(i)}_{conf_g}) log\big[1 - St_{conf_g}(x^{(i)})\big]$
    end for
    $\theta \leftarrow Adam(\bigtriangledown_\theta \frac{1}{m}\sum_{i=1}^{m} L^{(i)}, \theta, \alpha, \beta_1, \beta_2)$
end while
```

6.4.1 Architecture

The architecture used is a combination of the VGG16 architecture and feature extraction layers added with an 2 outputs as previously said in Section 6.2. The VGG16 deep learning architecture is shown in Figure 6.3. This architecture was modified in order to input a time signal. The kernel size for the convolutional layers are modified to $\begin{bmatrix}1 & 3\end{bmatrix}$ and the max pooling layers to $\begin{bmatrix}1 & 2\end{bmatrix}$. The Classification block as shown in Figure 6.3 is eliminated and the feature extraction layers are then added. The Feature extraction layers involve one more block that consist of a convolutional layer with 1024 filters a ReLu layer and then another convolutional layer with 1024 filters. The Output layers, confidence and class outputs, are added to the feature extraction. The confidence output is a convolutional layer with 2 filters and the class output is a convolutional layer with 6 outputs.

6.5 Training

The training was done in Matlab 2022a using Matlab's deep learning toolbox. Training took 36 hours with a dataset of 70000 signals. The loss after 36 hours was more or less 15. The SSPQDD first started correctly detecting the disturbances in the window frame and then started correctly classifying the disturbances.

6.6 Conclusion

An innovative Single Shot Power Quality Disturbance Detection algorithm was developed, trained, validated and tested. This new type of algorithm is based on object detection algorithms for detection and classification of objects in an image. The algorithm is mostly based on the YOLO and the SSD algorithms which uses the VGG16 algorithm as a base network and it is modified in order to comply to the use. The algorithm uses the VGG16 algorithm as a base network modified for time series signal. Feature extraction layers are added with 2 outputs, one for the confidence and the other for the class.

A dataset was generated with the simulink model shown in Section 3.2. This dataset contains 6 disturbances including the sag, swell, harmonics, transient, notch, and interruption. This dataset was augmented in order to

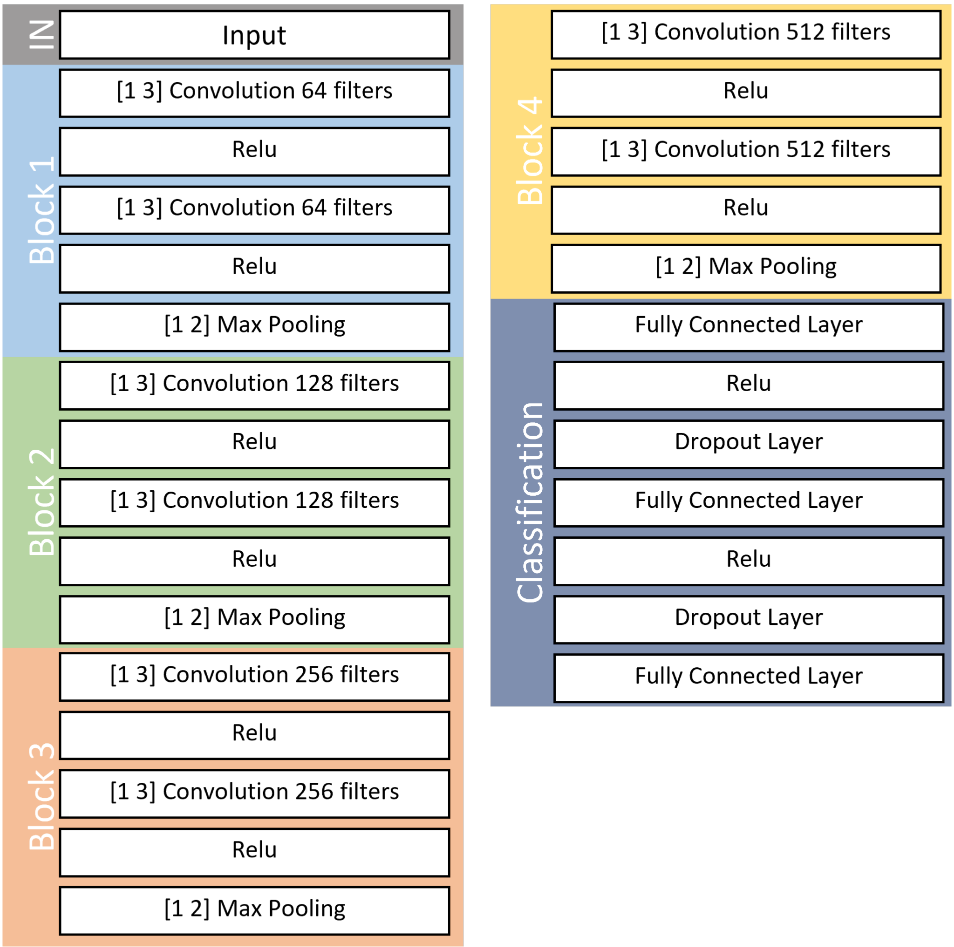


Figure 6.3: VGG-16 Architecture

obtain a generalized dataset. The data augmentation was done by shifting the disturbances in the window frame. different disturbances were added in the signal frame in order to train the algorithm to detect and classify more than one disturbance in a window frame.

Training took 36 hours in order to obtain good results. The results are shown in 7 and compared to state of the art deep learning architectures.

Chapter 7

Results and Discussion

In this chapter, the results and discussion of the performance of the SSPQDD is presented. The performance of the SSPQDD is measured by comparing it with state of the art algorithms. Section 1 gives an introduction of the state of the art algorithms used in this experiment and why were this algorithms were chosen. Section 2 compares these algorithms with the SSPQDD calculating the precision, recall, F1-score, and the Area under the curve (AUC). A visual comparison is also made comparing the computational effort of this algorithm and its accuracy. Another comparison was made with the sliding window method and the SSPQDD. Finally, Section 6 presents the conclusion of this chapter. [1]

7.1 Introduction

In this section the proposed SSPQDD was compared to well-known deep learning architectures that show good results in image classification tasks. In order to compare these architectures, an experiment was made training all architectures and comparing their performances. Due to the nature of classification of the grid-like structured in early stages, the SSPQDD can

[1] The results presented in this chapter has been published in "An Innovative Single Shot Power Quality Disturbance Detector Algorithm" in *in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-10, 2022, Art no. 2517210*, [93]

detect disturbances in voltage signals that have duration from few nanoseconds up to over 1 min. The dataset used for training the different deep learning architecture was the same used for the SSPQDD, as described in the previous section. The architectures used to compare the performances of the proposed SSPQDD are the following:

- DarkNet , which is a fast and simple deep learning-based object detection framework [94].
- AlexNet , which is a milestone in deep CNN and it is based on 8 layers (five convolutional layers and three fully-connected layers) [95].
- ResNet which is a significantly deep network implemented using layer skips [96].
- The basic VGG16 network [97].

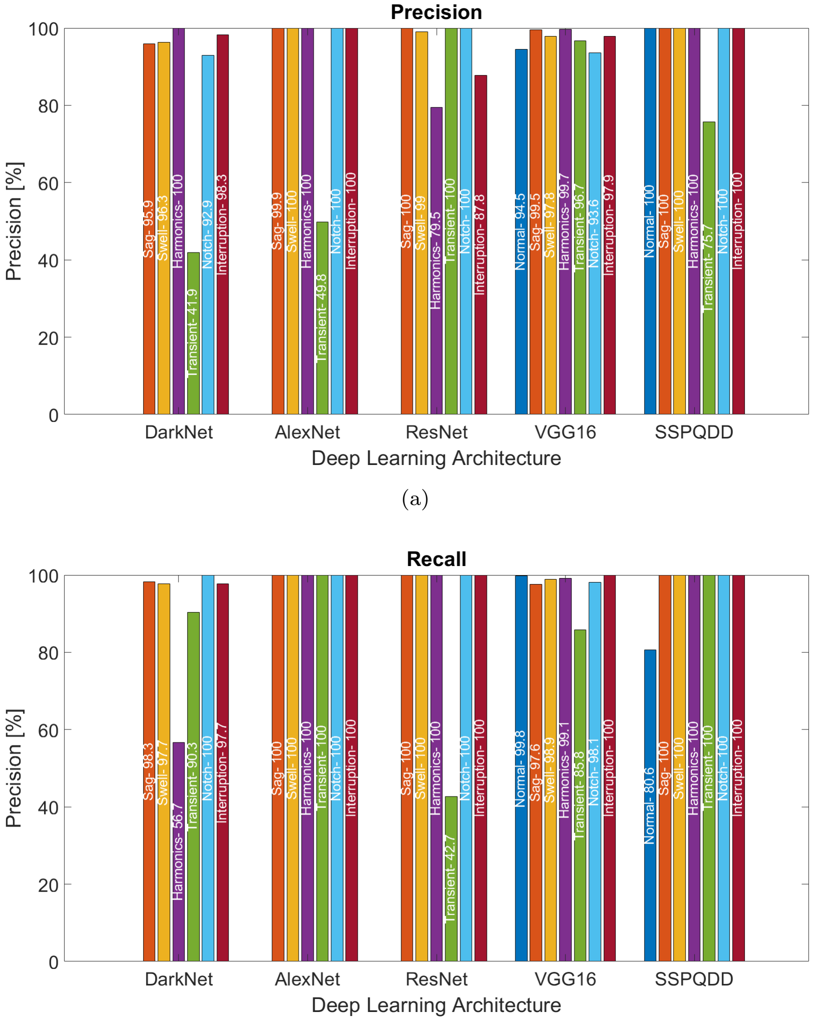
7.2 Comparison Results

The results of the comparison are shown in 7.2(a) and 7.2(b). More in detail, the first comparison has been carried out measuring the accuracy, the number of layers and the computational resources required for each architecture, as shown in 7.2(a). Similarly, 7.2(b) compares the accuracy of each architecture, taking into account also the number of networks parameters. Figure 7.3 shows the 3D visualization of the comparison. The highest accuracy was obtained using the basic VGG16. However, the latter network requires a lot of computational effort due the considerably great number of parameters, as it is possible to see analyzing Figure 7.2 (b). Quite the contrary, the proposed SSPQDD architecture does not require that much of memory space, and it is capable to achieve remarkably high accuracy despite a substantially lower number of parameters. Investigating more in detail the results obtained for every PQD under analysis, the different architectures show good results for each disturbance except for the transient and the normal condition. The VGG16 represents the exception since it is capable to found most of the transients even better than the proposed network. This is the reason why it has been chosen for the base network of the SSPQDD. The problem with the VGG16 is that it fails to identify where the disturbances are in the window frame.

Table 7.1 Comparison of different state of the art architectures

		Normal	Sag	Swell	Harm	Trans	Notch	Interrupt
Darknet	Precision	0	95.9	96.3	100	41.9	92.9	98.3
	Recall	0	98.3	97.7	56.7	90.3	100	97.7
	F1-Score	0	97.08	96.99	72.37	57.24	96.32	97.99
	AUC	0.5	1	0.9821	0.9412	0.941	0.9977	0.9985
Alexnet	Precision	0	99.9	100	100	49.8	100	100
	Recall	100	100	100	100	100	100	100
	F1-Score	0	99.95	100	100	66.48	100	100
	AUC	0.5	0.999	0.9993	1	0.9149	1	0.9995
Resnet	Precision	0	100	99	79.5	100	100	87.6
	Recall	0	100	100	100	42.7	100	100
	F1-Score	0	100	99.49	88.57	59.85	100	93.5
	AUC	0.5	1	0.995	0.8977	0.8866	1	0.9389
VGG-16	Precision	94.5	99.5	97.8	99.7	96.7	93.6	97.9
	Recall	99.8	97.6	98.9	99.1	85.8	98.1	100
	F1-Score	97.08	98.54	98.35	99.4	90.92	95.8	98.94
	AUC	0.994	0.9847	0.9938	1	0.9286	0.9836	0.998
SSPQDD	Precision	100	100	100	100	75.7	100	100
	Recall	80.6	100	100	100	100	100	100
	F1-Score	89.26	100	100	100	86.17	100	100
	AUC	1	0.9997	0.9996	0.9874	0.8678	0.9907	1

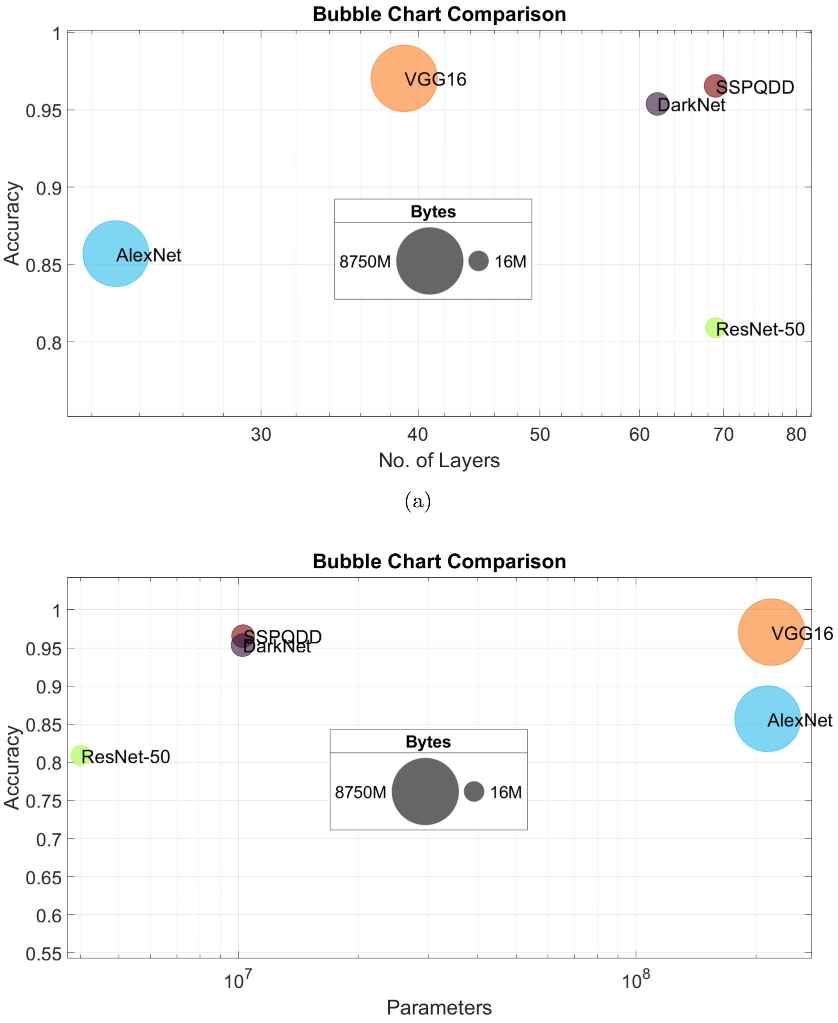
Table 7.1 shows the precision, recall, F1-score and the Area Under the ROC (Receiver operating characteristic) curve (also known as AUC) of the different architectures for each class of disturbance under consideration. For the normal class or no disturbance DarkNet and ResNet gave a 0% result due to their limited capabilities differentiating the normal from the transient. Similarly, also the AlexNet resulted in 0% Precision and 100% recall for the normal class. On the other hand, the VGG16 and the SSPQDD resulted in high precision, recall, F1-score (i.e., the harmonic mean between precision and recall) and AUC for the normal class. Other than the normal class, each architecture gave good results, even if the VGG16 and the proposed SSPQDD network provide the highest accuracy and the better results for each analyzed class. Overall, the VGG16 provide a total accuracy of 97.04% while the proposed network reaches the 96.55%. Results using training data also highlight how the proposed SSPQDD represent the only available approach able to detect multiple disturbances with different durations and different locations in a single window frame. A visual summary of this results are shown in Figure 7.1.



