FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Título da Dissertação

Nome do Autor

VERSÃO DE TRABALHO

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Nome do Orientador

15 de maio de 2023

Resumo

A proliferação de centrais fotovoltaicas de dimensão industrial levou à necessidade de métodos para detetar e classificar falhas nos seus componentes, sendo que estas que podem ter impactos económicos significativos. Neste trabalho, o estado da arte das ferramentas de deteção de falhas e estimação do estado aplicadas ao campo dos sistemas PV será explorado, com foco na compreensão do seu funcionamento, identificando-se pontos fortes e possíveis limitações. Nota-se que métodos baseados em aprendizagem computacional são os mais utilizados. Ainda assim, reconhece-se o contributo de diversos domínios para solucionar este tipo de problema, desde teoria dos grafos a processamento de sinal, aprendizagem profunda e aprendizagem quântica. Serão comparadas e propostas melhorias às abordagens existentes ou desenvolvida uma nova abordagem para abordar esta problemática. Com a inspeção das ferramentas mais bem sucedidas até à data e pela potencial oferta de uma nova abordagem, o objetivo deste trabalho é fornecer aos operadores de instalações fotovoltaicas um aumento na fiabilidade e eficiência dos seus sistemas. Além disso, há a esperança de que a tarefa desenvolvida possa ser futuramente generalisada para problemas de coesão de dados, impactando positivamente outros tipos de domínios orientados a dados.

Abstract

The increase in utility-scale photovoltaic power plants has led to the need for effective methods for detecting and classifying component faults, which can have significant economic impacts. This work assesses the current state of fault detection and state estimation tools in the field of PV systems, focusing on understanding how these tools work and identifying their strengths and limitations. It is seen that machine learning makes up the majority of state-of-the-art fault detection and classification algorithms. Still, many fields have contributed to this problem, from graph theory to signal processing, deep learning, and quantum machine learning. Consequently, this work compares and proposes improvements to existing approaches or a novel technique developed to address this issue. By examining the most successful tools to date and offering new solutions, the intention is to help PV plant operators improve the reliability and efficiency of their systems. The developed methodology is also expected to become a generalistic data cohesion algorithm, positively impacting other data-driven fields.

Agradecimentos

Aliquam id dui. Nulla facilisi. Nullam ligula nunc, viverra a, iaculis at, faucibus quis, sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur magna ligula, ornare luctus, aliquam non, aliquet at, tortor. Donec iaculis nulla sed eros. Sed felis. Nam lobortis libero. Pellentesque odio. Suspendisse potenti. Morbi imperdiet rhoncus magna. Morbi vestibulum interdum turpis. Pellentesque varius. Morbi nulla urna, euismod in, molestie ac, placerat in, orci.

Ut convallis. Suspendisse luctus pharetra sem. Sed sit amet mi in diam luctus suscipit. Nulla facilisi. Integer commodo, turpis et semper auctor, nisl ligula vestibulum erat, sed tempor lacus nibh at turpis. Quisque vestibulum pulvinar justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam sed tellus vel tortor hendrerit pulvinar. Phasellus eleifend, augue at mattis tincidunt, lorem lorem sodales arcu, id volutpat risus est id neque. Phasellus egestas ante. Nam porttitor justo sit amet urna. Suspendisse ligula nunc, mollis ac, elementum non, venenatis ut, mauris. Mauris augue risus, tempus scelerisque, rutrum quis, hendrerit at, nunc. Nulla posuere porta orci. Nulla dui.

Fusce gravida placerat sem. Aenean ipsum diam, pharetra vitae, ornare et, semper sit amet, nibh. Nam id tellus. Etiam ultrices. Praesent gravida. Aliquam nec sapien. Morbi sagittis vulputate dolor. Donec sapien lorem, laoreet egestas, pellentesque euismod, porta at, sapien. Integer vitae lacus id dui convallis blandit. Mauris non sem. Integer in velit eget lorem scelerisque vehicula. Etiam tincidunt turpis ac nunc. Pellentesque a justo. Mauris faucibus quam id eros. Cras pharetra. Fusce rutrum vulputate lorem. Cras pretium magna in nisl. Integer ornare dui non pede.

David Freire

"You should be glad that bridge fell down.
I was planning to build thirteen more to that same design"

Isambard Kingdom Brunel

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CXN	Cell Complex Neural Network
DC	Direct current
DL	Deep Learning
DNN	Deep Neural Network
LSTM	Long short-term memory
MCD	Minimum Covariance Determinant
ML	Machine Learning
PV	Photovoltaic
RBFNN	Radial basis function neural network
RMM	Recurrent Neural Network
SC	Short Circuit
SRC	Sparse Representation Classifier
STC	Standard Test Conditions
SVM	Support Vector Machine

Capítulo 1

Introduction

The XIX century marked a significant shift in the world's perception of energy resources as the desire to invest in renewable energy sources to power modern societies grew. This transition was driven by the need to reduce dependency on fossil fuels, mitigate the effects of global warming, and slow climate change. Renewable energy sources offer a range of benefits, including reduced greenhouse gas emissions, increased energy security, and air quality. Solar photovoltaic energy is a desirable renewable energy source due to its abundance, accessibility, and environmental benefits. While solar photovoltaic energy has proven to be both cost-efficient and environmentally friendly, it also comes with unprecedented challenges, such as its intermittent nature, low electrical inertia, complex forecasting, and geographic-dependent operating conditions. Despite these challenges, recent reports [**?**] show that the economic benefits of investing in renewable energy outweigh the complications, as there is an increasing global investment trend in these sources.

The general construction of PV farms, particularly on the utility-scale, has led to a need for effective maintenance and monitoring to ensure maximum efficiency and operational reliability. Towards this, various algorithms and routines are used to monitor the state of PV farms and identify any potential issues that may arise. Fault detection is crucial to this process, allowing PV farm operators to identify and address problems quickly. Detecting faults and identifying the necessary steps can prevent or minimize downtime and ensure optimal performance. Given the importance of maintaining high levels of operation, knowing if action is needed to restore or fix components from an anomalous scenario is desirable for reducing investment risk and maximizing profits.

Integrating intermittent energy resources into modern electric grids has led to stricter requirements for connecting such power systems to ensure safe grid operating conditions. As a result, companies that own or plan to build photovoltaic farms must comply with these requirements and have adequate power electronics and monitoring/control capabilities. Failure to meet these requirements can result in sanctions or fines for the responsible party, as well as potential impacts on system availability, asset value, and disturbance propagation to the grid. To minimize these risks and maximize the value of their assets, companies may opt to implement fault detection and state estimation tools. These tools allow for the early detection and resolution of potential issues and can prevent or minimize downtime. The need to create or improve existing fault detection and

state estimation tools, and the search for the most effective methodologies for addressing these issues, drive research in this field.

Having laid the basis for why there must be system behavior assessment in utility-scale PV plants, it is necessary to understand what business concepts are crucial to this field. In the course of this work, the presented topics will go over the following questions:

- What components mostly fail in photovoltaic power systems?
- What is the average frequency of faults?
- What fault detection/state estimation tools exist for photovoltaic power systems?
- What are the most successful ones?
- What's their structure? Are they mostly centralized or decentralized?
- What are their computational costs/efficiency?
- What is the expected magnitude of precision and confidence?
- Which key performance indicators can evaluate the success of these tools?
- What are their implementation difficulties?

With these questions uncovered, the main objective is to adapt or design a novel algorithm/approach to fault detection based on modern artificial intelligence solutions. However, this can be split into finer goals:

- Identify and study existing fault detection tools for photovoltaic power systems.
- Adapt or develop a new tool.
- Apply and test the new tool in real case study PV assets.
- Validate the developed methodologies by comparison to reference tools.

Before reviewing state-of-the-art fault detection tools, types of failures in photovoltaic systems need to be understood: find which components usually fail, which ones fail more often, and how often. For this, it is necessary to understand such components' physical and electrical properties and the modeling techniques used to characterize them. There will be an assessment of utility-scale power plants architecture through literature, alongside the detection objective of state-of-the-art fault detection tools applied in this field. Then, there shall be an extensive analysis and review of what tools have been designed and used in this field. In this step, critical evaluation of the literature is a must for understanding the tool's scope, ease of implementation, and understanding that the data sets available for this work are compatible. Having selected the most prominent ones, they're to be qualitatively and quantitatively compared to each other in their application context so that

the results allow objective evaluations. This process requires implementing these tools, following the guidelines in the respective article/book/report, verifying their metrics, and checking if the achieved results resemble the same as the literature suggests. It will require gathering data sets, which can either be artificially generated through simulation or provided by an enterprise that services photovoltaic plant owners.

There's a desire that, in the end, the developed work helps achieve an improved method for fault detection and state estimation in photovoltaic power systems, resulting in a production-ready software application agile enough to deploy for multiple PV assets. It's intended that the algorithm specializes in data cohesion as a means of anomaly inference, allowing asynchronous and self-healing data transfers between the considered components. Depending on the new algorithm's characteristics, it could result in an approach capable of generalization and application to other engineering systems, benefiting more than just PV systems. No matter the chosen methodology, fault detection will, in most cases, result in an economic benefit, catastrophe prevention, and safety increase.

Capítulo 2

Fault detection in Utility Scale Photovoltaic Plants

2.1 Utility-Scale Photovoltaic System's Architecture

Utility-scale photovoltaic (PV) power plants are large-scale systems connected to the electrical grid, having installed capacities ranging from kilowatts peak (kWp) to megawatts peak (MWp). These systems typically consist of many PV panels interconnected through power electronics to aggregate and inject power into the grid. The number and type of components in a PV power plant depend on the plant's scale and topology, with different configurations possible for large-scale applications, including central inverters, string inverters, and multi-string inverters [**?**]. The physical installation of PV modules can include solar tracking apparatuses, such as single and dual-axis trackers [**?**], which add to system complexity and change production behavior. Understanding the architecture and components of PV power plants is vital for designing, operating, and maintaining these systems, as it helps optimize their performance and reliability.

Figure **??** presents a typical utility-scale PV plant architecture using the central inverter (or possibly multi-string inverter) configuration. It is noticeable that many system components may fail in one or more ways, which is why monitoring and fault detection algorithms are essential to maintain state estimation. The main subsystems considered in this work are the following:

- Solar photovoltaic panels (with or without bypass diodes).
- Tracking mount.
- Electrical cabling.
- Inverter(s) (mostly with Max Power Point Trackers).
- AC Transformer(s).
- Protection components (circuit breakers, fuses, surge protectors, etc.)

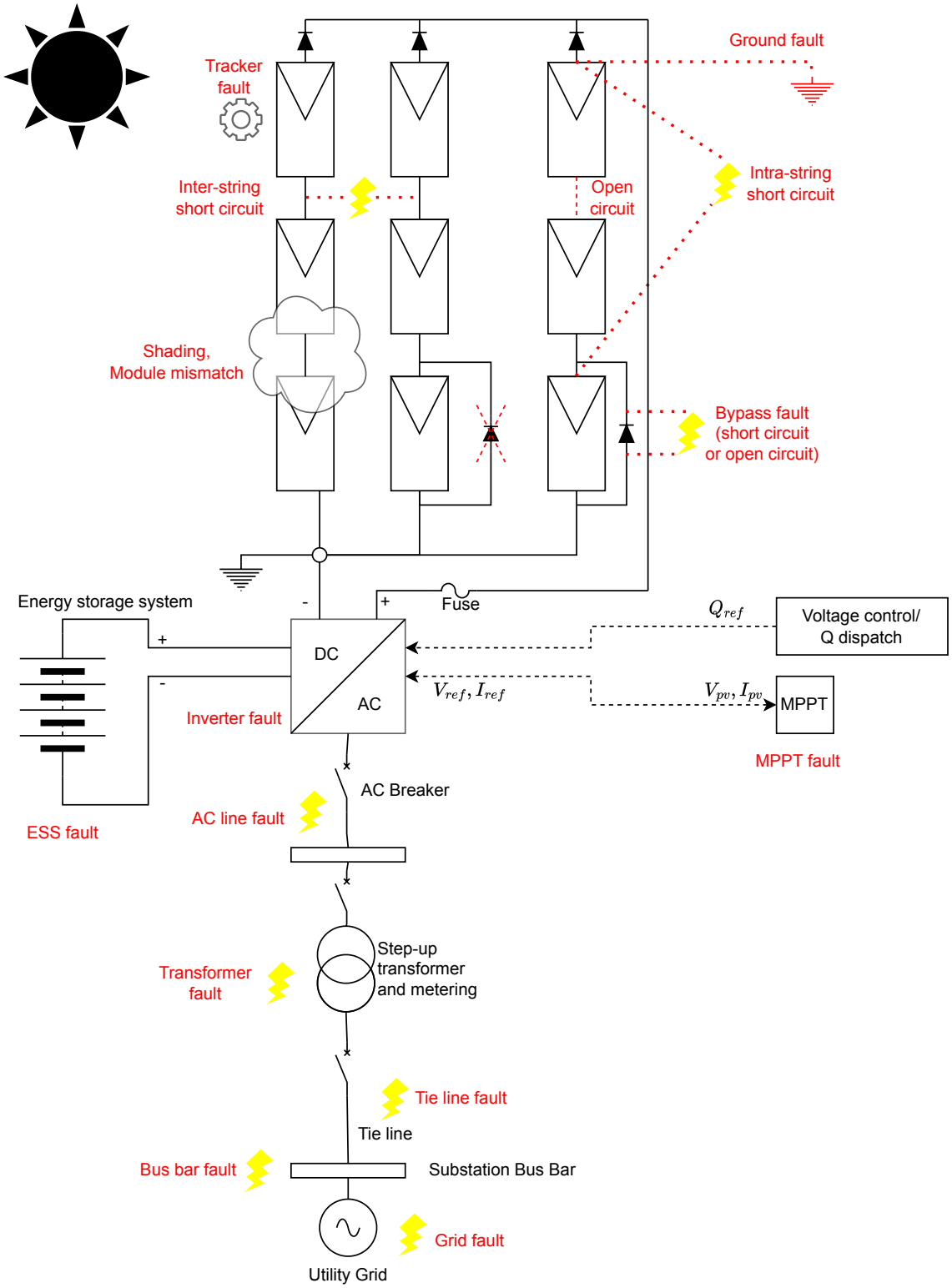

Figura 2.1: Representation of utility-scale PV plant components and some possible faults.

Most of these components have intrinsic variables, such as voltage and current values, that can help determine their operation states. Given that the utility grids (and the associated electricity

market) integrate large-scale PV assets, some of the before-mentioned components require constant monitoring and control, achieved with adequate embedded systems and sensor infrastructure [**?**]. Since monitoring utility-scale PV assets relies on the investment and technologies employed, engineers must consider data availability when developing data-driven algorithms. Thanks to the continuous advancements in communication technologies, namely in IoT (Internet Of Things), data acquisition is becoming faster, more reliable, and more precise. Not only is this fundamental for real-time asset assessment, but it also allows better training of fault detection algorithms. However, on the industrial scale (in the order of MWp production), having sensors embedded in every PV module comes with a high economic cost. Inverters are the components that usually possess monitoring capabilities, though the grid-tie connection should also be equipped with sensors. These can be considered the primary sources of information from utility-scale PV plants, with the most accurate, fast, and reliable data acquisition.

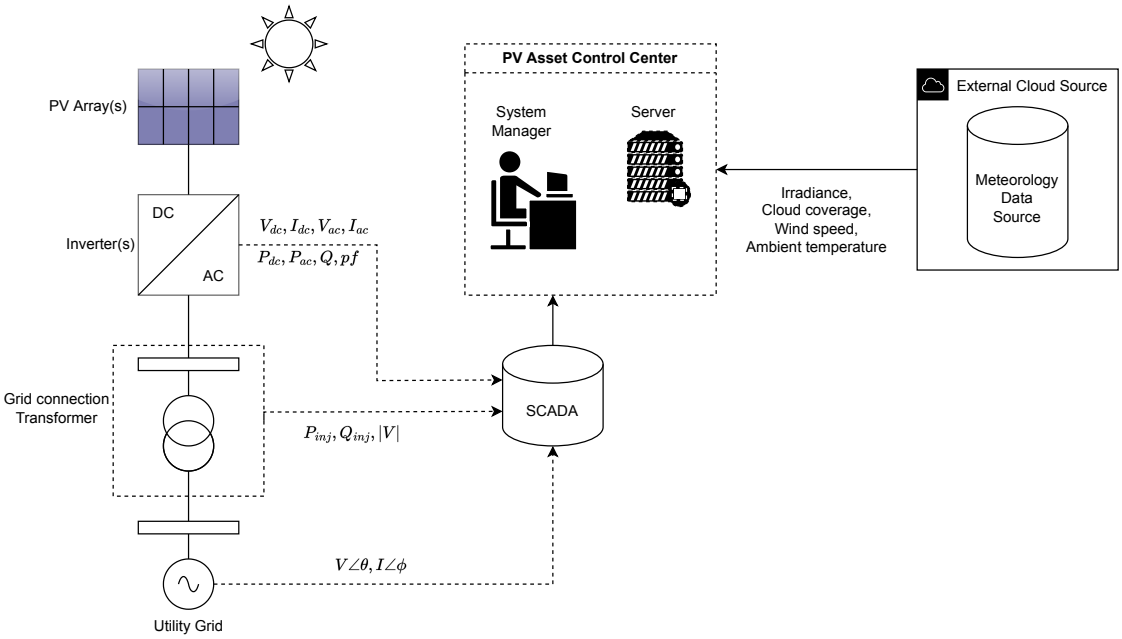

Figura 2.2: Typical data flow of utility-scale PV power plants.

Figure **??** represents a simplified data flow representation of a grid-tied PV system's most commonly available state variables, with most of them suggested by the IEC 61724 standard [**?**]. An external meteorological data source is defined since the PV system manager usually needs climate information for (at least) forecasting purposes.