(a)

(b)

Figure 7.1: (a) Precision comparison of deep learning architectures (b) Recall comparison of deep learning architectures



(a)

(b)

Figure 7.2: (a) Accuracy vs No. Layers (b) Accuracy vs No. of parameters

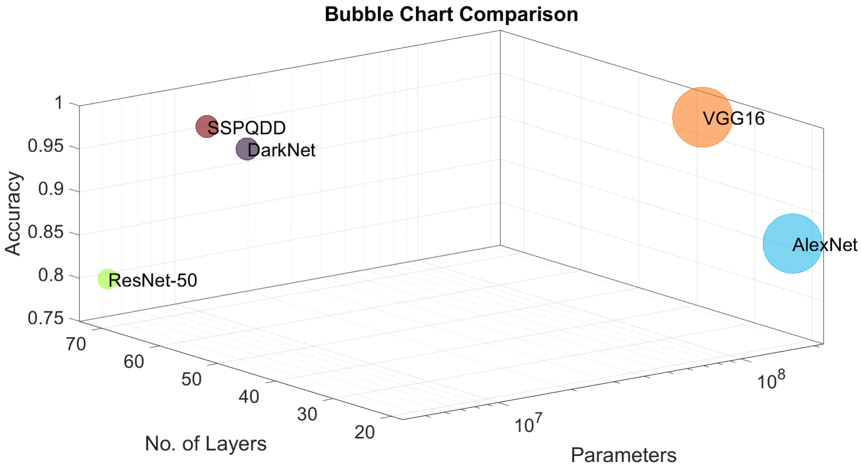


Figure 7.3: 3D bubblechart of the deep learning architectures comparison

Figure 7.4 to 7.13 illustrates the results obtained using the simulated dataset in case of two or more disturbances of different kinds and different durations in the same window frame. The random signal illustrated in Figure 7.4 includes an small notch. The SSPQDD is the only one able to detect and classify the disturbance, while DarkNet, resNet and VGG-16 can detect only classify the window frame. The Darknet and the Resnet missclassified in this example. In Figure 7.5 there is a small harmonic distortion followed by a small interruption and a swell. The SSPQDD detected all three disturbances and correctly classified them while the other deep learning architectures all detected the swell except for the darknet that detected the harmonic distortion. In Figure 7.6 the signal contains a small notch folled by a harmonic distortion of 750 miliseconds. Once again the SSPQDD was successful at detecting both disturbances while the other four architectures only detected the harmonic distortion. In Figure 7.7 the SSPQDD made two missclassifications by classifying a sag and a swell while in the time frame there is only a notch. All the other architecture correctly classified the notch. Figure 7.8 shows the SSPQDD detecting the small notch at the beginning of the time frame followed by the successful detection and classification of the interruption while the Darknet only detected the notch and the VGG16 detected the interruption. The other two architectures classified a transient. The next example shown in Figure 7.9, the disturbances on example shown in Figure 7.8 are switch. The SSPQDD detected and classified both while the other architectures classified only the interruption. Figure 7.10 shows a time fram with a harmonic distortion followed by a transient. The SSPQDD missclassified the harmonics and classified a notch and a sag but successfully detected the transient while the only architecture to classified the harmonics was the darknet while the others classified a swell. Figure 7.11 7.12 and 7.13 had one disturbance in each time frame being a swell, harmonic and interruption respectively. The SSPQDD detected and correctly classified all three. While the other architectures were not always successful. The proposed SSPQDD algorithm is the only one able to detect all the disturbance in a single window. The DarkNet identifies only the harmonics, while AlexNet, ResNet and VGG-16 identify only the swell. This means that all the state-of-the-art networks used for comparison completely miss to identify a long interruption of over 200 ms, which is not acceptable in almost every application. The analysis in 7.2(a) 7.2(a) and 7.3 summarizes entirely the major contributions brought by the proposed algorithm and the research gap filled by the SSPQDD. These figures highlight perfectly the most critical, powerful and important feature of the approach, ensuring also levels of accuracy on the single disturbance comparable with the most outstanding works available.

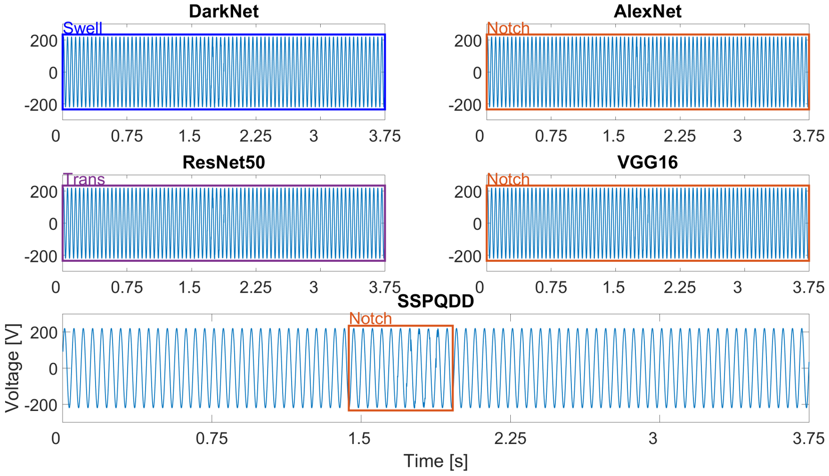


Figure 7.4: Result 1

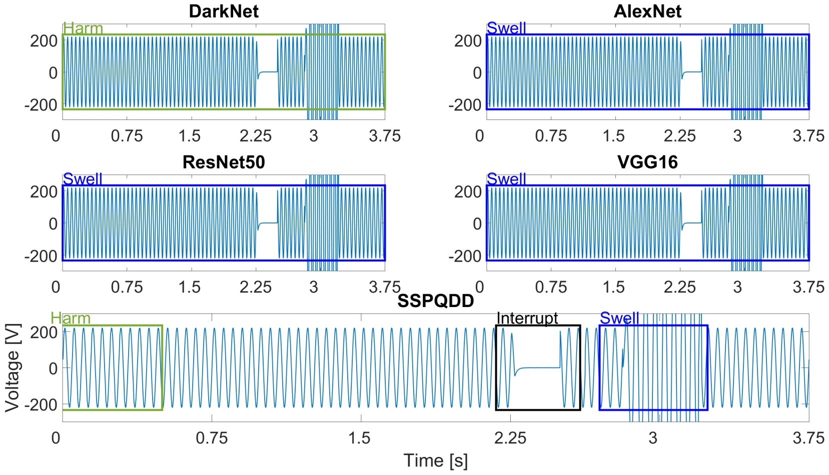


Figure 7.5: Result 2

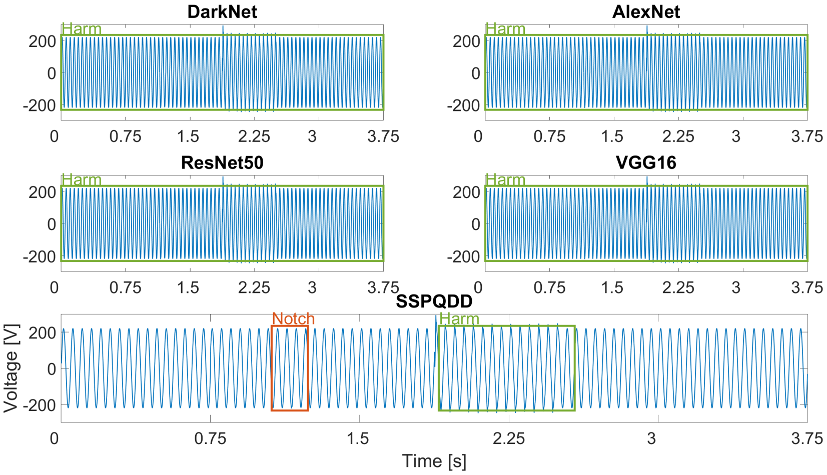


Figure 7.6: Result 3

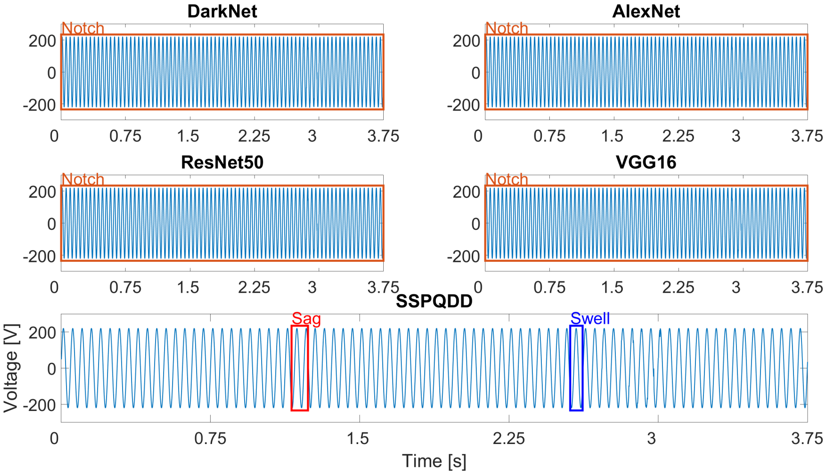


Figure 7.7: Result 4

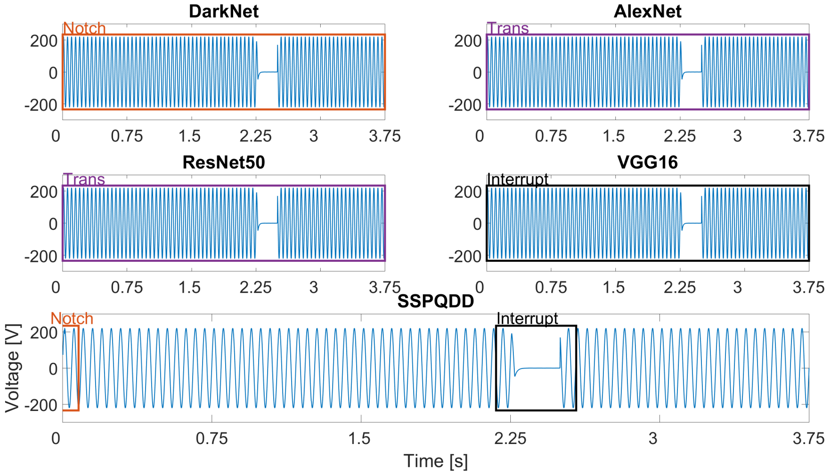


Figure 7.8: Result 5

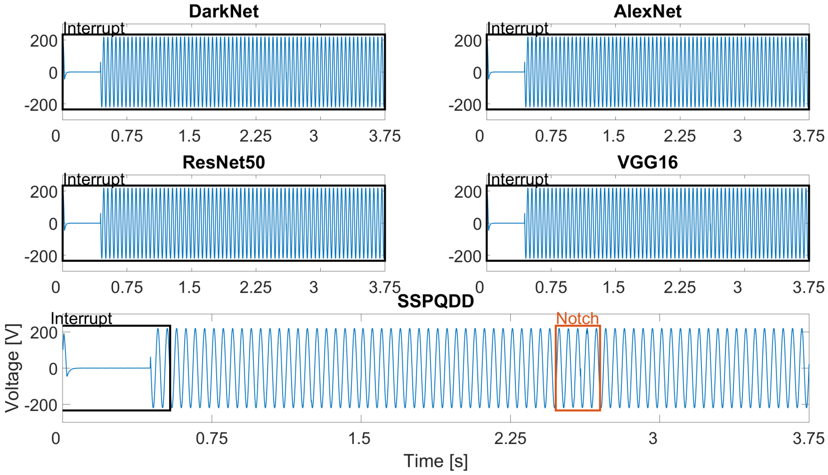


Figure 7.9: Result 6

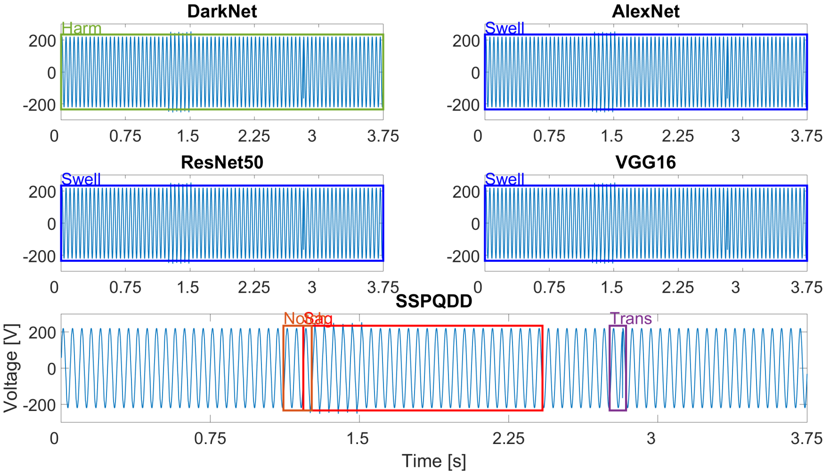


Figure 7.10: Result 7

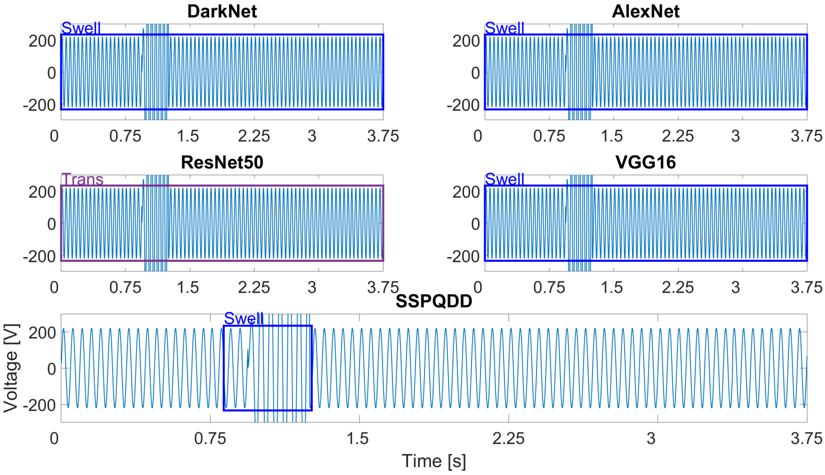


Figure 7.11: Result 8

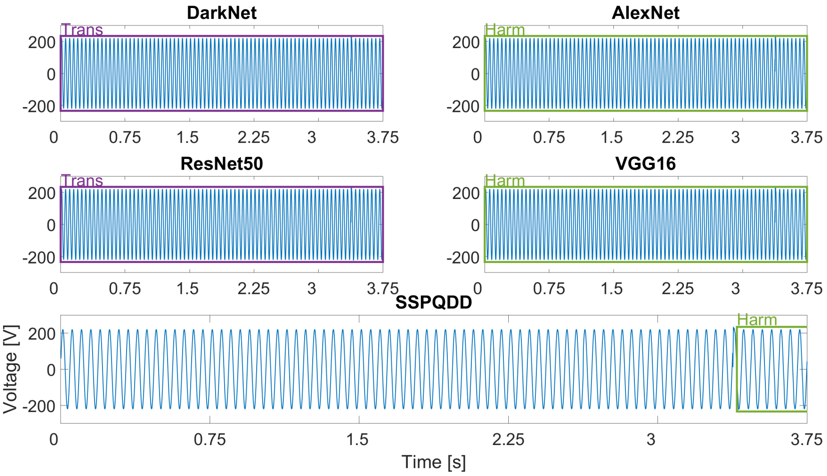


Figure 7.12: Result 9

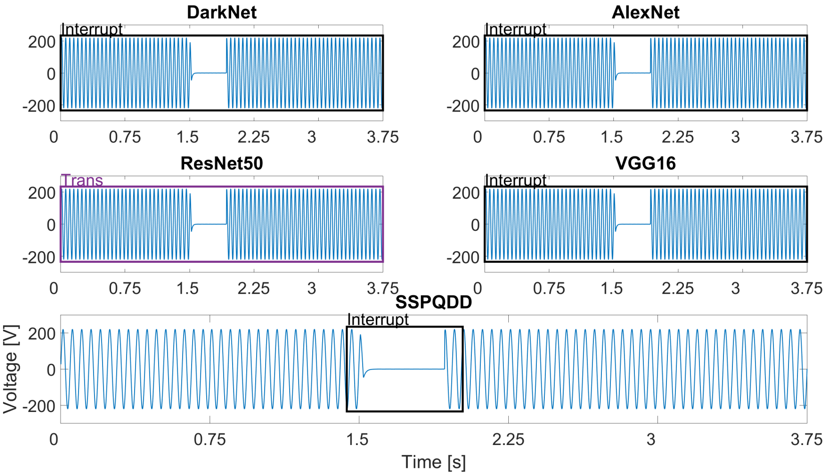


Figure 7.13: Result 10

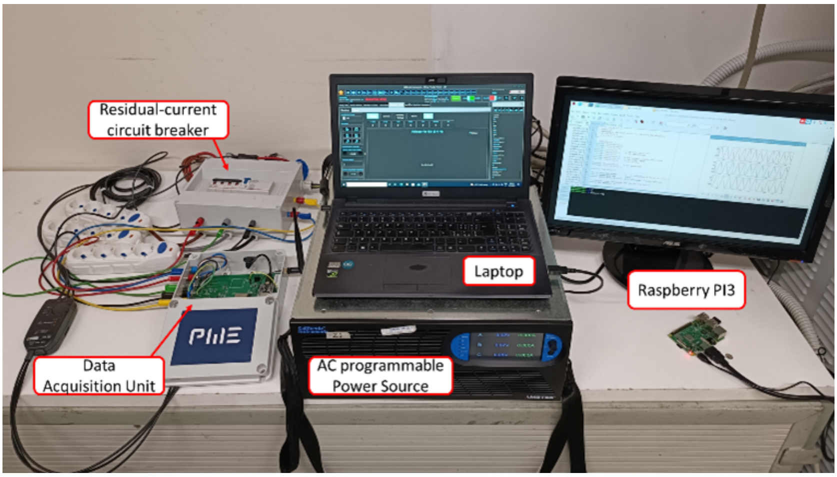

Residual-current circuit breaker
Laptop
Raspberry PI3
Data Acquisition Unit
AC programmable Power Source

Figure 7.14: Experimental Setup

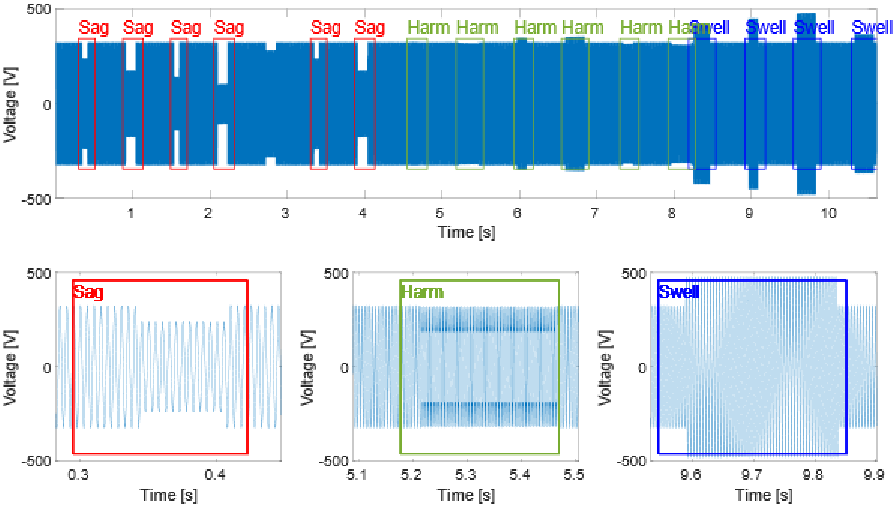


Figure 7.15: Experimental Setup

7.3 Conclusion

A simulated dataset of voltage signals containing different voltage disturbances has been generated using MATLAB Simulink as shown in Chapter 3 to train the proposed SSPQDD deep learning architecture. The simulated dataset are generated using 8kHz with 30000 samples per window frame. The dataset was augmented as explained in Chapter 6. The training of the SSPQDD has been carried out following the guidelines presented in the proposed Algorithm 1. To test the effectiveness and the contributions of the proposed approach, the results obtained using the SSPQDD have been compared with classical deep learning architectures which includes DarkNet, AlexNet, ResNet, and VGG16. The proposed SSPQDD has proven to be superior to the other approaches in almost all aspects (including accuracy, number of layers, computational complexity, and number of parameters). In fact, other than being superior in performances, the SSPQDD has proven to be efficient in terms of use of computational resources. This is due to the lack of feed forward network that usually requires most of the computational effort. With that been said, the SSPQDD can be an outstanding candidate for use in embedded electronics, where the amount of computational resources plays a significant role in the selection process of the DL architecture. Experimental results prove that the SSPQDD can effectively and efficiently detect and classify multiple voltage disturbances in a single window frame. These disturbances varied in duration and intensity, and the SSPQDD detected and classified each one of theme effectively. Quite the contrary, the other methods available in literature fail to classy more than one disturbance in a single window, missing to consider even long and major disturbances.