2.2 Faults in Photovoltaic Systems

Several types of faults can occur in utility-scale photovoltaic (PV) power plants, which impact the performance and reliability of the system negatively. Unfortunately, some are very challenging

to detect and protect the electrical installation against, thus requiring sophisticated detection algorithms [**?**]. Besides the economical price, their occurrence may even cause safety hazards, such as fires [**?**], thus the urgency in early detecting or preventing such events.

According to [**?**], these faults can fit into three categories: electrical, mechanical, and environmental. Electrical faults include short circuits, open circuits, and inverter failure, affecting the PV panels' power output and the system's overall efficiency. Mechanical faults include broken panels, damaged cables, and defective inverters, which can lead to system downtime and reduced performance (although not mentioned, solar tracker failures could also belong in this category). Environmental faults include extreme weather events, such as hail or strong winds, which can damage the PV panels and other components [**?**].

The authors in [**?**] cover a comprehensive review of most types of faults studied in the ambit of detection and classification algorithms. However, authors in [**?**] have a more succinct fault categorization that better fits this work's scope. They categorize all the major PV system faults into either DC-side or AC-side. Figure **??** represents this detailed categorization with a tree-like structure.

Although also prone to failure, most literature on fault detection and classification for photovoltaic systems does not encompass solar tracking faults: most studies cover fixed PV systems. The supervision and assessment of these subsystems' correct functioning can be sensor-based [**?**] or image-based. Some authors developed fault detection methods for these apparatuses [**?**], using image processing on aerial photography to determine modules' slopes. This category of failures should be better supported when developing electrical data-driven algorithms since they can significantly affect the system's efficiency. Hence, this work will attempt to include said fault category in the proposed fault detection methodology.

Throughout the literature [**?**], some of the most noted faults in the context of fault detection are:

- Shading: partial coverage, temporary or not, of a PV array or module. It might result in a Hot Spot fault.
- Soiling: dirt accumulation, blocking sunlight from reaching PV Cells. It might also result in a Hot Spot fault.
- Short circuit: either line-line or line-ground.
- Open circuit: connection breakage between modules.
- DC arc fault: electricity plasma arc formed on broken connections.

According to a 2017 survey conducted on five utility-scale PV plants in Italy [**?**], the authors observed failure rates from <1% to 3% in the majority of plants and 81.8% in the worst scenario. The high failure rate of the latter had a demonstrated cause that originated from manufacturing mistakes: snail trails. Besides this phenomenon, hot spot faults and bypass diode faults/disconnections were among the most common.

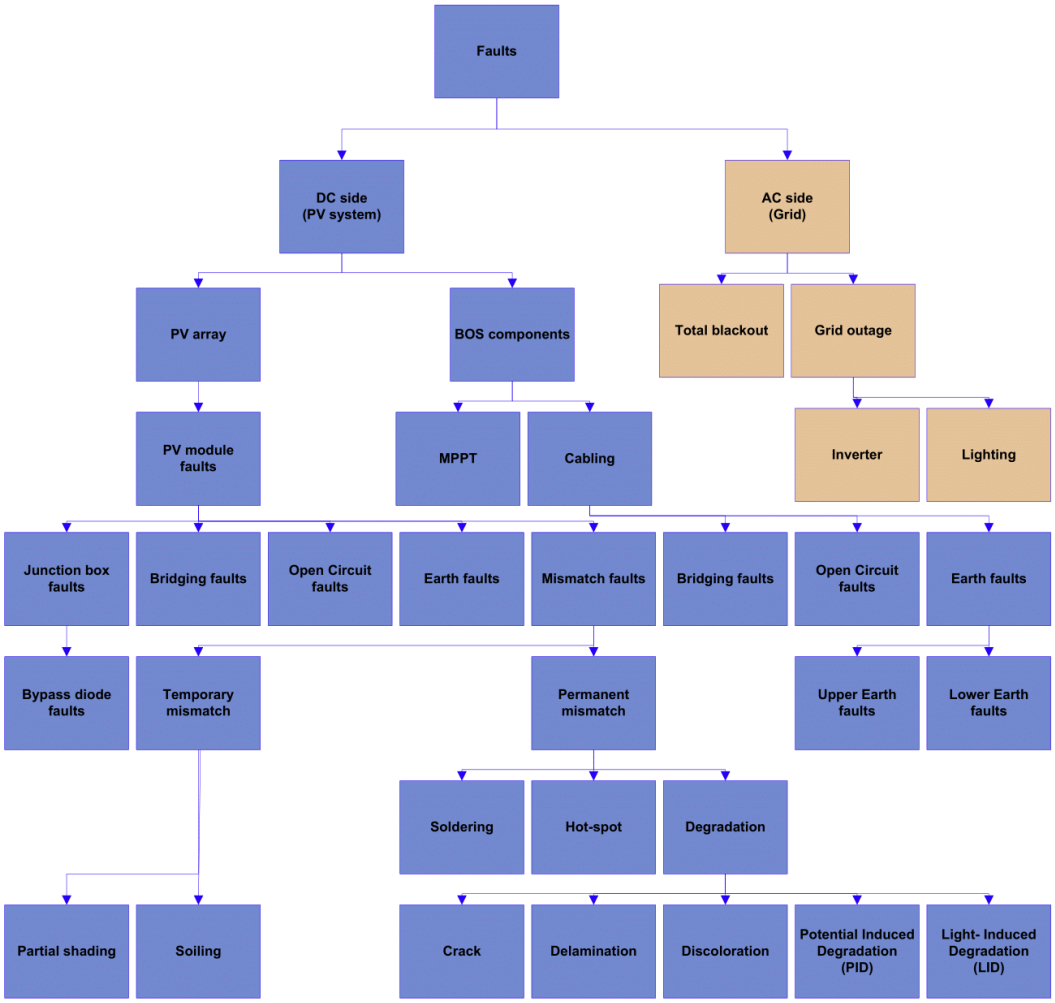

Figura 2.3: "Failures in grid-connected PV systems."
Image source and copyright: [**?**].

Alongside manufacturing failures, installation, planning, and other external effects can be the root cause for many of the presented faults [**?**].

Having the distribution of fault types from real-life scenarios is quite helpful for formulating fault detection algorithms. It allows for better generation/selection of training data and class decisions. In figure **??**, it is possible to observe the failure type distribution for 24.254 inspected modules. Soiling, shading, and mechanically related failures were not as prominent, with only a group share of around 6%. It is relevant to note that discoloration represents almost a quarter of all faults.

Although the study had a limited geographic scope, with only a few power plants diagnosed, it allows for a more realistic view of the common scenarios encountered in typical operational ground-mounted utility-scale PV power plants.

Due to the difficulty of classifying some of these faults, given their similarity on the consequent effect in the system, it will be seen in further sections that most fault detection algorithms only endeavor to classify between two to five types of reviewed faults.

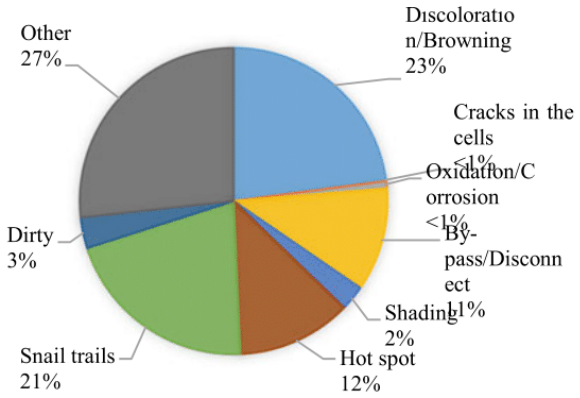

Figura 2.4: "Circle chart related to the module defects in the 5 plants (over the total number of failures)."

Image source and copyright: [**?**].

2.3 Modeling photovoltaic's physical/electrical behavior

Photovoltaic cells are the fundamental components of photovoltaic panels. They are made from semiconductor materials, such as silicon, and absorb photons that generate an electric current. Their electrical behavior is characterizable using the current-voltage (I-V) equation **??**. This equation, which represents a fundamental relationship governing the operation of PV cells, can be used to predict their performance under various operating conditions, such as differing solar irradiance and temperatures.

$$I = I_{ph} - I_d \times (e^{\frac{q\times(V_{pv}+I_{pv}\times R_s)}{n\times k\times T}} - 1) - \frac{V_{pv} + I_{pv} \times R_s}{R_p} \tag{2.1}$$

I_{ph} (A) is the light-generated current; I_0 (A) is the reverse saturation current; V_{pv} is the module's terminal voltage; I_{pv} is the module's output current; R_s (Ω) is the series resistance; R_p (Ω) is the shunt resistance; n (adimensional) is the diode ideality factor; k (J/K) is the Boltzman constant; T (K) is the cell temperature; q (C) is the electron charge;

For state estimation, it is crucial to accurately model PV modules' performance from the DC side of power converters. This information is vital for designing and optimizing PV power systems, as it enables predicting PV module performance under different conditions, as mentioned before. Accurate PV module models are also essential for state estimation and fault detection, as they provide critical information about the health and performance of PV modules, allowing for early identification of potential issues. In addition, they can be used to optimize the control and operation of PV power systems, improving their efficiency and reliability [**?**].

Physical and empirical models broadly categorize the several state-of-the-art methods for modeling photovoltaic modules [**?**]. Physical models lie on the fundamental physical principles governing PV modules' operation. They typically require detailed knowledge of the PV module's electrical and optical properties, such as its current-voltage (I-V) characteristics, spectral response, and temperature dependence. These models can accurately predict the PV module's performance

under a wide range of operating conditions, but they may be complex and computationally intensive to implement [**?**]. On the other hand, empirical models are based on experimental data and are typically more straightforward to implement. However, they may not be as accurate as physical models, especially under conditions that differ significantly from those used to generate the experimental data (usually STC) [**?**]. Some examples of state-of-the-art physical models for PV modules include the single-diode model (also known as the five-parameter model), and the two-diode model [**?**]. In contrast, one of the most used state-of-the-art empirical models is the Sandia model [**?**]. The choice of modeling method will depend on the specific application and the required level of accuracy and complexity; in some cases, there can be a combination of physical and empirical models.

Suppose the need arises to model PV modules in this work. In that case, it is critical to select a simple methodology so that the module's datasheet characteristics are sufficient to model the PV arrays accurately. In the case of utility-scale PV systems, detailed knowledge of the module's electrical and optical properties of empirical data may be limited, and building a model is only possible by recurring to the datasheet information. A complex model that requires more detailed information may not be feasible in such cases, and a simpler model that relies on fewer input parameters is more appropriate. The single-diode model seems appropriate for this use case, given the excellent trade-off between complexity and accuracy.

2.3.1 The five-parameter model

Figure **??** presents the single-diode model representation of the photovoltaic module. According to the five-parameter model, the unknown parameters are determined by fitting the model to experimental data or using data from the PV module's datasheet. The single-diode model can predict the PV module's performance under a wide range of operating conditions while maintaining reasonable accuracy. However, remembering that the single-diode model is a simplified representation of the PV module, it will have poor accuracy under certain situations compared to the more representative two-diode model [**?**].

2.4 Literature on Fault Detection and Classification for Photovoltaic Systems

The parent field of fault detection is anomaly detection (also known as outlier detection), a highly studied subject in the scope of statistics [**?**], applied in many scientific areas. Classification is also a well-studied subject in this field, with applications in numerous scientific contexts, from medical diagnosis to airport safety [**?**]. Consequently, adaptations of generic tools and ad hoc methodologies have originated to aid in solving fault detection and classification problems in photovoltaics.

According to [**?**], the tools dedicated to PV fault detection and state estimation mostly come from mathematical/statistical methodologies, machine learning, and deep learning applications.

Regarding the three general problem-solving principles mentioned before, it's known that machine learning and deep learning are the most popular and successful ones for recent applications that ought to solve complex problems. However, this categorization is somewhat limited, with contemporary literature suggesting an abundance of developed methodologies from different backgrounds, thoroughly reviewed in [**?**] and [**?**]. In [**?**], the authors consider two principal fault detection and classification algorithm branches: image-based and electrical-based; while [**?**] also distinguishes numerical-based techniques. Image-based refers to aerial or visual capture of the PV array by photography and thermal imaging, commonly used along with artificial intelligence algorithms for assessing the photovoltaic module's state. Although the contribution and importance of such methods are appreciable, this work will mainly focus on the electrical-based and numerical-based ones, as the use case of the developed tool is bound to this type of data.

Categorizing methodologies becomes fuzzy, considering that some literature mixes physical behavior models with machine learning, statistics, and signal processing. Figure **??** is an attempt to present a structure inspired by the review made by [**?**], [**?**], and this work's, with a focus on the more relevant techniques (for this work's scope). Hybrid models are ubiquitous since combining robust statistical, signal processing, ML, or DL models and PV's electrical characterization can achieve more remarkable results. Hence, a better representation than figure **??** would be an incomprehensible mesh of connections representing the permutations between category aggregation.

To not wander in the literature, there must be a decision on which methodologies to revise. The developed tool in this work must meet certain real-life constraints, such as data availability, frequency, accuracy, PV system configuration, and context. Therefore, the (qualitative) potential estimation for each methodology will be based on the capability of adapting the proposed algorithms to the same expected restrictions. This evaluation process confines the methodology review to emphasize the ones thought to be most capable of implementation in a real scenario. Therefore, the following sections will not cover an extensive literature review, as it is not intended to repeat the works of [**?**] and [**?**], only presenting interesting or adequate methodologies related to this work's scope.

2.4.1 Statistical and Signal Processing Algorithms

Statistical methodologies look into historical data to find the characteristics of how samples relate to the population (interpolation). These methodologies yield good results in case studies of PV farms that have been logging data for a considerable time, suffering in the cases that do not. Therefore, they are limited in that it is required to have curated data sets of historical significance for relevant features of the studied systems.

The literature on statistical and signal processing fault detection algorithms for PV is mostly quite dated ([**?**], [**?**], [**?**]), given that more recent machine learning methods have become increasingly attractive in this matter. Nonetheless, anomaly (or outlier) detection statistical algorithms can be used for fault detection in PV systems by identifying unusual patterns or deviations from normal behavior in the data collected from the PV system. Distance-based methods, such as the Euclidean, Mahalanobis, and MCD-based distances [**?**], may be adequate. Although simple,

these techniques might only work for detecting outliers in the context of PV systems if they are scale-invariant (due to the different magnitude in the system's state variables) and resilient to outlier contamination (which only MCD-based distance is capable of). In [**?**], the authors applied Analysis of Variance (ANOVA) and Kruskal-Wallis test for inverter failure detection, with the only downside of only being able to identify outliers in a sub-array resolution, i.e., not for specific string or module failures.

Some algorithms consider incoming data from PV systems as signals, allowing the adaptation of signal processing theory to develop ad hoc algorithms. Coming up with a relatively simple algorithm, the authors in [**?**] propose a power-based fault detection method that only requires delayed samples of the PV array's power output and a threshold. Its reasoning is that since the power output of PV systems can't vary beyond a given point, considering a very short-term period (milliseconds), significant perturbations in this variable can be associated with faults. Although the simplicity and ease of implementation, it's clear that the success of this method requires feeding the algorithm with relatively high-frequency data, which would only be feasible on-site (and with specialized monitoring equipment).

In [**?**], the authors successfully formulated a graph signal processing algorithm for fault classification that yields increasingly better results when there is a considerable amount of labeled data, although its training is only semi-supervised. The results outperformed other standard machine learning methods for the same training data, given 30% or more of labeled data. On another note, the data utilized came from the PVWatts [**?**] dataset, and the PV system is on a small scale (ASU testing facility [**?**]) possessing a monitoring density and capability that can be considered irrealistic for utility-scale. This same data source is present in many other reviewed works.

The authors in [**?**] displayed another excellent use for graph theory, although not specifically for fault detection: they implemented a consensus-based distributed approach to minimize the impact of noise in acquired data from the PV array. By formulating a data propagation algorithm that resulted in measurement convergence, they achieved higher accuracy for state estimation.

With both graph theory-based algorithm proposals, this field sparks interest in its usage for the upcoming formulated methodology, given that it would be desirable to achieve an algorithm that features fault detection alongside data consensus.

2.4.2 Machine Learning Algorithms

Machine learning is the trending way of solving increasingly complex and non-linear problems, as neural networks (or other learning structures) can better model complex, non-trivial, and nonlinear relations between data. Still, they are as good as the training data, with many designs requiring a lot of representative learning examples to achieve good results. Their output can also be very obfuscated (depending on the technique), meaning that many methods do not allow a direct interpretation of the relationship between inputs and outputs. This "black-box"characteristic, specifically of neural networks, is considered a disadvantage. Besides, extrapolating data remains a challenge when classically using these structures. Still, they have immense applications for PV

systems, from MPP (Max Power Point) estimation to power forecasting, soiling, and fault prediction.

In [**?**], an ANN is utilized to classify short circuit and hot spot faults. This algorithm achieved an outstanding 98.4% classification accuracy, yet the data was simulated in *MatLab/Simulink* and only considered two classes of faults. Because the inputs were the variation of voltage and current ($\frac{dV}{dt}, \frac{dI}{dt}$), the algorithm required data sampling with relatively high frequency (>5Hz). The present work will not regard such methodologies as background for the upcoming tool since requiring high-frequency simulated data while covering only two fault types is quite far from a real utility-scale PV system scenario.

The trend of utilizing simulated data (sometimes without even added noise) has been a target of criticism in [**?**]. Accordingly, this work also emphasizes that the literature shows many proposed ML (and other types of) techniques that fall into this concept, which makes selecting appropriate methodologies to base future work on a challenging task.

The proposed ANN solution in [**?**] is remarkable by the diversity of fault classification achieved: STC, short circuit, varying temperature, partial shading, complete shading, degraded modules, ground fault, and arc fault. It presents one of the most fault class coverage with high accuracy, considering the literature that utilizes synthetic noiseless data. Hence, the cyber-physical conceptualization and data preprocessing (clustering) demonstrated can be admired, but not forgetting that validation data came from a relatively unrealistic setting.