Chapter 8

Experimental Setup and Measurements

This chapter explains an implementation of the SSPQDD. The experimental setup and measurements are presented. The main goal of this chapter is to quantify the performance of an inductive passive PQ filter. The experiment is done using an AC programmable power source were PQDs are generated. The experiments are done using linear loads, linear inductive loads, and non linear loads. The data acquisition is done using the Dewesoft. Section 1 presents an brief introduction. Section 2 presents the system identification procedure followed by Section 3 which clearly shows the experimental setup for these tests. Section 4 shows the results with linear loads, inductive loads and non linear loads where the SSPQDD algorithm detects and classifies the disturbances and calculates its PQ parameters susch as V_{rms}*,*I_{rms}*,* THD_v*,* THD_i*, Real Power, Reactive Power, and Apparent Power. Finally, Section 5 presents the conclusion.*

8.1 Introduction

The purpose of this chapter is to show that the developed algorithm can be used as a tool to automatically detect and classify power quality disturbances. In this work, this algorithm is used to characterize a Power Quality Inductive Passive Filter (PQIPF). The goal is to measure the input and output of the filter in order to understand its behavior. The filter is connected to an AC programmable voltage disturbance generator that is programmed to generate different types of disturbances. The output is connected to linear and non linear loads. The goal of the algorithm is to detect and classify the disturbances in the input and in the output to determine the filtering capacity of the filter and evaluate it's performance.

In order to perform this experiment and to quantify it's performance, an experimental test bench must be created. First, the test bench must include an AC programmable voltage disturbance generator that is directly connected to the PQIPF which is then connected to the load. The voltage and currents are then measured in the input and at the output. The algorithm then detects and classify each disturbance and calculates the PQ parameters in each disturbance.

8.2 System Identification

The PQIPF is an inductive filter connected in a certain way that claims to reduce the Total Harmonic Distortions (THD) and a the same time the reactive power which corrects the power factor (pf). The experiment is divided in two parts. First, the Programmable AC disturbance generator will be connected to the filter which is then connected to a linear load. The scope of this experiment is to study the effects of the PQIPF on simple linear loads where it receives disturbances from the power source. The main purpose of this is to use the algorithm to detect and classify disturbances in voltages. The second part of the experiment is to connect motors and non linear loads to the filter. The scope of this experiment is to study the effects of the PQIPF on dynamic and non linear loads. Also to study the effects of the PQIPF when the disturbances are produced from the loads. In this part, the algorithm is used on voltages in order to detect and classify disturbances and to calculate the PQ parameters on the disturbances in order to study the effects of the filter. The experimental block diagram is shown in Figure 8.1.

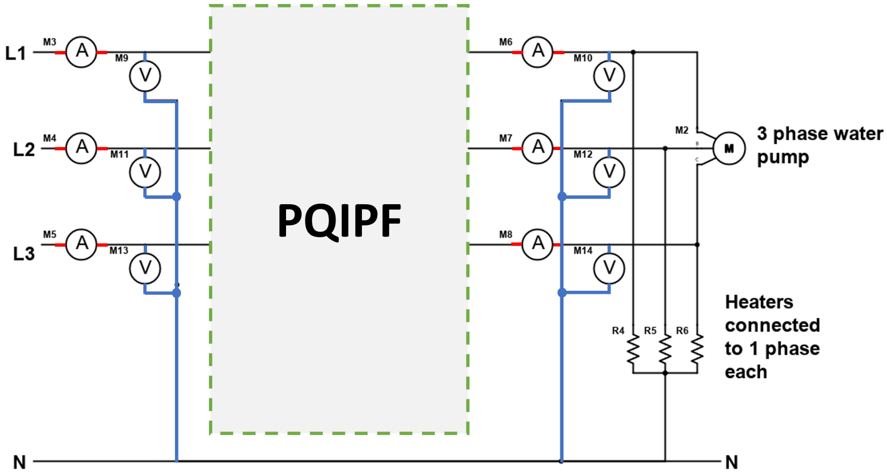


Figure 8.1: Experimental setup schematic

8.3 Experimental Setup

The experimental setup consist of the Amtek AC programmable power source by California Instruments as the Programmable AC disturbance generator. This is connected to the E-Wall from Energia Europa which is the Power Quality filter. And finally to the loads. The linear loads are three heaters which are basically resistive components. The non linear loads are composed of AC pumps that simulate a water reserve. The Input and Output measurements for each phase is measured using the Dewesoft measuring device. The system schematics is shown in Figure 8.2.

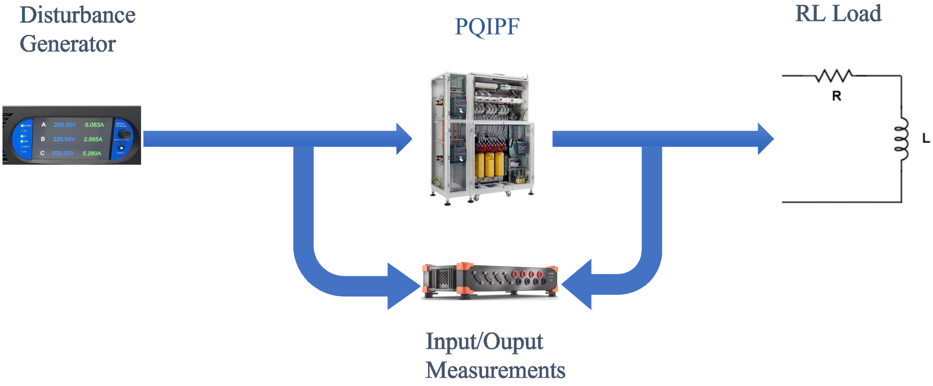


Figure 8.2: Experimental setup schematic

8.3.1 Disturbance Generator

The disturbance generator shown in Figure 8.3 is the Amtek AC programmable source by California Instruments as said in the previous section. This is a 6kW up to 400 V and 8A for each phase. The Ametek AC programmable power source can be programmed to generate disturbances using the software shown in Figure 8.4 and Figure 8.5. Figure 8.4 shows the dashboard to induce harmonic components into the signal and Figure 8.5 shows how the disturbances are generated in sequence. The programmable AC source is able to establish different disturbances with different time intervals. With this equipment the Sag, Swell, Harmonics, Transients, Notchs and Interruptions are programmed varying their amplituds and durations. For the harmonic distortions, different harmonic components are determined. All disturbances are programmed following the IEEE 1159 recommended practices standard.

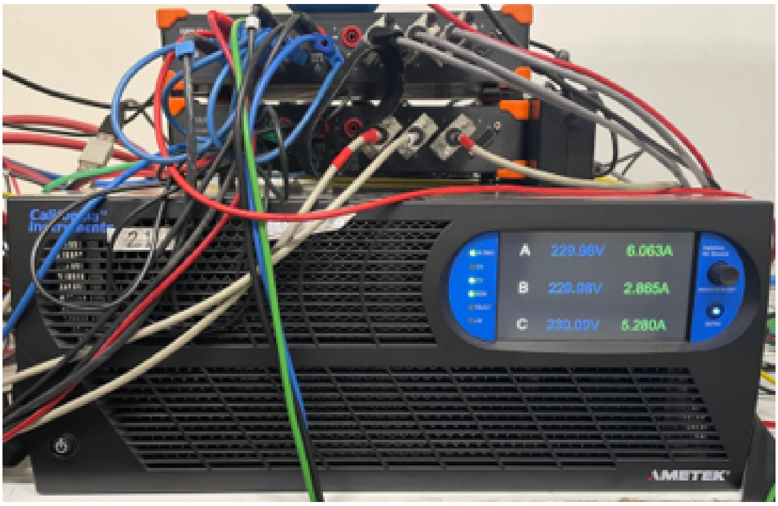
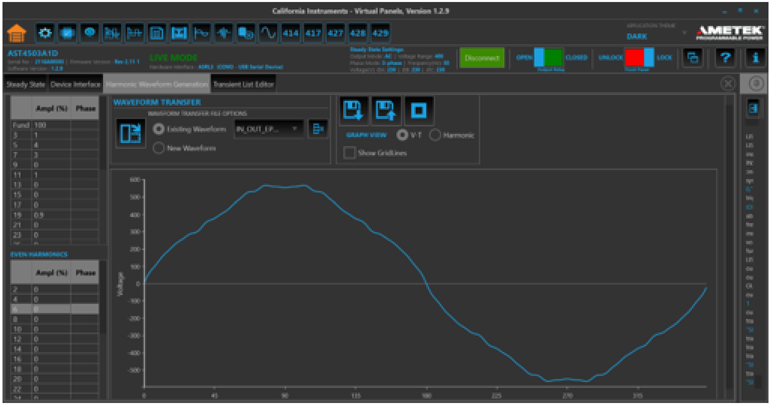


Figure 8.3: Ametek AC Programmable Source.


California Instruments - Virtual Panels, Version 1.2.9

AST4503A1D

LIVE MODE

Steady State | Device Interface | Harmonic Waveform Generation | Transient List Editor

WAVEFORM TRANSFER

WAVEFORM TRANSFER FILE OPTIONS

● Existing Waveform IN_OUT_EP...

○ New Waveform

GRAPH VIEW ● V-T ○ Harmonic

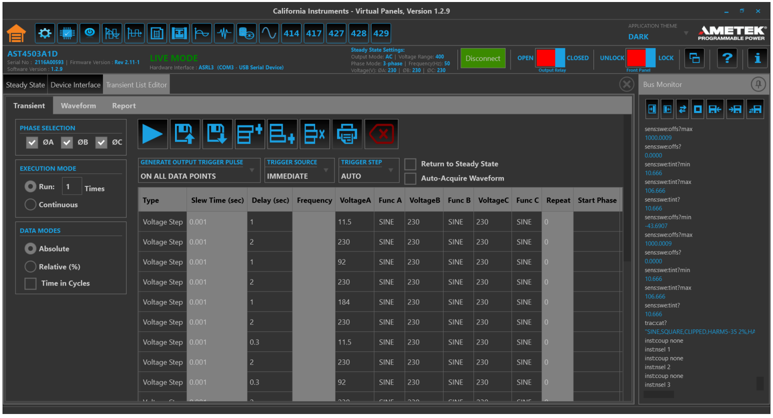
□ Show GridLines

	Ampl (%)	Phase
Fund	100	
3	1	
5	4	
7	3	
9	0	
11	1	
13	0	
15	0	
17	0	
19	0.5	
21	0	
23	0	

EVEN HARMONICS

	Ampl (%)	Phase
2	0	
4	0	
6	0	
8	0	
10	0	
12	0	
14	0	
16	0	
18	0	
20	0	
22	0	

Figure 8.4: Ametek AC Programmable Source Software.


California Instruments - Virtual Panels, Version 1.2.9

APPLICATION THEME DARK

AST4503A1D

Serial No : 2116A00593 | Firmware Version : Rev 2.11-1

Software Version : 1.2.9

LIVE MODE

Hardware Interface : ASRL3 (COM3 - USB Serial Device)

Steady State Settings:

Output Mode: AC | Voltage Range: 400

Phase Mode: 3-phase | Frequency(Hz): 50

Voltage(V): ØA: 230 | ØB: 230 | ØC: 230

Steady State | Device Interface | Transient List Editor

Transient | Waveform | Report

PHASE SELECTION

☑ ØA ☑ ØB ☑ ØC

EXECUTION MODE

● Run: 1 Times

○ Continuous

DATA MODES

● Absolute

○ Relative (%)

□ Time in Cycles

GENERATE OUTPUT TRIGGER PULSE: ON ALL DATA POINTS

TRIGGER SOURCE: IMMEDIATE

TRIGGER STEP: AUTO

□ Return to Steady State

□ Auto-Acquire Waveform

Type	Slew Time (sec)	Delay (sec)	Frequency	VoltageA	Func A	VoltageB	Func B	VoltageC	Func C	Repeat	Start Phase
Voltage Step	0.001	1		11.5	SINE	230	SINE	230	SINE	0	
Voltage Step	0.001	2		230	SINE	230	SINE	230	SINE	0	
Voltage Step	0.001	1		92	SINE	230	SINE	230	SINE	0	
Voltage Step	0.001	2		230	SINE	230	SINE	230	SINE	0	
Voltage Step	0.001	1		184	SINE	230	SINE	230	SINE	0	
Voltage Step	0.001	2		230	SINE	230	SINE	230	SINE	0	
Voltage Step	0.001	0.3		11.5	SINE	230	SINE	230	SINE	0	
Voltage Step	0.001	2		230	SINE	230	SINE	230	SINE	0	
Voltage Step	0.001	0.3		92	SINE	230	SINE	230	SINE	0	

Bus Monitor

```
sens:swe:offs?max
1000.0009
sens:swe:offs?
0.0000
sens:swe:tint?min
10.666
sens:swe:tint?max
106.666
sens:swe:tint?
10.666
sens:swe:offs?min
-43.6907
sens:swe:offs?max
1000.0009
sens:swe:offs?
0.0000
sens:swe:tint?min
10.666
sens:swe:tint?max
106.666
sens:swe:tint?
10.666
trac:cat?
"SINE,SQUARE,CLIPPED,HARM5-35 2%,HA
inst:coup none
inst:nsel 1
inst:coup none
inst:nsel 2
inst:coup none
inst:nsel 3
```

Figure 8.5: Ametek AC Programmable Source Software.

8.3.2 Data Acquisition

The data acquisition instrument used in this experiment is the Dewesoft. The Dewesfot Sirius XHS shown in Figure 8.6 is a multiple purpose measuring instrument capable of a sampling frequency up to 15Mhz. The Dewesoft is capable of measuring 4 voltages and 4 currents simultaneously. That means that this device is capable of measuring phase 1,2,3 and neutral. At the same time this device can measure currents of phase 1,2 and 3 using the Dewesoft FluxGate transducers as shown in Figure 8.7. The Dewesoft is stacked with a second one in order to measure the Input and Output simultaneously.