In [**?**], there is a captivating proposal of utilizing an autoencoder and pruned neural network to separate the tasks of detecting and classifying faults, which resulted in one of the most performant ML approaches in the literature. The algorithm classifies five states: degraded, shaded, soiled, short circuit, and STC, utilizing nine inputs representing voltage, current, power, and irradiance available from the MPPT, datasheet, or meteorological sources. While the neural network pruning adds complexity, it resulted in a better generalized and lighter-weight trained model suitable for faster detection times. Even though using data from a small-scale PV system, the presented algorithm and its assumptions may make it possible to adapt and implement in an industrial scenario.

On the note of performance, the work in [**?**] proposes a sparse representation classifier (SRC) that evaluates if the system has line-to-line or line-to-ground faults for varying operating conditions. Although a drop in accuracy occurred for extreme circumstances, it is impressive that the algorithm identifies faults in such varied operating conditions: 10 to 50 degrees ambient temperature, 200 to 1000 W/m^2 irradiance, 10 to 60 % of mismatch, and 0 to 25 Ω of fault resistance. The feature extraction step was also very impressive, which could be a determining factor in the method's performance. Unfortunately, this work also does not validate results with experimental data and only uses simulation as a source. However, the demonstrated computational performance, both in terms of training cost and utilization speed, its usage without the need for training for parameter tuning, the straightforward implementation, and consistent convergence, suggests the potential for this alternative in the face of other ML methodologies. The authors also emphasize that sparse representation might be utilized alongside different learning algorithms for classification, opening the door to many possible future implementations.

An exciting yet far-fetched proposal was made in [**?**], where a quantum neural network (QNN) is formulated for PV fault classification. The QNN was trained for predicting just two scenarios: faulty or standard, but required up to four days of training, resulting in 93.89% accuracy. For comparison, the classical ANN took twenty seconds to train and achieved 95.39% accuracy. Although the methodology showcases the potential of quantum computing for this field, its preliminary results still distance itself from the traditional methods.

An abundance of ML methods have been tested and reviewed in this field ([**?**],[**?**]), utilizing structures such as SVM, KNN, RF, etc. Nonetheless, the results of [**?**]-[**?**] sparked the most interest in this work's scope.

2.4.3 Deep Learning Algorithms

The field of deep learning is a branch of machine learning, with the term "deep"referring to amplified machine learning structures that ought to understand data patterns through more complex and intertwined artificial neuron connections. A simple example of a deep learning model would be the design of an artificial neural network with multiple hidden layers (DNN), with the intuition that each of these "extra"layers achieves feature/pattern recognition in a cascade. Other DL structures include the LSTM, CNN, and RBFNN. They have been explored alongside classical machine learning techniques for PV fault detection, although the known disadvantage is a usually high computational cost and relatively tricky implementation. These techniques are typically applied to image-based solutions [**?**] since they require classification based on 2D data from various image acquisition equipment [**?**], [**?**]. Given the 1D characteristic of raw electrical data, not much literature considers these techniques for fault detection, as it implies an extra step of increasing dimensionality. However, there are some promising results in doing so [**?**].

In [**?**], not only is a DL technique presented for fault detection and classification, but there is also the best attempt at comparative evaluation against other methodologies. As already mentioned, this work exposes that much of the literature presents results solely based on particular datasets comprising simulated noiseless data, which invalidates any significant quantitative comparison. Consequently, a CNN model based on the pre-trained AlexNet [**?**] is used both for classification or feature extraction, possibly allied with a classical ML model for classification in the latter. The classified faults were arc fault, partial shading, open circuit, and short circuit. While the experiments utilized simulation data, adding noise and an abundance of heterogenous operating conditions better resembled a real scenario. Considering the same noisy data, other tested methodologies present 22-70% average accuracies, with the proposed fine-tuned AlexNet CNN reaching a maximum of 70.45%. This work presents one of the best benchmarks in the literature, with decent coverage of other state-of-the-art ML and DL algorithms, while demonstrating the most realistic results and a sophisticated methodology proposal.

2.4.4 Proposed method's scope

While classical fault detection resides in the synchronous and direct evaluation of state and climate variables, realistic industrial scenarios can have data from various types, sources, and acquisition rates. It's also important to realize that monitoring equipment can register erroneous information, and current communication technology is also susceptible to delays and data loss. With this in mind, recent developments in the intelligent composition of deep learning structures aligned with graph theory spark some interest in their application to this field, such as the new deep learning technique named Cell Complex Neural Networks [**?**]. The motivation for choosing such a structure comes from its data propagation and consensus capability. The propagation techniques utilized in a CXN appeal to graph theory, dividing a system into other subsystems and components (nodes, also called cells in [**?**]) that share information. Even if the direct application of this structure might not be feasible or grant better results in the context of fault detection, its modification to meet the scope's needs could result in a robust and efficient solution. Further investigation of this state-of-the-art tool will unroll throughout the development of this work in an attempt to adapt this knowledge to the PV fault detection field.

According to the reviewed methodologies, the proposed tool should pertain to the DL or the hybrid category since, while having a central component of DL, it may also require modeling the PV system's components. The intention of proposing such a novel approach is to contribute to the deep learning methodology ecosystem, explicitly formulated for electrical-based PV fault detection and classification. As mentioned before, it aims at an asynchronous and online application, which differs from most current methods and presents a novel DL paradigm considering current knowledge. It is also desired that this work brings a comprehensive benchmark between popular methods (likewise [**?**]), utilizing a richer dataset with samples from tangible utility-scale PV assets, allowing accuracy assessment in a realistic scenario.

Tabela 2.1: Comparison of literature that inspired this work.

Reference and year	Data Source	Inputs	Proposed methodology	Classified Faults (alongside STC)	Validation data realism	Computational cost	Notes	Drawbacks
[?] 2020	Simulated PV System, added noise	Irradiance, Temperature, Short circuit current, Open circuit voltage, PV current, MPP current, MPP voltage, MPP power. Boost converter Maximum current, Voltage and power.	Pre-trained CNN (AlexNet) for feature extraction and classification	Arc fault, Partial shading, Fault during partial shading, Open circuit, Line to line SC	Moderate	High	Resilient against noisy data. Outperforms classical ML methodologies.	Requires data samples from the MPPT boost converter.
[?] 2020	Simulated PV System, no added noise	MPP voltage, MPP current, Short circuit current, Open circuit voltage, Irradiance	Sparse representation classifier	Line to line SC Line to ground SC	Low	Low	Very fast learning speed compared to classical ML structures. Straightforward implementation. Good feature extraction process.	Validation data was very idealistic. Only classifies line to line and line to ground faults.
[?] 2020	2*PVWatts dataset	MPP voltage, MPP current, Short circuit current, 2*Open circuit voltage, Irradiance, Fill factor, Temperature, Gamma ratio, Maximum power	Graph signal processing	2*Shading 2*Degraded modules Soiling Short circuit	2*High	Low	Semi-supervised, allows usage of unlabeled data for training. Better accuracy relatively to other ML methods for less labeled data. Low training cost.	-
[?] 2021			Autoencoder for detection and pruned neural network for classification			Medium	Separate the task of detection from classification, allowing for other combinations. Good performance method considering the algorithms complexity. Pruning creates an ANN less prone to overfitting.	Requires more complex training phase, for two different networks, and utilizing a dropout algorithm for pruning.

Table **??** represents a summary that compares four of the most inspiring reviewed proposals. These were chosen based on the authors' attention to detail and depth, traits not always found throughout the literature.

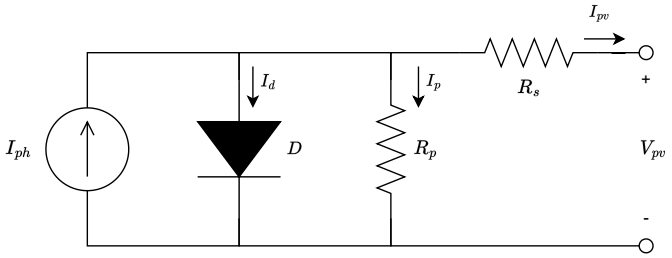

Figura 2.5: Single-diode model for photovoltaic modules.

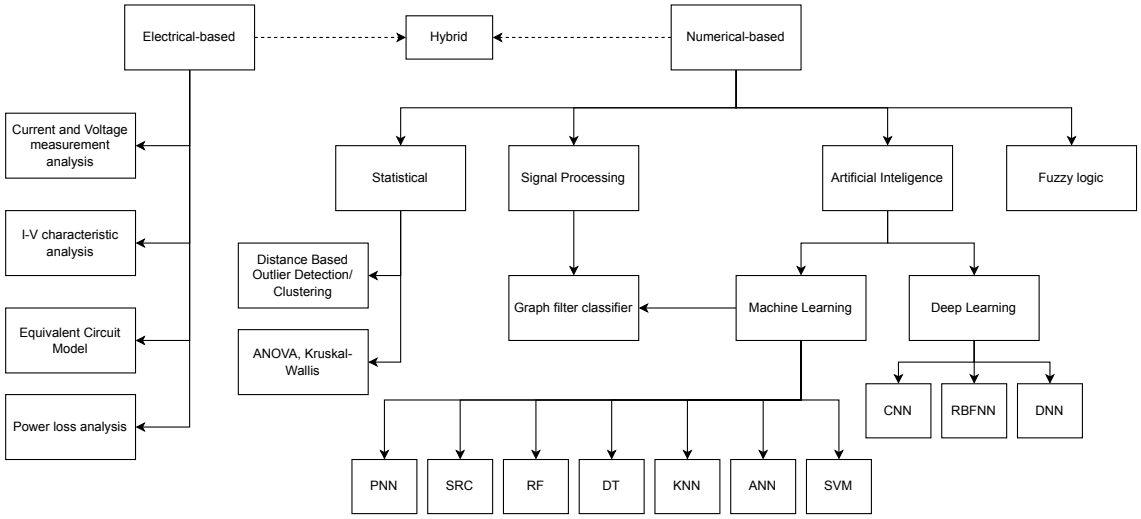

Figura 2.6: Representation of some of the methodologies employed in fault detection for PV systems.