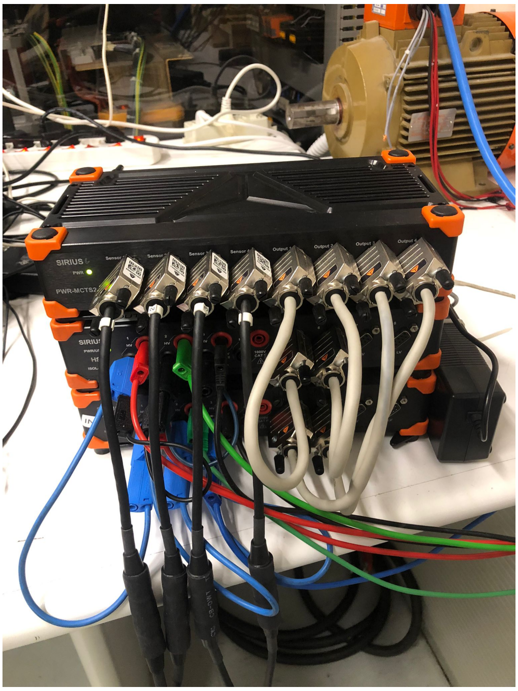


Figure 8.6: Dewesfot Sirius XHS

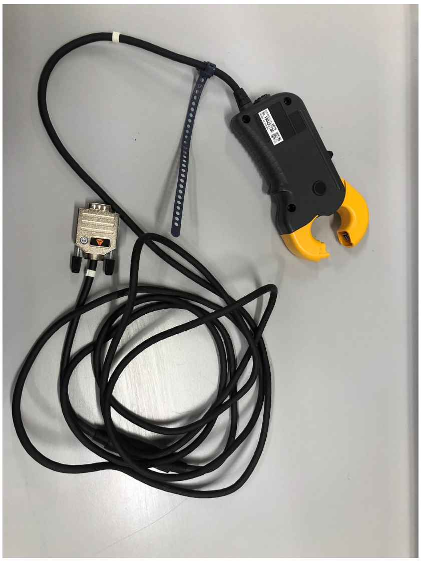


Figure 8.7: Dewesfot FLuxGate Transducers

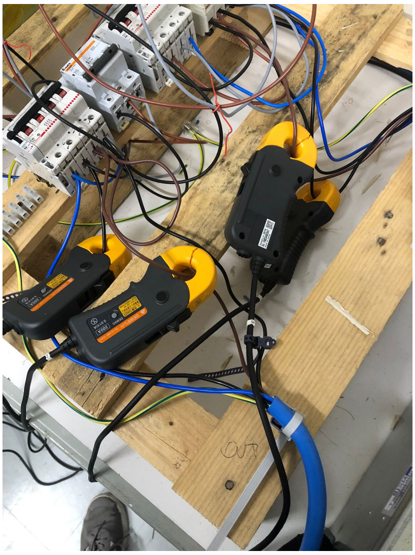
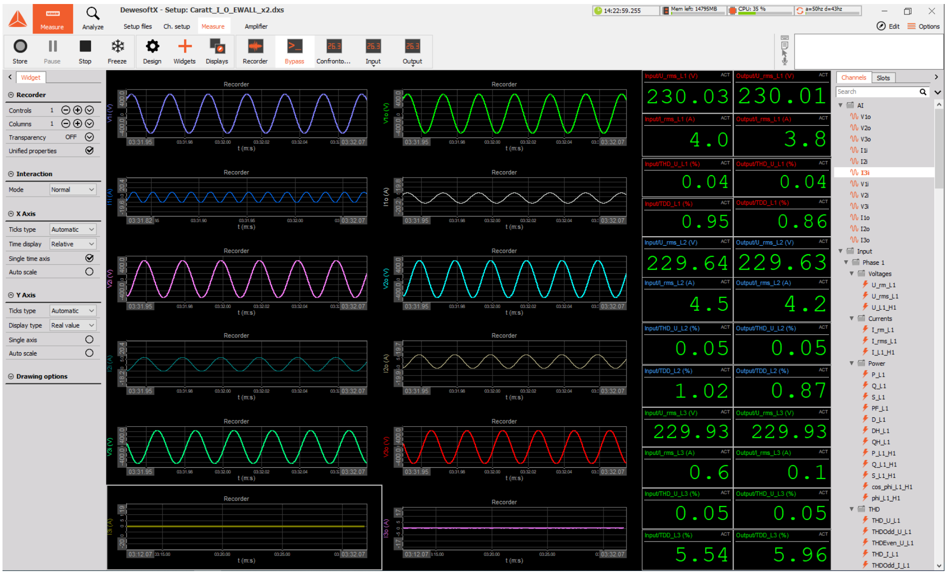


Figure 8.8: FLuxGate Transducers Setup

The voltages are measured from line to neutral at the input and output of the PQIPF. The currents are measured using the Dewesoft FluxGate Transducers as shown in Figure 8.8. The measurements recorder using the Dewesoft Sirius XHS are shown in Figure 8.9. The first column of the measurements are the inputs of voltage 1, current 1, voltage 2, current 2, voltage 3 and current 3 from top to bottom respectively. The second column of the measurements are the outputs of voltage 1, current 1, voltage 2, current 2, voltage 3 and current 3 from top to bottom respectively. On the panel on the right, PQ parameters are calculated and shown. These include, voltage and current RMS, THD and TDD. The Real Power, Reactive Power, and Apparent Power are calculated but not shown on the Dewesoft measurment panel.



DewesoftX - Setup: Caratt_I_O_EWALL_x2.dxs

Mem left: 14795MB
CPU: 35 %
a=50hz d=43hz

Recorder
Controls 1
Columns 1
Transparency OFF
Unified properties ☑

Interaction
Mode Normal

X Axis
Ticks type Automatic
Time display Relative
Single time axis ☑
Auto scale ○

Y Axis
Ticks type Automatic
Display type Real value
Single axis ○
Auto scale ○

Drawing options

Input/U_rms_L1 (V) 230.03	Output/U_rms_L1 (V) 230.01
Input/I_rms_L1 (A) 4.0	Output/I_rms_L1 (A) 3.8
Input/THD_U_L1 (%) 0.04	Output/THD_U_L1 (%) 0.04
Input/TDD_L1 (%) 0.95	Output/TDD_L1 (%) 0.86
Input/U_rms_L2 (V) 229.64	Output/U_rms_L2 (V) 229.63
Input/I_rms_L2 (A) 4.5	Output/I_rms_L2 (A) 4.2
Input/THD_U_L2 (%) 0.05	Output/THD_U_L2 (%) 0.05
Input/TDD_L2 (%) 1.02	Output/TDD_L2 (%) 0.87
Input/U_rms_L3 (V) 229.93	Output/U_rms_L3 (V) 229.93
Input/I_rms_L3 (A) 0.6	Output/I_rms_L3 (A) 0.1
Input/THD_U_L3 (%) 0.05	Output/THD_U_L3 (%) 0.05
Input/TDD_L3 (%) 5.54	Output/TDD_L3 (%) 5.96

Figure 8.9: Dewesfot Sirius XHS

8.3.3 Power Quality Passive Filter

The object under study is the Power Quality Filter that claims to filter disturbances and reduce cost caused by power quality disturbances by 4 %. This filtering device is an inductive passive filter connected in series with the loads. In theory,since it is connected in series with the loads, the input and output voltages must remain the same. On the other hand, the filter should reduce harmonic components of the input currents. The input and output currents are to be passed through the algorithm in order to evaluate the filtering capabilities of the inductive passive filter. The inductive passive filter is shown in Figure 8.10.

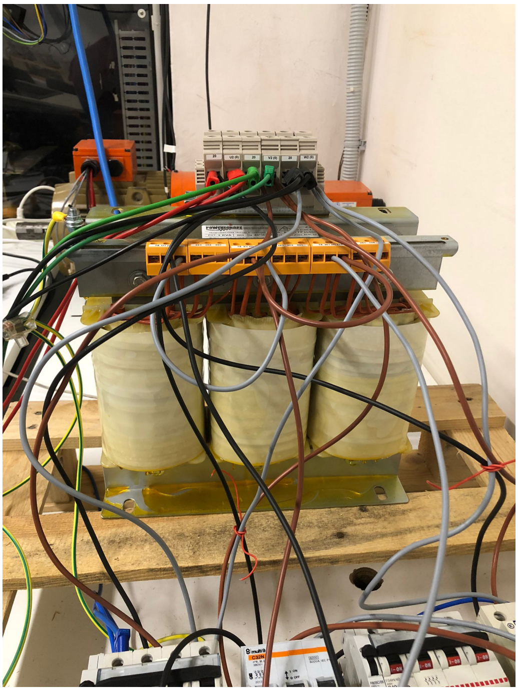


Figure 8.10: E-Wall

8.3.4 Loads

The Loads for the first part of the experiments are heating devices that are resistive loads. This is done to analyze the voltage and currents of the input and output of the inductive passive filter. The second part of the experiment is to use the resistive loads with the induction motors and some switching devices. For this part of the experiment, only the 3 phase voltages are analyzed because the current becomes to distorted and the algorithm was trained with a purely sinusoidal input. The Loads to be used are shown in Figure 8.11.

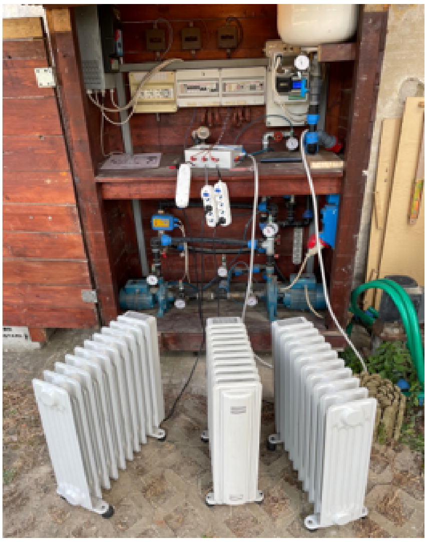


Figure 8.11: Linear Loads and Water Pumps

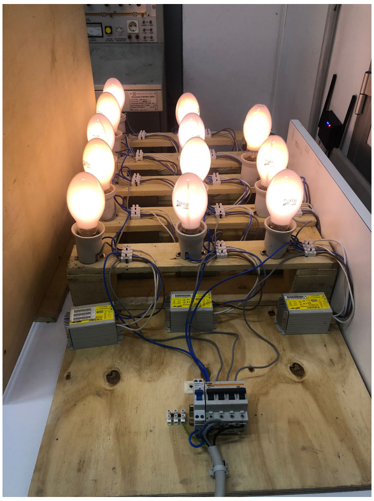


Figure 8.12: Non Linear Lighting Loads

Figure 8.13 shows the experimental setup. This figure shows the AC programmable power source, the data acquisition device and the inductive passive filter to characterize. The output is connected to the different loads presented in Figures 8.11 and Figure 8.12 which is controlled by switches installed on the panels.

Figure 8.14 shows the AC Programmable Power Source which is connected to the input of the PQIPF which is connected to a three phase switches for security reasons. The output which is also connected to switches for security reasons is connected to the linear and non linear loads via the blue cable shown. The loads are also controlled connected to switches to control the loads used in the experiments. The input and output currents are measured using the Dewesoft FluxGate transducers shown in Section 8.3.2

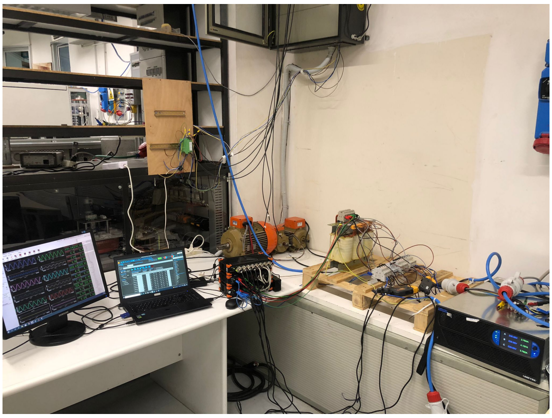


Figure 8.13: Experimental Setup

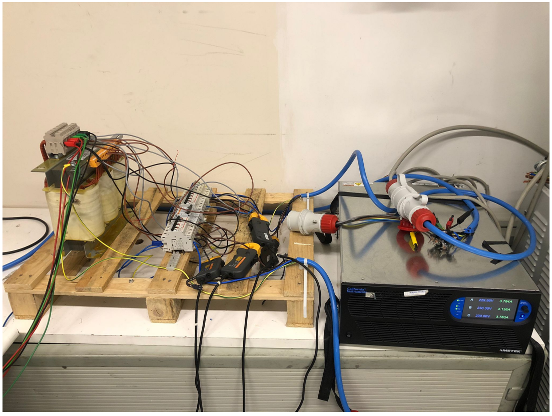


Figure 8.14: FluxGate Sensor

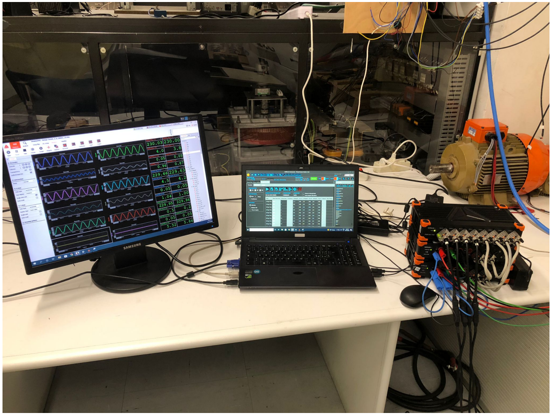


Figure 8.15: Experimental Setup

8.4 Experimental results

The experimental results were obtained by measuring the input and the output of the of the passive inductive filter. With the programmable AC power supply, disturbances were generated. These disturbances include the sag, swell and harmonic disturbances. The algorithm was implemented in the measurements obtained in order to detect and classify disturbances and extract the PQ parameters in those regions. The experiment was conducted using a linear loads and with inductive loads.

8.4.1 Results with Linear Load

In this test, sag disturbances were generated onto resistive linear loads in order to observe the effectiveness of the inductive passive filter on sag disturbances. The experiment was carried out by programming the AC programmable power source to generate sags disturbances with different duration's and amplitudes. The disturbances were Long disturbances with 1 second duration, medium disturbances with 300 milisecond, short disturbances with 150 milisecond, and micro disturbances with 10 milisecond. The durations were combined with different amplitude with 5 %, 40 %, and 90 % amplitude from its nominal value.

In Figure 8.16, the input and output voltages where measured. The SSPQDD was able to detect all disturbances according to the IEEE-1159 standard. Given the detected disturbances, the PQ parameters were calculated in order to calculate the loss given by each disturbance. As shown in Figure 8.17, the results show that the Inductive passive filter drops the V_{rms} by $5V$ in each instance. On the other hand, the current has a drop of $2mA$-$5mA$. The voltage THD remains almost constant on all instances while a reduction in the current's THD can be observed.

Figure 8.18, it is observed that the real power remains constant while a substantial reduction in reactive power is observed. Given that, a reduction in apparent power can also be appreciated.

In this test, swell disturbances were generated onto resistive linear loads in order to observe the effectiveness of the inductive passive filter on swell disturbances. Like the sag experiment, the experiment was carried out by programming the AC programmable power source to generate swells disturbances with different duration's and amplitudes. The disturbances were Long disturbances with 1 second duration, medium disturbances with 300

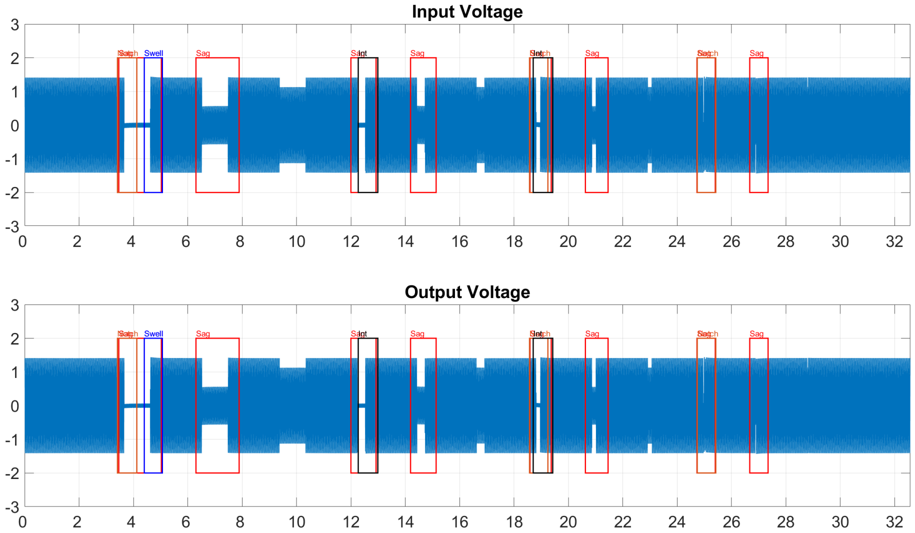


Figure 8.16: Linear Load Detection of Sag and Interruption.

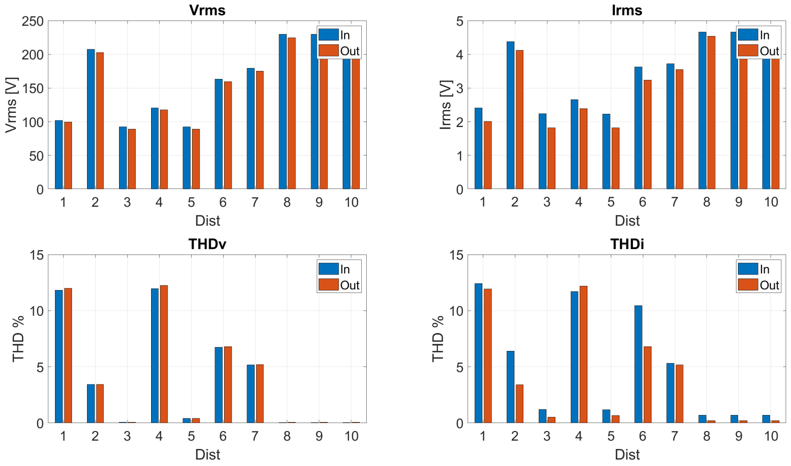


Figure 8.17: PQ Analysis, Vrms, Irms, THDv and THDi.

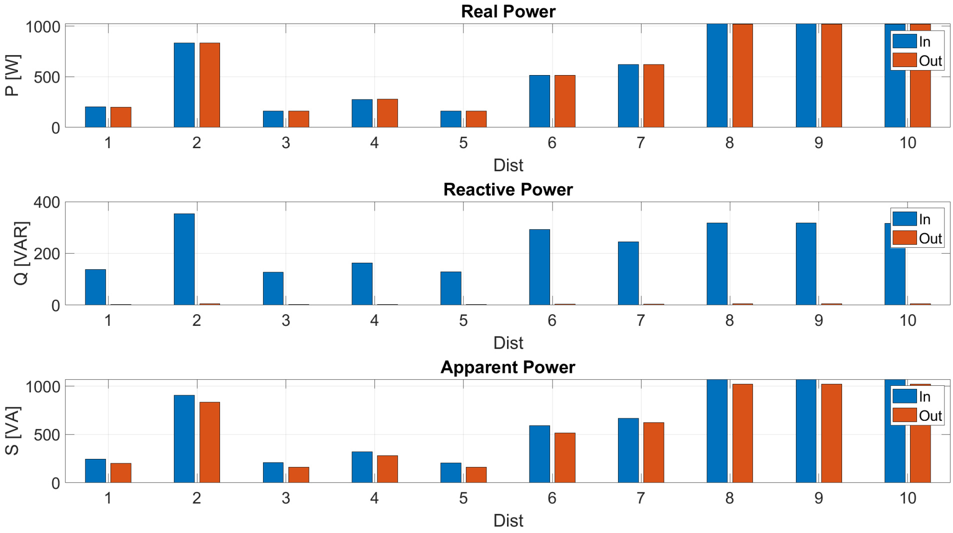


Figure 8.18: PQ Analysis power measurments.

milisecond, short disturbances with 150 milisecond, and micro disturbances with 10 milisecond. The durations were combined with different amplitude with 110 %, 120 %, and 130 % amplitude from its nominal value.

In Figure 8.19, the input and output voltages where measured. The SSPQDD was able to detect all disturbances according to the IEEE-1159 standard. In this experiment the algorithm made 4 missclassifications in the first disturbance. The following disturbances, it successfully detected all disturbances and classified a notch along the swell due to the rapid change in voltage amplitude. Given the detected disturbances, the PQ parameters were calculated in order to calculate the loss given by each disturbance. As shown in Figure 8.20, the results show that the Inductive passive filter drops the V_{rms} by $5V$ in each instance. On the other hand, the current has a drop of $2mA$-$5mA$. The voltage THD remains almost constant on all instances while a substantial reduction in the current's THD can be observed.

Figure 8.21, it is observed that the real power remains constant while a substantial reduction in reactive power is observed. Given that, a reduction in apparent power can also be appreciated.

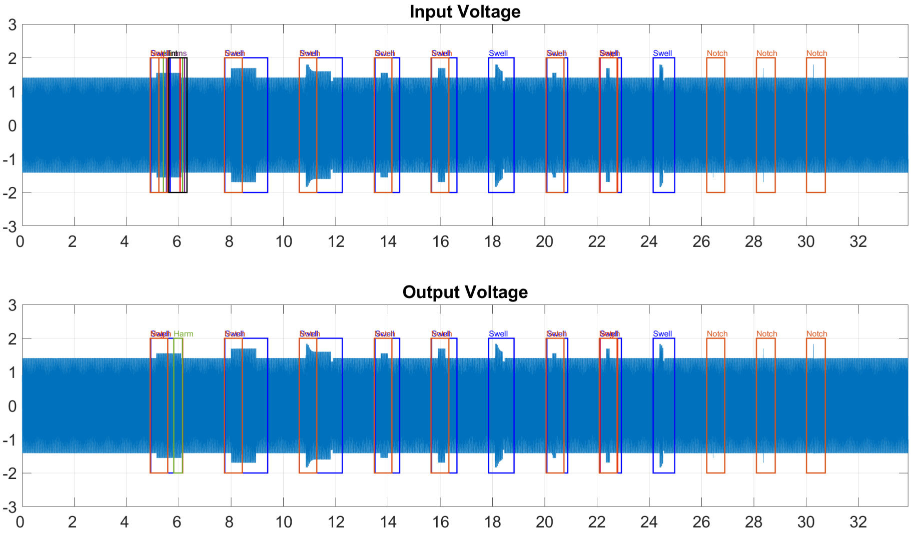


Figure 8.19: Linear Load Detection of Swell.

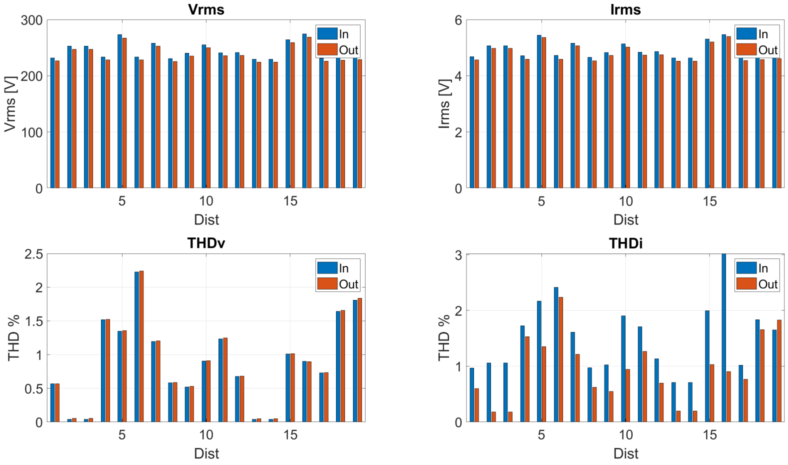


Figure 8.20: PQ Analysis, Vrms, Irms, THDv and THDi.

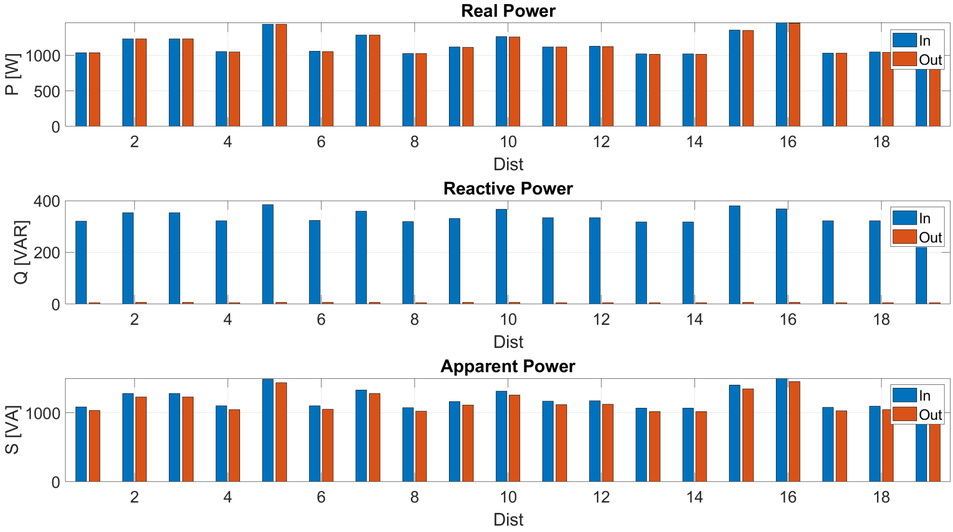


Figure 8.21: PQ Analysis power measurments.

In this test, harmonic disturbances were generated into resistive linear loads in order to observe the effectiveness of the inductive passive filter on harmonic disturbances. Like the sag and swell experiment, the experiment was carried out by programming the AC programmable power source to generate harmonic disturbances with different duration's and harmonic components. The disturbances were Long disturbances with 1 second duration, medium disturbances with 300 milisecond, short disturbances with 150 milisecond, and micro disturbances with 10 milisecond. The durations were combined with different harmonic components being:

- Harm 1 : $2^{nd} \rightarrow 50\%$
- Harm 2 : $10^{nd} \rightarrow 40\%$
- Harm 3 : $25^{nd} \rightarrow 30\%$
- Harm 4 : $30^{nd} \rightarrow 20\%$
- Harm 5 : $35^{nd} \rightarrow 2\%$

- Harm 6 : $40^{nd} \rightarrow 2\%$
- Harm 7 : $45^{nd} \rightarrow 2\%$
- Harm 8 : $50^{nd} \rightarrow 2\%$

In Figure 8.22, the input and output voltages where measured. The SSPQDD was able to detect all disturbances according to the IEEE-1159 standard. In this experiment the algorithm made some unexpected classifications. In the first harmonic distortion, the algorithm detected disturbances that lasts milliseconds as shown in Figure 8.25. It successfully detected all disturbances and classified swell, notch and transient along with the harmonic due to the increase in voltage and the high frequency components with short duration. Given the detected disturbances, the PQ parameters were calculated in order to calculate the loss given by each disturbance. As shown in Figure 8.23, the results show that the Inductive passive filter also drops the V_{rms} by $5V$ in each instance as the sag and the swell. On the other hand, the current has a drop of $1mA$. The voltage THD remains almost constant on all instances as well for the currents THD.

Figure 8.24, it is observed that the real power remains constant while a substantial reduction in reactive power is observed. Given that, a reduction in apparent power can also be appreciated.

8.4.2 Results with Inductive Loads

In this test, sag disturbances were generated onto resistive linear loads in order to observe the effectiveness of the inductive passive filter on sag disturbances. The experiment was carried out by programming the AC programmable power source to generate sags disturbances with different duration's and amplitudes. The disturbances were Long disturbances with 1 second duration, medium disturbances with 300 milisecond, short disturbances with 150 milisecond, and micro disturbances with 10 milisecond. The durations were combined with different amplitude with 5 %, 40 %, and 90 % amplitude from its nominal value.

In Figure 8.26, the input and output voltages where measured. The SSPQDD was able to detect all disturbances according to the IEEE-1159 standard. Given the detected disturbances, the PQ parameters were calculated in order to calculate the loss given by each disturbance. As shown in Figure 8.27, the results show that the Inductive passive filter drops the

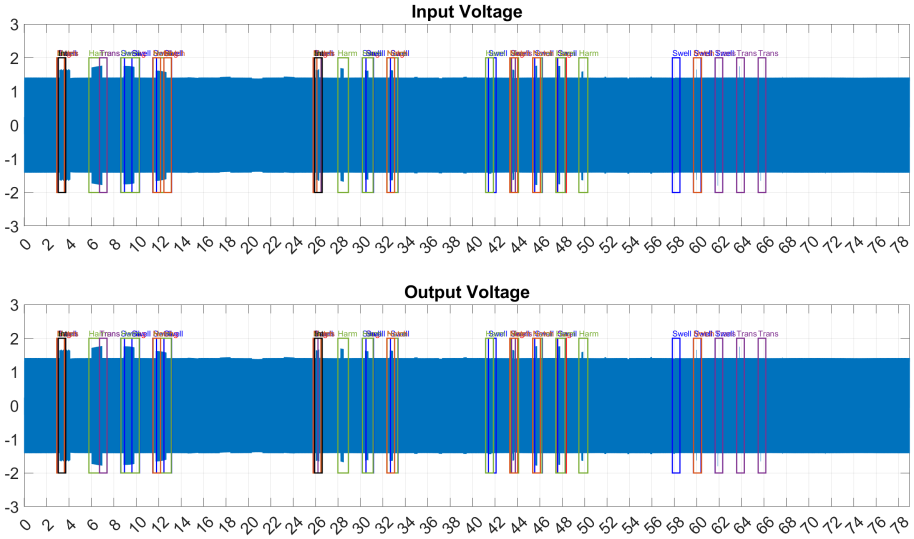


Figure 8.22: Linear Load Detection of Harmonics and Transients.

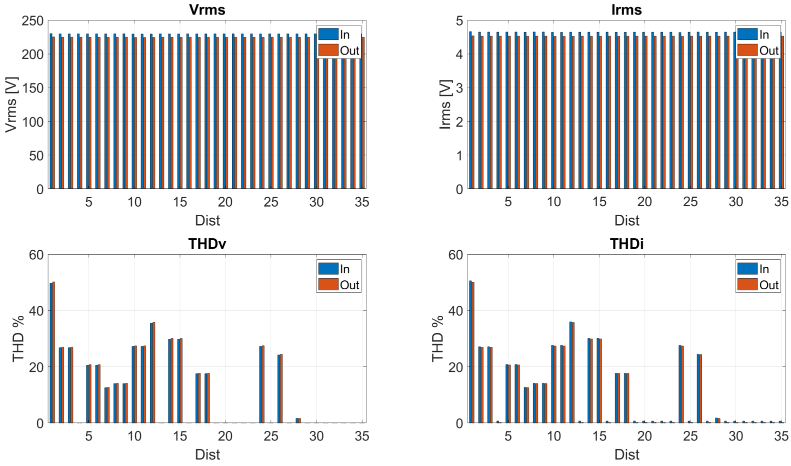


Figure 8.23: PQ Analysis, Vrms, Irms, THDv and THDi.

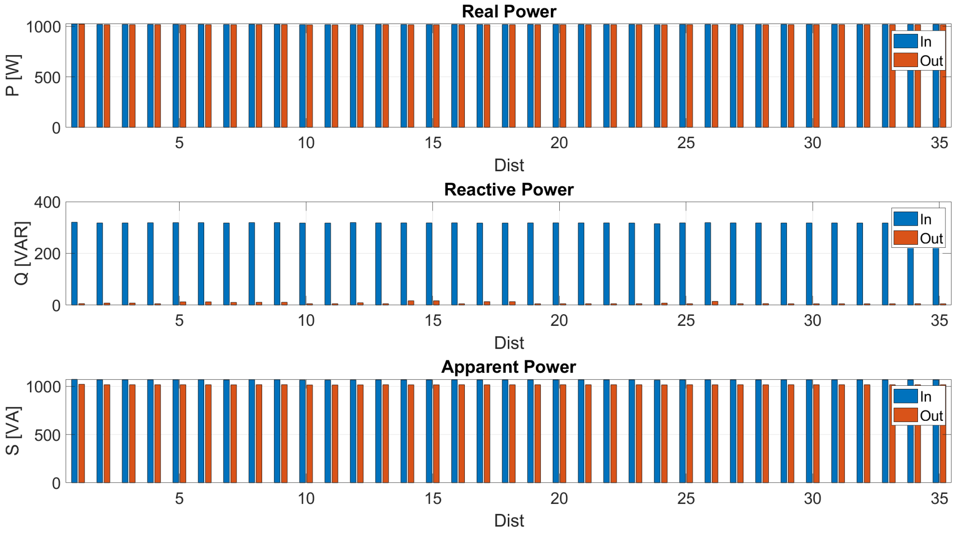


Figure 8.24: PQ Analysis power measurments.

V_{rms} by $5V$ in each instance. On the other hand, the current has a drop of $2mA$-$5mA$. The voltage THD remains almost constant on all instances while a reduction in the current's THD can be observed.

Figure 8.28, it is observed that the real power remains constant while a substantial reduction in reactive power is observed. Given that, a reduction in apparent power can also be appreciated.

In this test, swell disturbances were generated onto resistive linear loads in order to observe the effectiveness of the inductive passive filter on swell disturbances. Like the sag experiment, the experiment was carried out by programming the AC programmable power source to generate swells disturbances with different duration's and amplitudes. The disturbances were Long disturbances with 1 second duration, medium disturbances with 300 milisecond, short disturbances with 150 milisecond, and micro disturbances with 10 milisecond. The durations were combined with different amplitude with 110 %, 120 %, and 130 % amplitude from its nominal value.

In Figure 8.29, the input and output voltages where measured. The SSPQDD was able to detect all disturbances according to the IEEE-1159

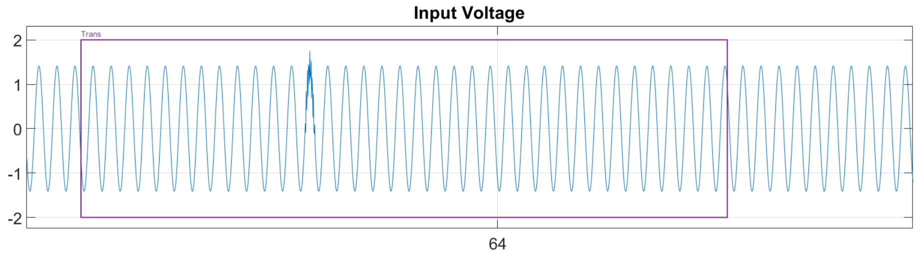


Figure 8.25: Transient distortion zoom

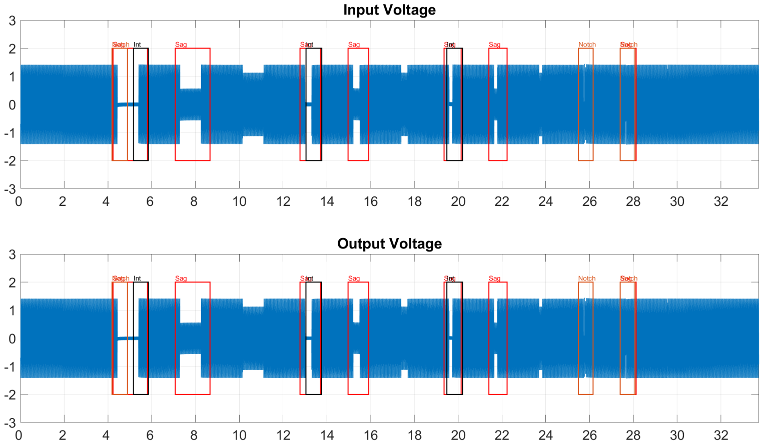


Figure 8.26: Inductive Load Detection of Sag and Interruption.

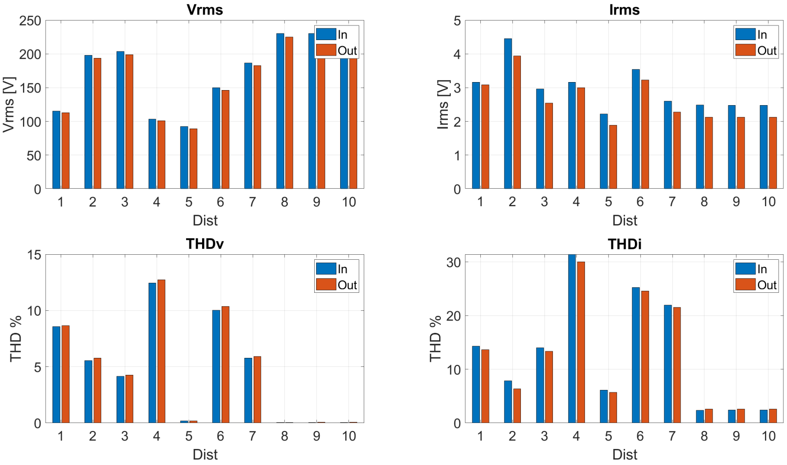


Figure 8.27: PQ Analysis, Vrms, Irms, THDv and THDi.