Capítulo 3

Preliminary Work Plan

3.1 Academic and Industrial Setting

The present work unrolls at Faculdade de Engenharia da Universidade do Porto by a student employee of SmartWatt [**?**]. One of the employer's requirements is that the formulation of the final algorithm shall be done in the Python programming language [**?**]. Therefore, academic and industrial standards are considered during the development of the proposed tool.

Having the possibility of working with a company that provides projects for renewables, efficiency and artificial intelligence solutions for energy systems, there is the availability of PV asset data from various clients, mainly from the Iberian Peninsula and some other European countries. There will be a need to gather information from assets with historical significance and with the presence of faults. Although SmartWatt leases PV asset data for this work, it remains classified and will not be disclosed.

3.2 Work timeline

Figure **??** represents the expected chronological sequence of events. The most uncertainty comes from adapting the formulation of CXNs for PV fault detection and developing a Python application that implements its behavior, given that the know-how associated with this step is yet to be understood. There is also some doubt about the work time required to upscale the experimental environment toward actual deployment, knowing that the empirical setting of data science experiments may radically differ from software production environments. There will be an attempt to match the development environment with the implementation context, reducing the latter's risk.

According to the predefined workflow, the following chapters of this document will include:

- Deep analysis of Cell Complex Neural Networks and adaptation proposals towards PV fault detection and classification algorithms.
- The process of data gathering, defining data sources, and available information.

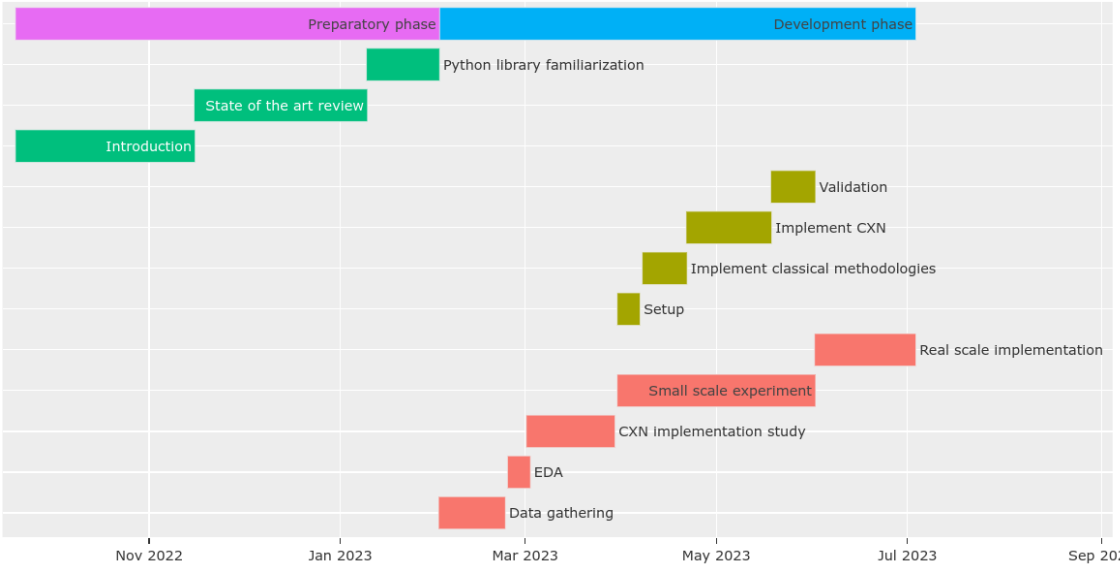

Figura 3.1: Dissertation work timeline represented by a Gantt chart.

- Exploratory data analysis (EDA) and data mining (feature extraction, clustering, etc.).
- Setup context for the small-scale experiment: dataset construction, cleaning, normalization pipelines, etc.
- Implementation of classical ML methodologies.
- Implementation of novel CXN methodology.
- Result validation.
- Productization of the developed algorithm.
- Large scale deployment of the application.

3.3 Development strategies

Data gathering will be accomplished by combining as many data sources for the same variables as possible, such as power meters and Scada measurements, local meteorological stations, and satellite data. Then, datasets will be organized into specific categories according to time resolution, acquisition method, availability, and accuracy. Geographical-based characterization might also be utilized to divide data even further. Data analysis, feature extraction, and clustering are essential in this step to harness valuable data from unlabeled samples. Complementary data from simulation might be considered to amplify the experimental dataset, with *Matlab/Simulink* as the preferred software for that purpose.

To implement classical ML methodologies, as well as the proposed tool, Python libraries such as TensorFlow [**?**], Scikit-learn [**?**], and SciPy [**?**] are the finest selection. They facilitate the creation of neural networks and other learning structures, possessing many different implementations

of scientific algorithms. The Mlflow [**?**] Python library will be used for experiment tracking. It features metric tracking, parameter tracking, and artifact storage, which eases the development of multiple ML models associated with a specific experiment. Other utilities discovered during the development process will be referenced along.

Software engineering guidelines will rule the software architecture of the final application, which should follow its well-studied organization patterns. Even though it is not a focus of this work, paying attention to code implementation should maximize the probability of expected application behavior, better scalability, and safety.

3.4 Final expectations

The final application aims to implement online fault detection for operational PV assets. For performance reasons, its deployment architecture is a consideration to take in prior, such as its potential for parallelization. It should be as generalized as possible, as that would increase its correct functioning for different PV assets of various plant operators. In the end, it is expected to resemble the usability and robustness of a finished software product so that PV plant operators can reliably utilize its outputs. Its success will result in higher PV system efficiencies and safety, advancing the world's energy transition.

Capítulo 4

CellTAN: Cellular Time Activation Networks

Distributed information systems characterized by time series data present various challenges, primarily due to their complex and dynamic nature. The sheer volume of data that must be processed and analyzed in real time is a significant challenge, leading to concerns over storage, computation, and scalability. Moreover, data quality issues such as missing or incomplete data and data heterogeneity arising from sourcing data from disparate sources with varying consistency and structure further exacerbate these challenges. Another critical challenge of these systems is handling the temporal aspects of time series data, requiring specialized pre-processing, feature extraction, and modeling techniques. The distributed nature of these systems further complexifies matters, with issues encompassing data synchronization and accuracy being a common concern. Furthermore, their implementation in real-world applications requires robust mechanisms for data security and privacy, which adds a layer of complexity to the design and implementation of these systems.

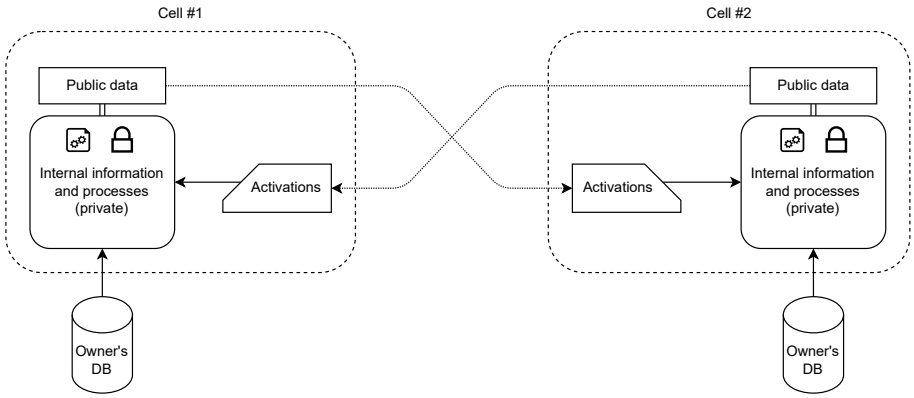

Figura 4.1: Simples CellTAN Network of two cells cooperating.

This chapter proposes a novel tool entitled CellTAN (Cellular Time Activation Network) that undertakes those challenges. CellTAN represents sparse yet interconnected components that func-

tion independently, cooperatively, and asynchronously. Inspired by other effective mechanisms like GNNs, CXNs, and Weighted Cross-Connection Networks (WCCNs), CellTAN uses a graph-like structure to represent a network of components, with nodes and connections. Following the introductory chapters, its primary purpose is to detect anomalous scenarios on PV systems. However, its generalized formulation introduces other valuable features which come naturally from fulfilling this goal. Such are state estimation, forecasting, and capturing the value in data from different PV asset owners without violating their privacy. For briefness' sake, we will the details about its benefits to be unfolded during the rest of this chapter.