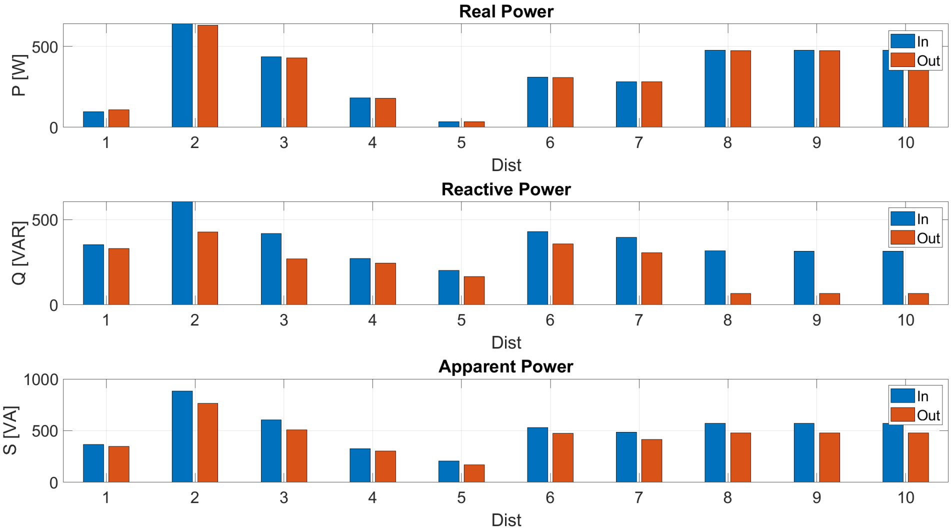


Figure 8.28: PQ Analysis power measurments.

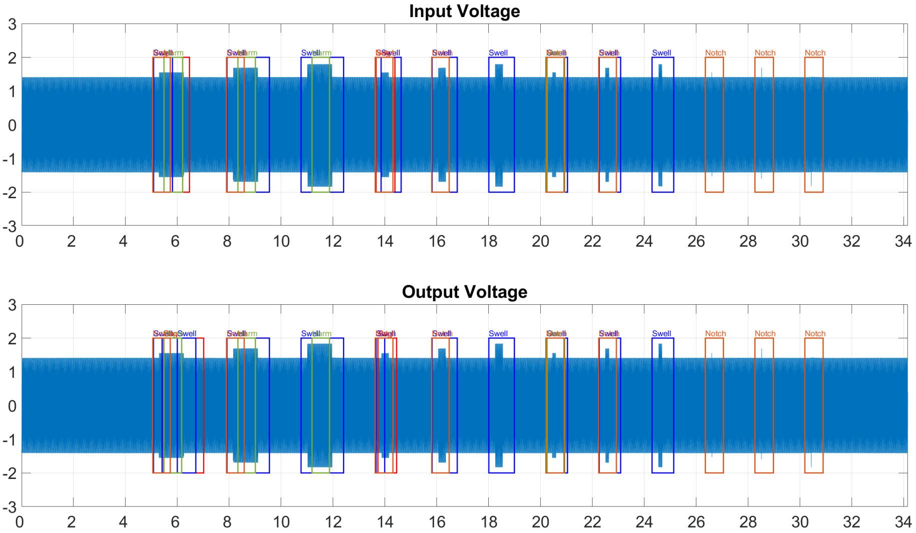


Figure 8.29: Inductive Load Detection of Swell.

standard. In this experiment the algorithm made 6 missclassifications in the first and fourth disturbance where it classified a sag. The following disturbances, it successfully detected all disturbances and classified a notch and a harmonic along with the swell due to the rapid change in voltage amplitude. Given the detected disturbances, the PQ parameters were calculated in order to calculate the loss given by each disturbance. As shown in Figure 8.30, the results show that the Inductive passive filter drops the V_{rms} by $5V$ in each instance. On the other hand, the current has a drop of $2mA$-$5mA$. The voltage THD remains almost constant on all instances while a substantial reduction in the current's THD can be observed.

Figure 8.31, it is observed that the real power remains constant while a substantial reduction in reactive power is observed. Given that, a reduction in apparent power can also be appreciated.

In this test, swell disturbances were generated onto the inductive loads in order to observe the effectiveness of the inductive passive filter on harmonic disturbances. Like the sag and swell experiment, the experiment was carried out by programming the AC programmable power source to generate

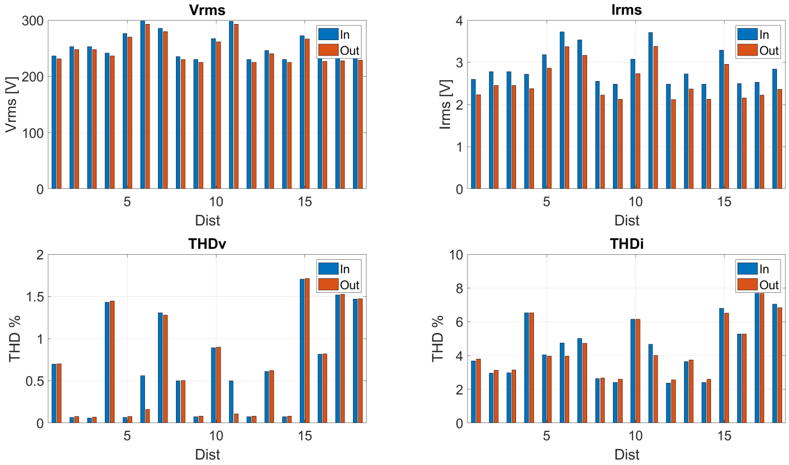


Figure 8.30: PQ Analysis, Vrms, Irms, THDv and THDi.

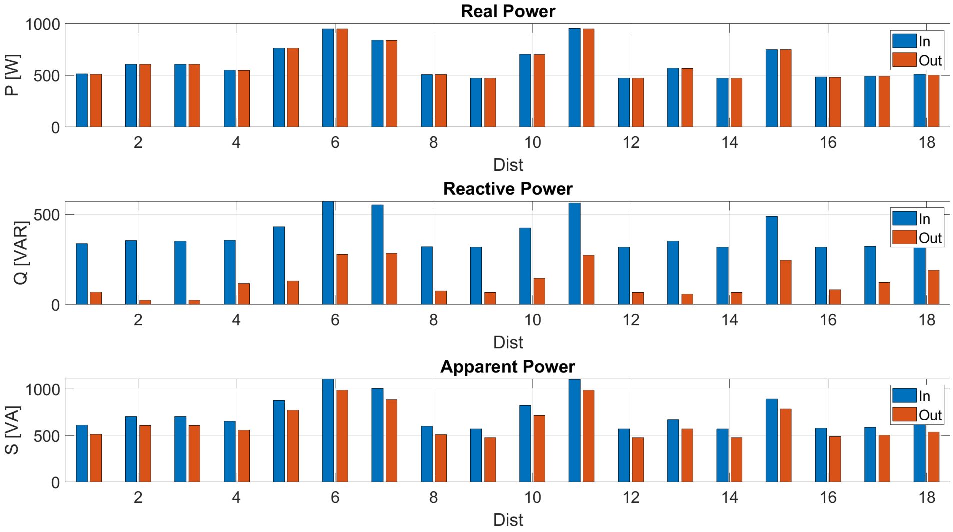


Figure 8.31: PQ Analysis power measurments.

harmonic disturbances with different duration's and harmonic components. The disturbances were Long disturbances with 1 second duration, medium disturbances with 300 milisecond, short disturbances with 150 milisecond, and micro disturbances with 10 milisecond. The durations were combined with different harmonic components being:

- Harm 1 : $2^{nd} \rightarrow 50\%$
- Harm 2 : $10^{nd} \rightarrow 40\%$
- Harm 3 : $25^{nd} \rightarrow 30\%$
- Harm 4 : $30^{nd} \rightarrow 20\%$
- Harm 5 : $35^{nd} \rightarrow 2\%$
- Harm 6 : $40^{nd} \rightarrow 2\%$
- Harm 7 : $45^{nd} \rightarrow 2\%$
- Harm 8 : $50^{nd} \rightarrow 2\%$

In Figure 8.32, the input and output voltages where measured. The SSPQDD was able to detect all disturbances according to the IEEE-1159 standard. In this experiment the algorithm made some unexpected classifications. In the first harmonic distortion, the algorithm detected micro interruptions as shown in Figure. It successfully detected all disturbances and classified swell, notch and transient along with the harmonic due to the increase in voltage and the high frequency components with short duration. Given the detected disturbances, the PQ parameters were calculated in order to calculate the loss given by each disturbance. As shown in Figure 8.33, the results show that the Inductive passive filter also drops the V_{rms} by $5V$ in each instance as the sag and the swell. On the other hand, the current has a drop of $1mA$. The voltage THD remains almost constant on all instances as well for the currents THD.

Figure 8.34, it is observed that the real power remains constant while a substantial reduction in reactive power is observed. Given that, a reduction in apparent power can also be appreciated.

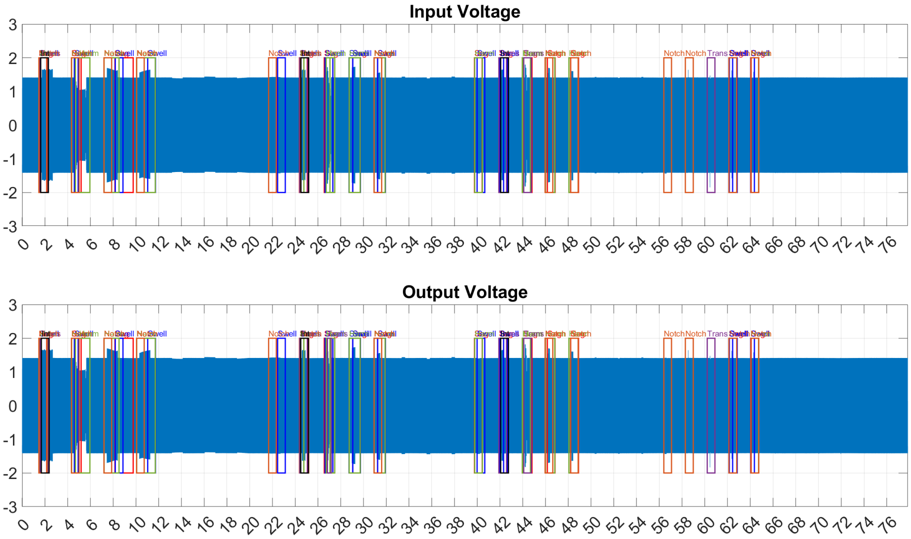


Figure 8.32: Inductive Load Detection of Harmonics and Transients.

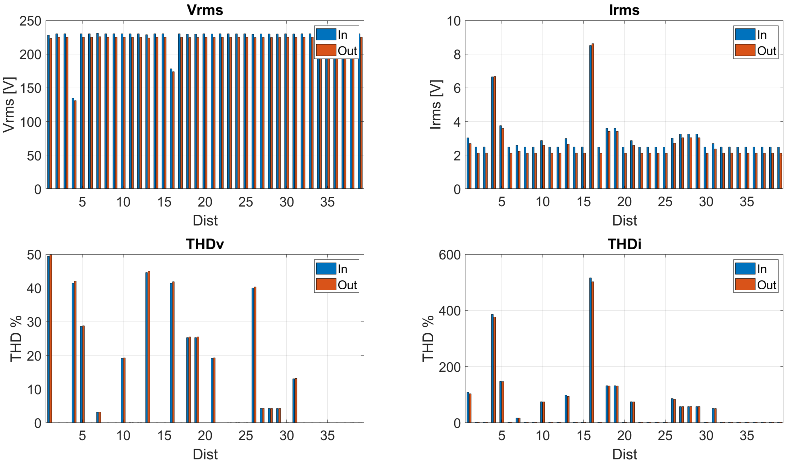


Figure 8.33: PQ Analysis, Vrms, Irms, THDv and THDi.

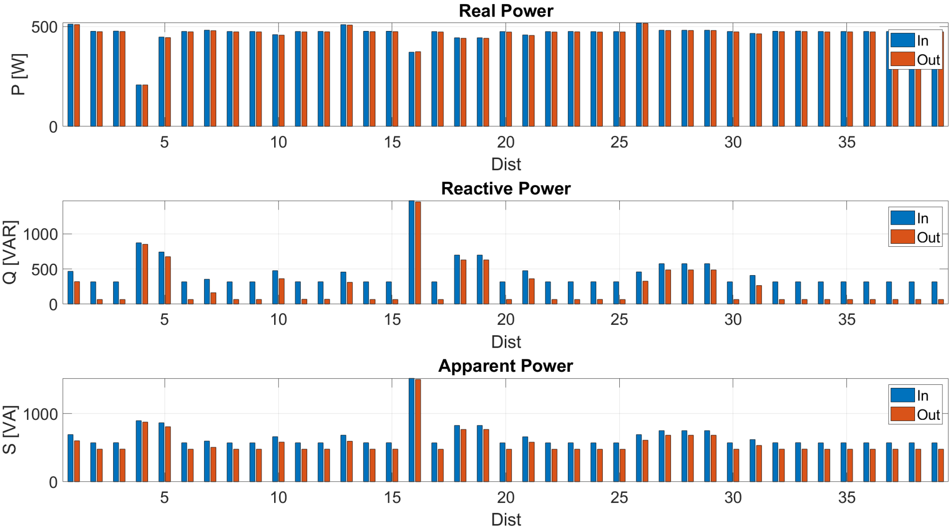


Figure 8.34: PQ Analysis power measurments.

8.5 Conclusion

This chapter discusses the use of the SSPQDD as a tool for system identification. This algorithm is used to detect and classify PQDs at the input and at the output of an inductive passive filter used to filter PQD. The goal is to quantify the performance of the PQD filter and measure the efficiency. For this, an experiment was designed as shown in Section 8.3. An experimental testbench was created in which the a programmable AC power source which is able to create PQD is connected to the inductive passive filter connected to linear, inductive linear and non linear loads. The input and the output was measured with a sampling rate of 8kHz.

Results shown a reduction in reactive power which should correct the power fact in each experiment made. Also, it proved to have significant improvement on the THD in the current. This is due to the fact that the inductive passive filter is connected in series with the loads. So the improvement should be appreciated in the current output. On average, the improvement in THD for the current is of 3% which is significant for the lifespan of electrical and electronic components. It can also be noted that the voltage signal does not show improvements other than a voltage drop of 5 V. Finally, the difference between the input and output of the apparent power is caused by the reduction of reactive power.

Chapter 9

Conclusion

9.1 Summary of contribution

This work, studies machine learning and deep learning algorithms for detection and classification of Power Quality Disturbances. Different architectures were tested including the LSTM, CNN and CNN-LSTM. All of this algorithms have shown to be effective and efficient at detecting and classifying PQD. Feature extraction techniques were also implemented to enhance the classification capabilities of the CNN.

Since the hallmark of the CNN is image classification, the STFT was implemented to convert the time series signal to a time-frequency 2D matrix. This has shown a small improvement to the output results. Due to the fact that the CNN uses convolutions in each layer that extract the frequency and magnitude components of the input signals, it is used as the base of the SSPQDD algorithm.

The SSPQDD was developed for the detection and classification of PQD. The SSPQDD uses a VGG16 CNN architecture for the feature extraction part and layers were then added for the detection and classification of PQD. This PQD detection and classification algorithm was developed, trained and tested using object detection algorithms like the SSD and YOLO as inspiration. This algorithm was compared with high performance algorithms as the VGG16, Alexnet, Darknet and Resnet. This algorithm outperforms these top tear algorithms in this application due to the fact of the nature of the architectures. The SSPQDD algorithm uses filters that extract 16 samples and is able to detect and classify transient disturbances. A visual comparison was made with these algorithms in which these top tear architectures fail to detect and classify multiple disturbance in a single time window.

This algorithm was used as a tool to detect and classify PQD in order to quantify the performance of an inductive passive filter. The purpose of this experiment is to use this algorithm to detect the disturbance at the input and output of the filter and calculate its PQ parameters on each of the disturbances detected in order to calculate the difference of the input and output of the PQ parameters. The algorithm was able to detect and classify most of the disturbances generated by the AC programmabe power source. Some were misclassifications and some were not able to detect. The overall results were that the algorithm was able to detect and classify the disturbances and the PQ paramenters were calculated in the time windows detected by the algorithm.

9.2 Directions for future work

For future work, the algorithm is to be trained with a more wide spectrum of disturbances in order to increase the precision and recall of the SSPQDD. This algorithm will be implemented in a cloud based real time PQ meter in which it will be deployed continuously in a real time IoT device. It will be monitored continuously and studied in order to prevent miss classifications due to data drifting.

Appendix A

Appendix

This appendix is related to the Single Shot Power Quality Disturbance Detector, previously presented in Chapter 6.

A.1 Detection and Classification of Power Quality Disturbances

This Section contains the validation of the training procedure for the SSPQDD. The data used was simulated using the simulations presented in Chapter 3. The validation was made using the disturbances studied containing different durations and in different positions in the window frame.

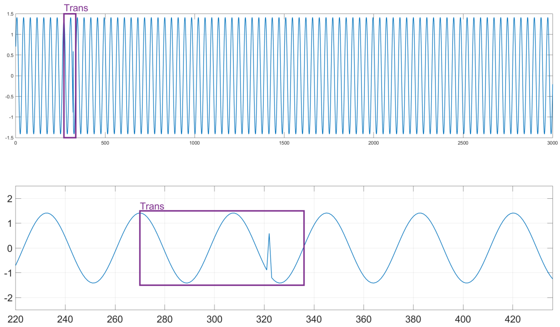


Figure A.1: Transient Disturbance

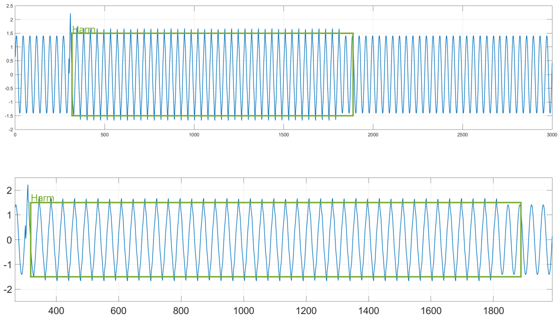


Figure A.2: Harmonic Disturbance

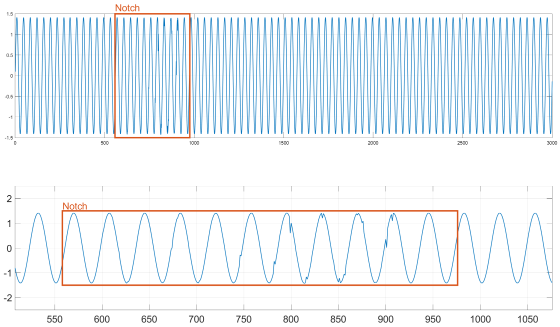


Figure A.3: Notch Disturbance

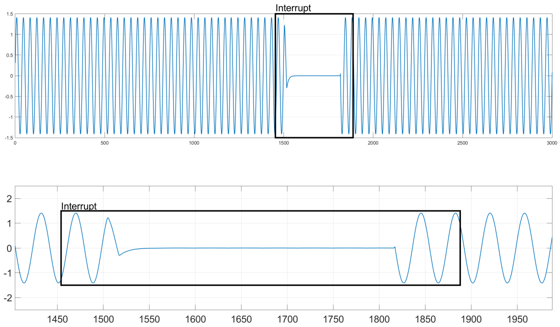


Figure A.4: Interruption Disturbance

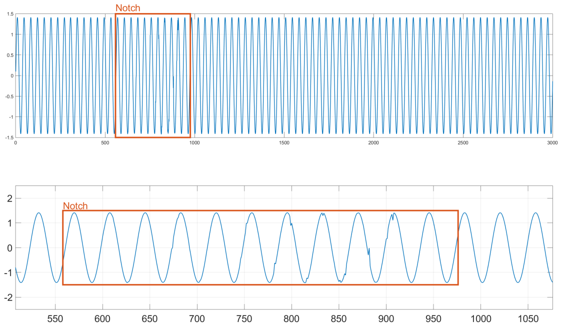


Figure A.5: Notch Disturbance

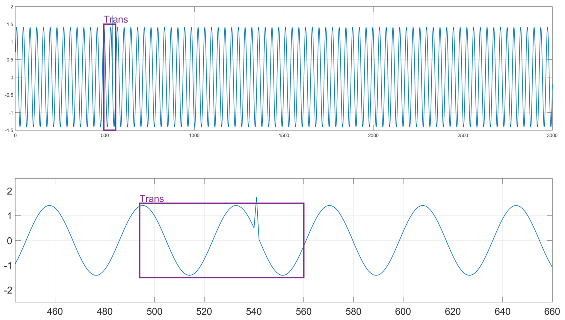


Figure A.6: Transient Disturbance

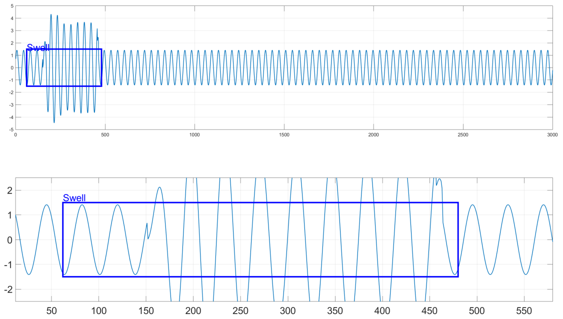


Figure A.7: Swell Disturbance

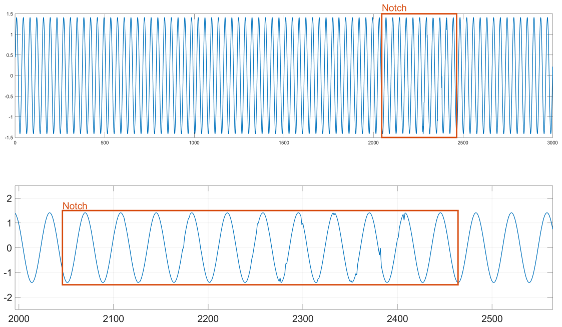


Figure A.8: Notch Disturbance

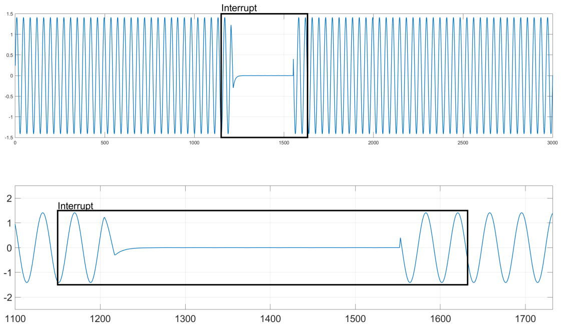


Figure A.9: Interruption Disturbance

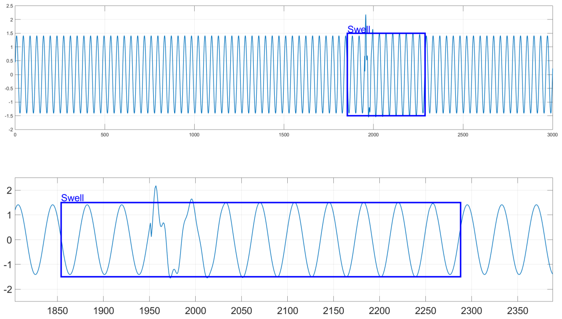


Figure A.10: Swell Disturbance

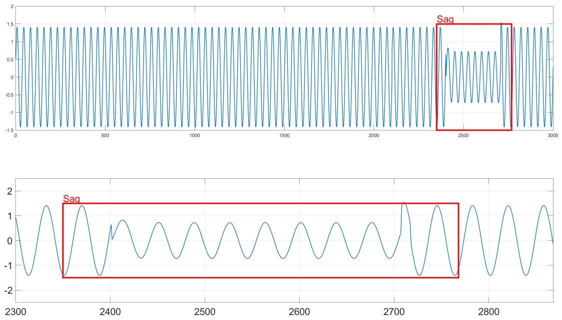


Figure A.11: Sag Disturbance

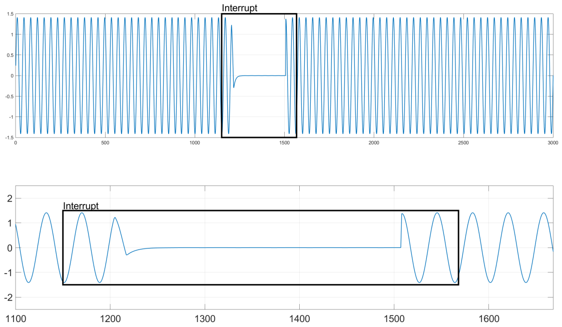


Figure A.12: Interruption Disturbance

A.2 Multiple Disturbance in Window Frame

This Section contains the validation of the SSPQDD with multiple disturbances in a window frame. The signals were generated randomly inserting different disturbances used in the training phase.

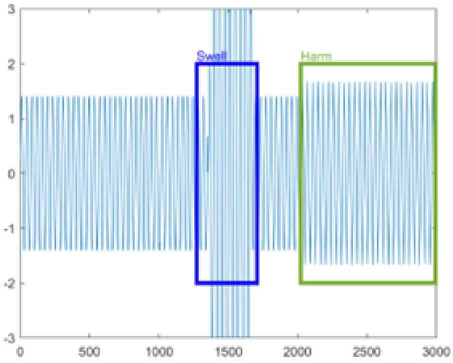


Figure A.13: Swell and Harmonic Disturbance

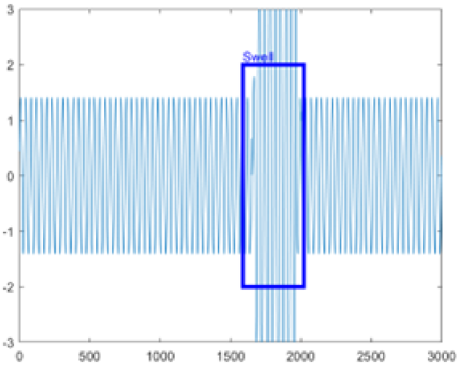


Figure A.14: Swell Disturbance

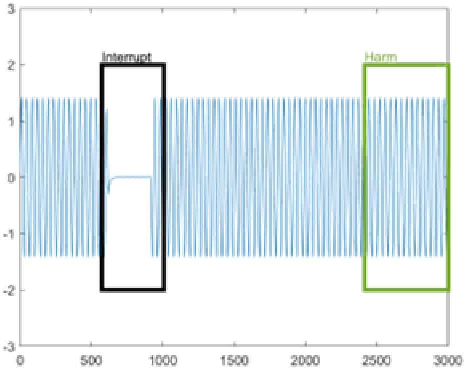


Figure A.15: Interruption and Harmonics Disturbance

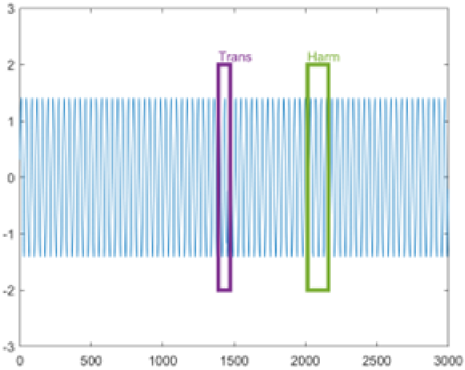


Figure A.16: Transient and Harmonics Disturbance

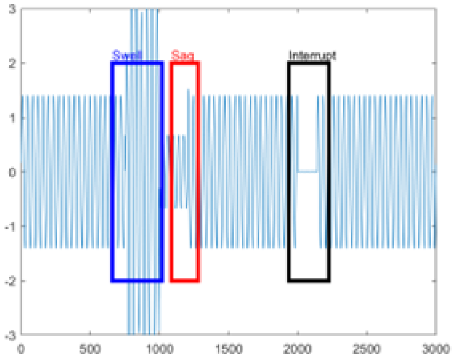


Figure A.17: Swell, Sag and Interruption Disturbance

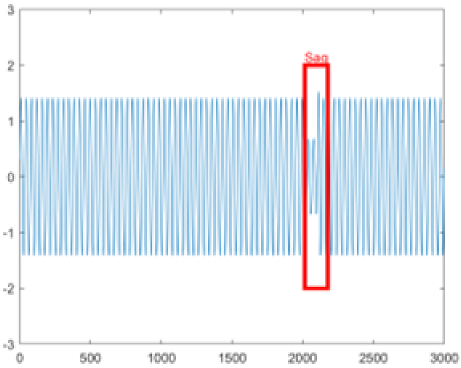


Figure A.18: Sagl Disturbance

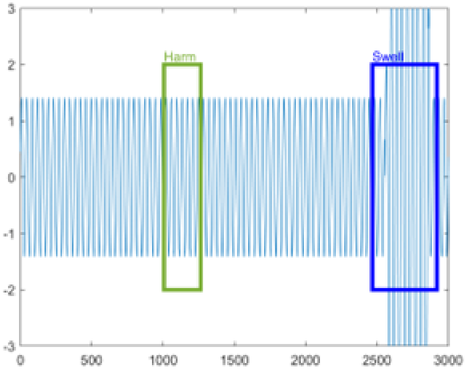


Figure A.19: Harmonics and Swell Disturbance

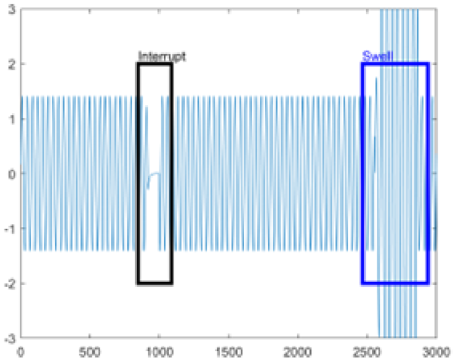


Figure A.20: Interruption and Swell Disturbance

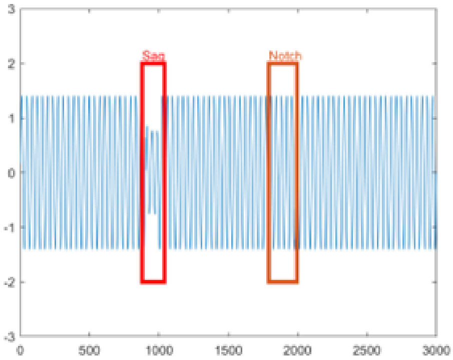


Figure A.21: Sag and Notch Disturbance

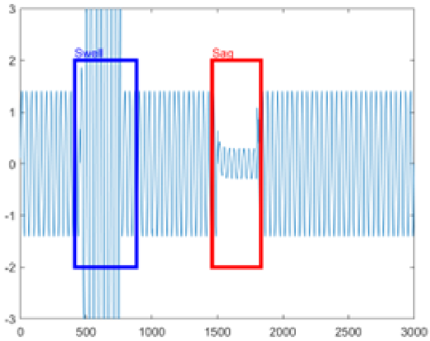


Figure A.22: Swell and Sag Disturbance

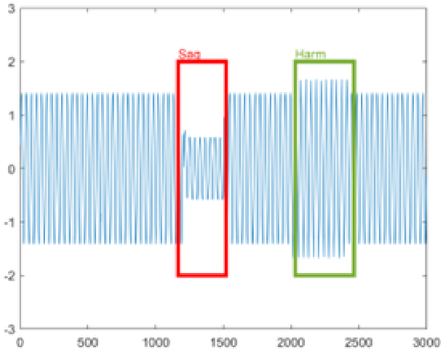


Figure A.23: Sag and Harmonics Disturbance

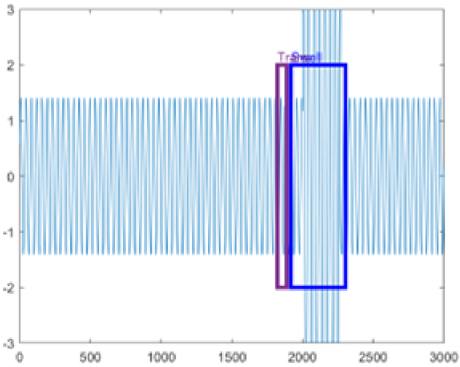


Figure A.24: Transient and Swell Disturbance

Appendix B

Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.

International Journals

1. C. Iturrino-García et al., "An Innovative Single Shot Power Quality Disturbance Detector Algorithm," in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-10, 2022, Art no. 2517210.
2. Talluri, G.; Lozito, G.M.; Grasso, F.; Iturrino García, C.; Luchetta, A. Optimal Battery Energy Storage System Scheduling within Renewable Energy Communities. Energies 2021, 14, 8480. https://doi.org/10.3390/en14248480
3. García, C.I.; Grasso, F.; Luchetta, A.; Piccirilli, M.C.; Paolucci, L.; Talluri, G. A Comparison of Power Quality Disturbance Detection and Classification Methods Using CNN, LSTM and CNN-LSTM. Appl. Sci. 2020, 10, 6755.

Submitted

1. C. Iturrino-García et al., Power Quality Analysis Based on Machine Learning Methods for Low Voltage Electrical Distribution Lines, SSRN Electron. J., 2022. “Hopefully the paper with this title will be accepted”

International Conferences and Workshops

1. M. Bindi et al., “Classification of Power Quality disturbances using Multi-Valued Neural Networks and Convolutional Neural Networks,” 2022 International Joint Conference on Neural Networks (IJCNN), 2022, pp. 01-08, doi: 10.1109/IJCNN55064.2022.9892536.

2. F. Grasso, C. I. Garcia, G. M. Lozito and G. Talluri, “Artificial Load Profiles and PV Generation in Renewable Energy Communities Using Generative Adversarial Networks,” 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON), 2022, pp. 709-714, doi: 10.1109/MELECON53508.2022.9843062.

3. G. Patrizi et al., “Reliability Prediction of an innovative Power Quality Meter,” 2022 IEEE International Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 and IoT), 2022, pp. 195-200, doi: 10.1109/MetroInd4.0IoT54413.2022.9831774.

4. M. Bindi et al., ”Comparison Between PI and Neural Network Controller for Dual Active Bridge Converter,” 2021 IEEE 15th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), 2021, pp. 1-6.

5. C. Iturrino, F. X. Arias, H. Sierra and E. Arzuaga, “Single-Shot Multispectral Image Acquisition for Low-Altitude Remote Sensing using Light Diffraction Techniques,” 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2019, pp. 1-5, doi: 10.1109/WHISPERS.2019.8920879.

Bibliography

[1] A. Akbarpour, ·. Mehdi Nafar, and M. Simab, "O R I G I N A L P A P E R Multiple power quality disturbances detection and classification with fluctuations of amplitude and decision tree algorithm," *Electrical Engineering.*

[2] Y. Zhang, Y. Zhang, and X. Zhou, "Classification of power quality disturbances using visual attention mechanism and feed-forward neural network," *Measurement*, vol. 188, p. 110390, jan 2022.

[3] D. Sharon, S. Member, J.-C. Montaño, A. López, M. Castilla, D. Borrás, and J. Gutiérrez, "Power Quality Factor for Networks Supplying Unbalanced Nonlinear Loads; Power Quality Factor for Networks Supplying Unbalanced Nonlinear Loads," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, vol. 57, no. 6, 2008.

[4] M. I. Muhamad, N. Mariun, and M. A. M. Radzi, "The effects of power quality to the industries," *2007 5th Student Conference on Research and Development, SCORED*, no. December, pp. 12–15, 2007.