Instead of tackling fault detection and classification in a classical centralized manner, which is already extensively showcased throughout the literature, this tool approaches this problem with a paradigm change: a distributed and asynchronous data coherence system. By having a virtual representation closely related to the physical form of sparse systems, relationships between components can be leveraged to assess their correct (or incorrect) operation. While initially designed for photovoltaic (PV) systems, the concepts of cells, connections, neighbors, time series, and uncertainty are universal and applicable to other fields such as biology, physics, and more. Thus, the potential for generic applicability sparks the interest to not bake specifics of PV systems directly into this tool, allowing its usage for other subjects. Figure **??** represents minimal scenario for a network: only two cells. It showcases some terms specific to this tool, that might be difficult to grasp at first, such as trust, events, and activations. Consequently, the glossary in **??** and detailed explanations throughout this chapter serve to clarify them.

Vast and scattered information across multiple agents is a common scenario faced by the industry of AI for energy systems, which cannot be aggregated due to privacy and confidentiality reasons. Nevertheless, its conjunction could have a lot of added value, given the similarity of certain assets: PV plants (as well as wind farms) from different owners in neighbouring geographical regions. This information-sharing potential for AI algorithms motivates the development of a mechanism that provides a way of communicating information between differently owned assets without any of the compromises above. However, and as stated before, data acquisition in PV scenario's is scarcely synchronous and might not occur in equal time resolutions for all the different components. The CellTAN addresses these issues with an instrument referred to as **Time activations**. It proposes a new way of communication that decouples from the needs of units, sampling rate, and synchronization, avoiding any resampling, normalization, or even obfuscation (to protect privacy). This mechanism is a core feature of the tool since it will be the means that will allow connecting different stakeholders' data, hence section **??** develops this matter thoroughly. Likewise, succeeding sections formulate the working of the **Cell** and its interactions within the network. Since it is the core component, understanding its behavior is crucial for a complete understanding of the tool.

4.1 Glossary

- **Knowledge base**: Refers to registered historical knowledge (samples) of time series variables.
- **Inputs**: Uniform fuzzy numbers (not necessarily, but it's the current choice) that represent one sample of the group of time series variables that define the state of a cell.
- **Outputs**: Similar to inputs, but obtained through mroe complex computations.
- **Time decay**: A process associated with increased uncertainty of variables over time.
- **Activations**: Timestamps of past ocurrences on a knowledge base that have a non zero value of membership against a set of recent inputs.
- **Events**: Occurences that the cell reports back to the hub, that in most cases are triggered by the exceeding of parametrized thresholds.
- **Trust**: A decimal number that represents how coherent two cells's activations are with each other. Besides its instantaneous computed value, it can also come from a cumulative computation on a defined time window.
- **Hub**: The central component of the cell network, which facilitates its visualization, management, and expansion. It acts as the proxy agent between the cell's communication.

4.2 The Cell

4.2.1 Principles

A cell is an independent entity composed of **data** and **processes**. The idea is to abstract fundamental system components (e.g. inverters, MPPT's) into this virtual entity. With an added intelligence layer and featuring a few different processes, it assesses its current state based on all available internal and external information, adding value to the existing data acquisition and monitoring systems. As an individual part of the system, it follows a set of rules that define its intrinsic and extrinsic behavior. These rules address data privacy, request boundaries, and real-world operational limitations.

Independence During operation, independence on neighbors or other network entities for continuous processing of outputs results in a more robust system and increases cell availability. Thus, given any connection cutoffs, the cell shall be unbothered by its surroundings and continue operating in an isolated state. Isolation is not preferred, but enduring it until outside contact is re-established avoids shutdown and startup procedures.

Selfish Computations The cell is selfish in that it will not perform any computations based on the request of others. This aspect creates a fundamental layer of protection against overloading the infrastructure in which it is deployed, which also increases availability.

Data accessibility Although selfish in computations, the cell shall provide access to select data valuable to the network. However, not all internal data is shared, and public data shall not compromise the cell's privacy (more on this later).

4.2.2 Processes and Data

The cell has a main process loop that executes a sequence of actions periodically:

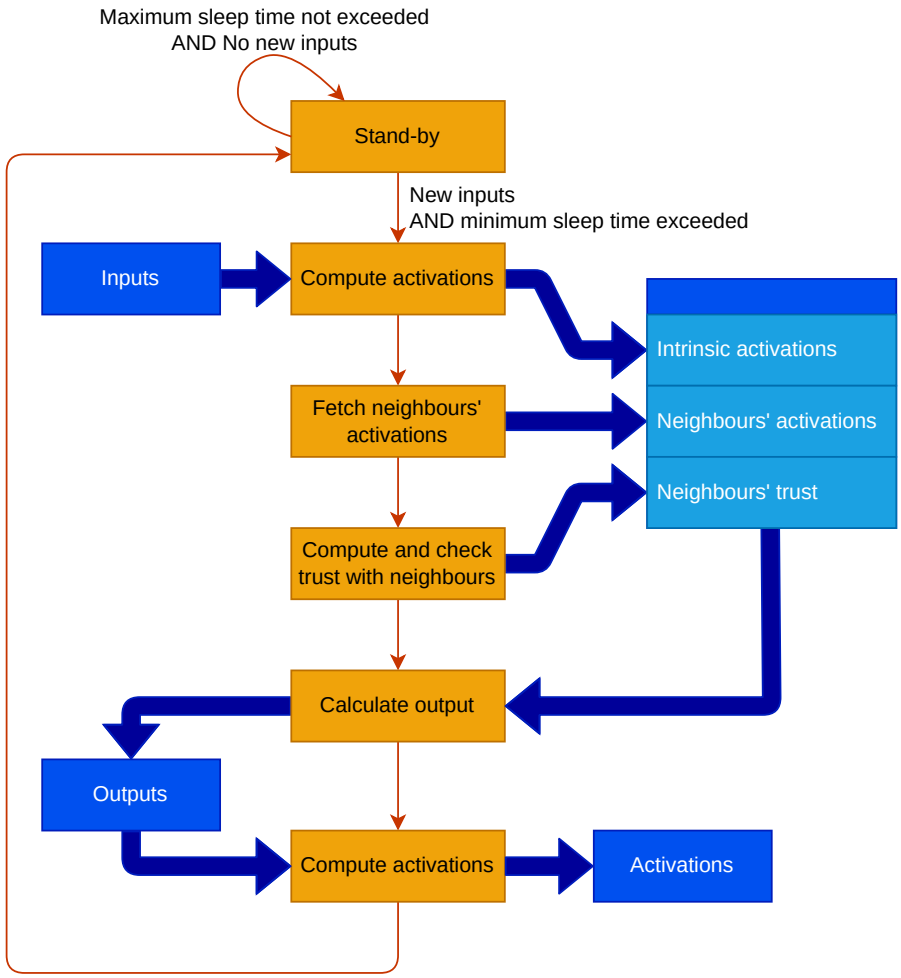

Figura 4.2: The cell's core sequence of actions (colored orange) and flow of data (colored blue).

Figure **??** showcases the core of the cell.