[5] R. S. Kumar, I. G. Christopher Raj, S. Saravanan, P. Leninpugalhanthi, and P. Pandiyan, *Impact of power quality issues in residential systems.* Elsevier Inc., 2021.

[6] B. Singh, K. Al-Haddad, and A. Chandra, "A Review of Active Filters for Power Quality Improvement," Tech. Rep. 5, 1999.

[7] G. Benysek, *Improvement in the Quality of Delivery of Electrical Energy using Power Electronics Systems.* Power Systems, Springer London, 2007.

[8] IEEE Power and Energy Society, *IEEE Recommended Practice for Monitoring Electric Power Quality*, vol. 1995. 2009.

[9] Pinyol Ramon, "Harmonics : Causes , Effects and Minimization," *Salicru White Papers*, no. August, pp. 1–32, 2015.

[10] IEEE, "IEEE Std 519-2014: IEEE Recommended Practice and Requirements for Harmonic Control," *ANSI/IEEE Std. 519*, vol. 2014, pp. 5–9, 2014.

[11] M. H. Bollen, "What is power quality?," *Electric Power Systems Research*, vol. 66, pp. 5–14, jul 2003.

[12] G. T. Heydt, P. S. Fjeld, C. C. Liu, D. Pierce, L. Tu, and G. Hensley, "Applications of the windowed FFT to electric power quality assessment," *IEEE Transactions on Power Delivery*, vol. 14, pp. 1411–1416, oct 1999.

[13] J. Mroczka, "DFT algorithm analysis in low-cost power quality measurement systems based on a DSP processor," tech. rep.

[14] R. R. Panigrahi, M. Mishra, J. Nayak, V. Shanmuganathan, B. Naik, and Y. A. Jung, "A power quality detection and classification algorithm based on FDST and hyper-parameter tuned light-GBM using memetic firefly algorithm," *Measurement*, vol. 187, p. 110260, jan 2022.

[15] C. I. Garcia, F. Grasso, A. Luchetta, M. C. Piccirilli, L. Paolucci, and G. Talluri, "A comparison of power quality disturbance detection and classification methods using CNN, LSTM and CNN-LSTM," *Applied Sciences (Switzerland)*, vol. 10, no. 19, pp. 1–22, 2020.

[16] S. Khokhar, A. A. Mohd Zin, A. P. Memon, and A. S. Mokhtar, "A new optimal feature selection algorithm for classification of power quality disturbances using discrete wavelet transform and probabilistic neural network," *Measurement*, vol. 95, pp. 246–259, jan 2017.

[17] B. Eristi, O. Yildirim, . H. Eristi, and Y. Demir, "A new embedded power quality event classification system based on the wavelet transform," 2018.

[18] K. Thirumala, M. Siva Prasad, T. Jain, and A. C. Umarikar, "Tunable-Q Wavelet Transform and Dual Multiclass SVM for Online Automatic Detection of Power Quality Disturbances," *IEEE Transactions on Smart Grid*, vol. 9, pp. 3018–3028, jul 2018.

[19] J. Li, Z. Teng, Q. Tang, and J. Song, "Detection and Classification of Power Quality Disturbances Using Double Resolution S-Transform and DAG-SVMs; Detection and Classification of Power Quality Disturbances Using Double Resolution S-Transform and DAG-SVMs," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 10, 2016.

[20] O. P. Mahela and A. G. Shaik, "Power quality recognition in distribution system with solar energy penetration using S-transform and Fuzzy C-means clustering," *Renewable Energy*, vol. 106, pp. 37–51, jun 2017.

[21] M. J. B. Reddy, K. Sagar, and D. K. Mohanta, "A multifunctional real-time power quality monitoring system using Stockwell transform," *IET Science, Measurement & Technology*, vol. 8, pp. 155–169, jul 2014.

[22] R. Kumar, B. Singh, D. T. Shahani, A. Chandra, and K. Al-Haddad, "Recognition of Power-Quality Disturbances Using S-Transform-Based ANN Classi-

fier and Rule-Based Decision Tree," *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS*, vol. 51, no. 2.

[23] R. Kumar, R. Kumar, S. Marwaha, M. C. In, and B. Singh, *S-Transform Based Detection of Multiple and Multistage Power Quality Disturbances; S-Transform Based Detection of Multiple and Multistage Power Quality Disturbances.* 2020.

[24] M. Sahani and P. K. Dash, "Automatic Power Quality Events Recognition Based on Hilbert Huang Transform and Weighted Bidirectional Extreme Learning Machine; Automatic Power Quality Events Recognition Based on Hilbert Huang Transform and Weighted Bidirectional Extreme Learning Machin," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, 2018.

[25] M. Jasa Afroni, D. Sutanto, S. Member, and D. Stirling, "Analysis of Nonstationary Power-Quality Waveforms Using Iterative Hilbert Huang Transform and SAX Algorithm," *IEEE Transactions on Power Delivery*, vol. 28, no. 4, 2013.

[26] W. Cochran, J. Cooley, D. Favin, H. Helms, R. Kaenel, W. Lang, G. Maling, D. Nelson, C. Rader, and P. Welch, "What Is the Fast Fourier Transform?," tech. rep., FFT, 1967.

[27] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," tech. rep.

[28] S. Santoso, W. M. Grady, S. Member, E. J. Powers, J. Lamoree, and S. C. Bhatt, "Characterization of Distribution Power Quality Events with Fourier and Wavelet Transforms," vol. 15, no. 1, pp. 247–254, 2000.

[29] U. Singh and S. N. Singh, "Application of Fractional Fourier Transform for Classification of Power Quality Application of fractional Fourier transform for classification of power quality disturbances," no. January, 2021.

[30] C. C. Liu, D. Pierce, L. Tu, and G. Hensley, "Applications of the Windowed PFT to Electric Power Quality Assessment," vol. 14, no. 4, pp. 1411–1416, 1999.

[31] F. Jurado and R. Saenz, "Comparison between discrete STFT and wa v elets for the analysis of power quality e v ents," vol. 62, 2002.

[32] C. Zhao, M. He, and X. Zhao, "Analysis of Transient Waveform Based on Combined Short Time Fourier Transform and Wavelet Trans form," no. November, pp. 21–24, 2004.

[33] M. Sifuzzaman, M. R. Islam, and M. Z. Ali, "Application of Wavelet Transform and its Advantages Compared to Fourier Transform," *Journal of Physical Sciences*, vol. 13, pp. 121–134, 2009.

[34] S. Santoso, S. Member, E. J. Powers, and P. H. Member, "Power quality assessment via wavelet transform analysis," vol. 11, no. 2, pp. 924–930, 1996.

[35] P. K. Dash, B. K. Panigrahi, and G. Panda, "Power quality analysis using S-transform; Power quality analysis using S-transform," *IEEE TRANSACTIONS ON POWER DELIVERY*, vol. 18, no. 2, 2003.

[36] D. Su, K. Li, and N. Shi, "Power quality disturbances recognition using modified s-transform based on optimally concentrated window with integration of renewable energy," *Sustainability (Switzerland)*, vol. 13, no. 17, 2021.

[37] Z. K. Peng, P. W. Tse, and F. L. Chu, "An improved Hilbert Huang transform and its application in vibration signal analysis," *Journal of Sound and Vibration*, vol. 286, pp. 187–205, aug 2005.

[38] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Snin, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[39] P. Flandrin, G. Rilling, and P. Gonçalvés, "Empirical mode decomposition as a filter bank," *IEEE Signal Processing Letters*, vol. 11, pp. 112–114, feb 2004.

[40] M. Feldman, "Hilbert transform in vibration analysis," *Mechanical Systems and Signal Processing*, vol. 25, pp. 735–802, apr 2011.

[41] M. A. Rodriguez, J. F. Sotomonte, J. Cifuentes, and M. Bueno-Lopez, "Classification of Power Quality Disturbances using Hilbert Huang Transform and a Multilayer Perceptron Neural Network Model," *SEST 2019 - 2nd International Conference on Smart Energy Systems and Technologies*, sep 2019.

[42] D. A. Pisner and D. M. Schnyer, "Support vector machine," *Machine Learning: Methods and Applications to Brain Disorders*, pp. 101–121, jan 2020.

[43] K. Manimala, K. Selvi, and R. Ahila, "Optimization techniques for improving power quality data mining using wavelet packet based support vector machine," *Neurocomputing*, vol. 77, pp. 36–47, feb 2012.

[44] L. Jorma and E. Oja, "IEEE Xplore Full-Text PDF:."

[45] J. Anishkumar, "View of Detection and Classification of Power Quality Abnormality Using S-Transform and KNN Classifier."

[46] S.-C. Wang, "Artificial Neural Network," *Interdisciplinary Computing in Java Programming*, pp. 81–100, 2003.

[47] M. Valtierra-Rodriguez, R. De Jesus Romero-Troncoso, R. A. Osornio-Rios, and A. Garcia-Perez, "Detection and classification of single and combined power quality disturbances using neural networks," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2473–2482, 2014.

[48] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, pp. 1735–1780, nov 1997.

[49] A. A. Abdelsalam, A. M. Hassanin, and H. M. Hasanien, “Categorisation of power quality problems using long short-term memory networks,” *IET Generation, Transmission and Distribution*, vol. 15, no. 10, pp. 1626–1639, 2021.

[50] S. K. Manikonda, J. Santhosh, S. P. K. Sreekala, S. Gangwani, and D. N. Gaonkar, “Power Quality Event Classification Using Long Short-Term Memory Networks,” *2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics, DISCOVER 2019 - Proceedings*, 2019.

[51] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” nov 2015.

[52] S. Wang and H. Chen, “A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network,” *Applied Energy*, vol. 235, pp. 1126–1140, feb 2019.

[53] J. Chung, C. Gulcehre, and K. Cho, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” tech. rep.

[54] R. Dey and F. M. Salemt, “Gate-variants of Gated Recurrent Unit (GRU) neural networks,” *Midwest Symposium on Circuits and Systems*, vol. 2017-Augus, pp. 1597–1600, sep 2017.

[55] X. Zu and K. Wei, “A simple gated recurrent network for detection of power quality disturbances,” *IET Generation, Transmission & Distribution*, vol. 15, pp. 751–761, feb 2021.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention Is All You Need,” *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 5999–6009, jun 2017.

[57] Y. Zhang, Y. Zhang, and X. Zhou, “Classification of power quality disturbances using visual attention mechanism and feed-forward neural network,” *Measurement*, vol. 188, p. 110390, jan 2022.

[58] N. Yadav and U. Binay, “Comparative Study of Object Detection Algorithms,” *International Research Journal of Engineering and Technology*, pp. 586–591, 2017.

[59] A. Raghunandan, P. Raghav, and H. V. Ravish Aradhya, *Object Detection Algorithms for Video Surveillance Applications; Object Detection Algorithms for Video Surveillance Applications*. 2018.

[60] G. Cheng and J. Han, “A survey on object detection in optical remote sensing images,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, jul 2016.

[61] R. Huang, J. Pedoeem, and C. Chen, *YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers; YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers.* 2018.

[62] P. Bharati and A. Pramanik, "Deep Learning Techniques R-CNN to Mask R-CNN: A Survey," *Advances in Intelligent Systems and Computing*, vol. 999, pp. 657–668, 2020.

[63] P. Doll, R. Girshick, and F. Ai, "Mask R-CNN ar,"

[64] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 779–788, 2016.

[65] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, feb 2019.

[66] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016.

[67] Z. Wang, S. Yang, M. Shi, and K. Qin, "FDA-SSD: Fast Depth-Assisted Single-Shot MultiBox Detector for 3D Tracking Based on Monocular Vision," 2022.

[68] F. Grasso, L. Paolucci, T. Bacci, G. Talluri, F. Cenghialta, E. D'Antuono, and S. de Giorgis, "Simulation Model and Experimental Setup for Power Quality Disturbances Methodologies Testing and Validation," *2019 IEEE 5th International forum on Research and Technology for Society and Industry (RTSI)*, pp. 359–363, 2019.

[69] R. H. Tan and V. K. Ramachandaramurthy, "Power Quality Issues in Distributed Generation - Google Libri."

[70] K. H. Kwan, P. L. So, and Y. C. Chu, "An output regulation-based unified power quality conditioner with Kalman filters," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4248–4262, 2012.

[71] S.-H. Cho, J.-U. Kim, I.-Y. Chung, and J.-H. Han, "Determination of Power-Quality Disturbances Using Teager Energy Operator and Kalman Filter Algorithms," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no. 1, pp. 42–46, 2012.

[72] P. N. Kumawat, D. K. Verma, and N. Zaveri, "Comparison between Wavelet Packet Transform and M-band Wavelet Packet Transform for Identification of

Power Quality Disturbances," *Power Research - A Journal of CPRI*, vol. 14, no. 1, pp. 37–45, 2019.

[73] Z. Liquan, G. Meijiao, and W. Lin, "Classification of multiple power quality disturbances based on the improved SVM," *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, vol. 2018-Janua, no. 4, pp. 2625–2628, 2018.

[74] M. Ijaz, M. Shafiullah, and M. A. Abido, "Classification of power quality disturbances using Wavelet Transform and Optimized ANN," *2015 18th International Conference on Intelligent System Application to Power Systems, ISAP 2015*, pp. 1–5, 2015.

[75] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Data augmentation using synthetic data for time series classification with deep residual networks," 2018.

[76] E. D. Cubuk, B. Zoph, D. Man, V. Vasudevan, and Q. V. Le, "AutoAugment : Learning Augmentation Strategies from Data," no. Section 3, pp. 113–123, 2012.

[77] N. Mohan, K. P. Soman, and R. Vinayakumar, "Deep power: Deep learning architectures for power quality disturbances classification," *Proceedings of 2017 IEEE International Conference on Technological Advancements in Power and Energy: Exploring Energy Solutions for an Intelligent Power Grid, TAP Energy 2017*, pp. 1–6, 2018.

[78] H. Xue, A. Chen, D. Zhang, and C. Zhang, "A Novel Deep Convolution Neural Network and Spectrogram Based Microgrid Power Quality Disturbances Classification Method," *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, vol. 2020-March, pp. 2303–2307, 2020.

[79] S. Aziz, M. U. Khan, Abdullah, A. Usman, and A. Mobeen, "Pattern Analysis for Classification of Power Quality Disturbances," *2020 International Conference on Emerging Trends in Smart Technologies, ICETST 2020*, 2020.

[80] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks," tech. rep.

[81] I. Goodfellow, Y. Bengio, A. Courville, and J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning," *Genetic Programming and Evolvable Machines*, vol. 19, p. 262035618, 2016.

[82] M. Buckland and F. Gey, "The relationship between Recall and Precision," *Journal of the American Society for Information Science*, vol. 45, no. 1, pp. 12–19, 1994.

[83] C. Iturrino-García, M. Bindi, F. Corti, A. Luchetta, F. Grasso, L. Paolucci, M. C. Piccirilli, and I. Aizenberg, "Power Quality Analysis Based on Machine Learning Methods for Low Voltage Electrical Distribution Lines," *SSRN Electronic Journal*, 2022.

[84] I. Aizenberg, *Complex-Valued Neural Networks with Multi-Valued Neurons Studies in Computational Intelligence , Volume 353*. Springer, 2011.

[85] I. Aizenberg, R. Belardi, M. Bindi, F. Grasso, S. Manetti, A. Luchetta, and M. C. Piccirilli, "Underground Medium Voltage Cables," 2021.

[86] I. Aizenberg, R. Belardi, M. Bindi, F. Grasso, S. Manetti, A. Luchetta, and M. C. Piccirilli, "A neural network classifier with multi-valued neurons for analog circuit fault diagnosis," *Electronics (Switzerland)*, vol. 10, no. 3, pp. 1–18, 2021.

[87] Z. Chen, Y. Q. Xu, H. Wang, and D. Guo, "Deep STFT-CNN for Spectrum Sensing in Cognitive Radio," *IEEE Communications Letters*, vol. 25, no. 3, pp. 864–868, 2021.

[88] J. Huang, B. Chen, B. Yao, and W. He, "ECG Arrhythmia Classification Using STFT-Based Spectrogram and Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 92871–92880, 2019.

[89] J. Nie, Y. Xiao, L. Huang, and F. Lv, "Time-Frequency Analysis and Target Recognition of HRRP Based on CN-LSGAN, STFT, and CNN," *Complexity*, vol. 2021, 2021.

[90] R. Ñanculef, P. Radeva, and S. Balocco, "Training Convolutional Nets to Detect Calcified Plaque in IVUS Sequences," *Intravascular Ultrasound: From Acquisition to Advanced Quantitative Analysis*, pp. 141–158, jan 2020.

[91] I. Aizenberg, A. Luchetta, and S. Manetti, "A modified learning algorithm for the multilayer neural network with multi-valued neurons based on the complex QR decomposition," *Soft Computing*, vol. 16, no. 4, pp. 563–575, 2012.

[92] I. Aizenberg, "MLMVN with soft margins learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 9, pp. 1632–1644, 2014.

[93] C. Iturrino-García, G. Patrizi, A. Bartolini, L. Ciani, S. Member, L. Paolucci, A. Luchetta, and F. Grasso, "An Innovative Single Shot Power Quality Disturbance Detector Algorithm," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, vol. 71, p. 2517210, 2022.

[94] J. Redmon, "Darknet: Open Source Neural Networks in C." \url{http://pjreddie.com/darknet/}.

[95] G. E. H. Alex Krizhevsky, Ilya Sutskever, "ImageNet Classification with Deep Convolutional Neural Networks."

[96] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, dec 2016.

[97] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, sep 2014.