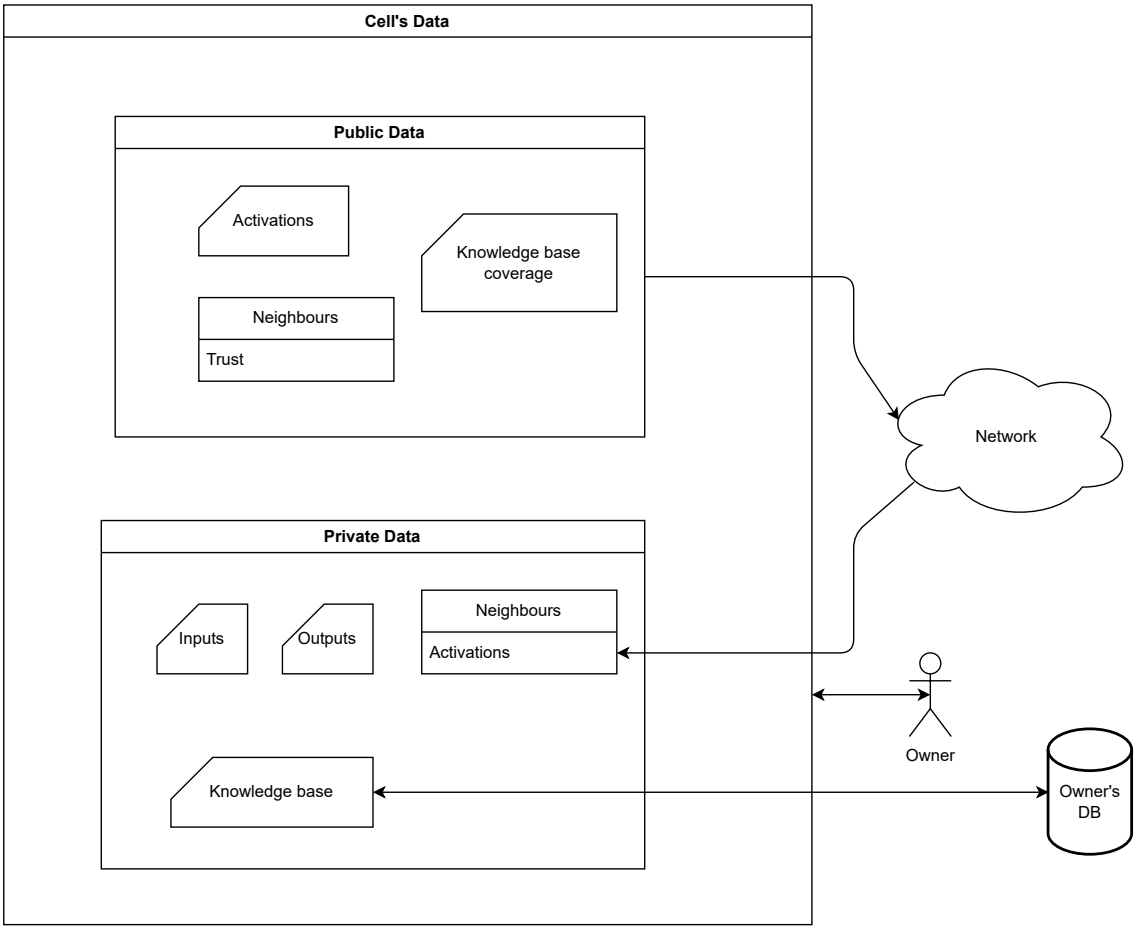

Figura 4.3: The cell's public and private data attributes.

4.2.3 Inputs and Outputs

The cell has inputs and outputs. Both are a view of the values that define its variables at a given timestamp, which is continuously rolling. While inputs are directly associated with raw sampled data from the system (injected by the owner), outputs are a byproduct of internal processes. The latter should present more accurate information since it is based on internal and external information (ideally) and be helpful for the cell's owner to assess its state.

Representing the cell's variables in a fuzzy (or probabilistic) manner can better capture the inherent uncertainty in time series data. For the cell's inner workings, we chose that inputs and outputs are not represented by crisp values but rather by classical sets. However, they are not limited to sets; fuzzy numbers or probability distributions could represent them also. Besides, they can be subject to a process called **time decay**, which ensures that the passage of time negatively affects uncertainty (more on section **??**). This mechanism increases the robustness of the cell by acknowledging the value of time in assessing its current state.

Summing up, the following may characterize inputs and outputs:

- Classical set: simple uncertainty band (e.g. uncertainty up, down, relative, etc);

- Fuzzy number: generalized fuzzy number representation [**?**] (e.g. triangular fuzzy number (a,b,d;h));
- Probability distribution: the distribution's characteristics (e.g. mean (μ) and standard deviation (δ) for Gaussian, mean rate of occurence (λ) for Poisson, etc.);

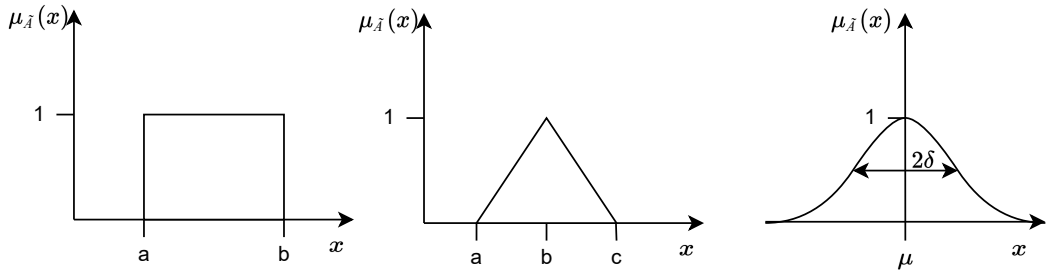

Figura 4.4: Classical set, triangular fuzzy number and gaussian distributions and the associated membership function.

The choice of using classical sets comes with some pros and cons: many operations become quite efficient, but we lose any notion . For example, filtering historical data becomes trivial: a value either lies within the bounds of the interval (membership value of one) or it does not (membership value of zero). This will be critical for some processes of the cell, primarily in temporal similarity extraction (**??**).

4.2.4 Time decay

In real-world dynamic systems that involve data acquisition, the certainty of the data collected tends to fade over time due to the nature of the data acquisition process. Typically, the most recent data is the most accurate representation of the system's current state, and as time elapses, the accuracy of previous data points decreases. This occurs naturally due to various factors such as environmental changes, equipment degradation, or other system's dynamics. As a result, it is crucial to consider the time dimension when analyzing dynamic systems and to develop methods that can account for the decay in data certainty over time.

As stated before, and towards considering the time dimension for the current state of a cell, we formulate a **time decay** method to ensure a more truthful and reliable assessment of the cell's current state. During the standby stage, this process ensures that the inputs and outputs suffer an increase in uncertainty, which is has different side effects depending on what represents them (classical set, fuzzy number or probability distribution).

Consider the following example of converting a crisp value and uncertainty to a classical set:

$$x = 5 \pm 1 \rightarrow [5-1, 5+1] = [4, 6]$$

To simulate the increase in uncertainty over time, we can apply the following formula to each bound:

$$lower = lower - (lower - minimum) \times \frac{age}{decay} \tag{4.1}$$

$$upper = upper + (maximum - upper) \times \frac{age}{decay} \tag{4.2}$$

The *age* variable refers to the time difference between the present time and the instant the value was created. For implementation, the unit considered is the second. The *decay* is a parametrized constant (same unit as *age*) that defines the time it takes for the set to expand into its limits when starting from the median. It should be chosen based on the characteristics of the variable. However, a good starting point is defining it as equal to the data acquisition period so that the set is entirely expanded until a new value is measured.

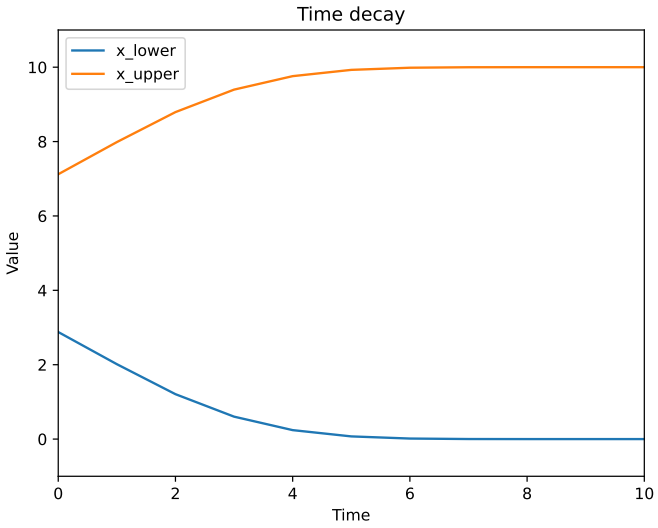

Figura 4.5: Visualization of the effect of time decay on x_{lower} and x_{upper}.

Table **??** and figure **??** represent the expansion of the set throughout a period equal to the decay parameter (ten seconds). When x reaches the age of ten seconds, its set represents complete uncertainty since its bounds became equal to the variable limits (x_{min} and x_{max}). We can see that the decreasing difference between the bounds and maximum/minimum values causes attenuation in the decay curve, as it displays a non-linear behavior. This behavior seems appropriate according to the reality of systems: as a variable becomes increasingly uncertain, there is less potential for its uncertainty to increase.

4.2.5 Temporal similarity extraction

Temporal similarity extraction the process of identifying recurring patterns in time series data. It involves the identification of past instances where the current state is observed to extract useful information about the system's behavior over time. By identifying historical periods with similar

states, temporal similarity extraction can help assess the current state or predict future trends. This technique is common in statistics, for purposes such as estimation and forecasting.

Using sets or probability distributions to filter historical data is one approach to simplify the process of temporal similarity extraction in multivariate time series data, while also making it more robust against noisy or incomplete data. This approach assigns membership values to each data point in the time series based on the corresponding set (or distribution) generated from the current cell inputs, and by eliminating samples past a certain threshold we can generate outputs. This is one of the reasons for deciding to represent the inputs and outputs of the cell in a fuzzy manner.

The proposal for similarity extraction in the cell consists of receiving new values (inputs) for the cell's variables from a data source (sensors or other data acquisition equipment, calculations, forecasting, etc.), generating a classical set, fuzzy number, or probability distribution from them, and then applying the bounds/membership function or probability density function to the knowledge base (see figure **??**). When historical samples are associated with membership values, there may be a process for determining outputs by combining data and membership values.

The initial choice is to use classical sets since filtering history becomes trivial and efficient: restrict the knowledge base to samples where all variables belong to the corresponding interval. Generating outputs with these can be as simple as constructing new sets based on the bounds of filtered knowledge (samples with membership equal to one). When filtering historical data with two or more variables, there is a trend of narrowing down the resulting data's space due to the intersection of constraints. Therefore, this process should result in sets that are an equal or better assessment of the present state (than the sets generated by inputs). However, filtering might also result in zero samples (membership value of zero on all knowledge base's rows), which indicates not having "memory"of any similar occurrence. This zero-sample filter is an excellent indicator for potentially anomalous scenarios, primarily if we know that the knowledge base is statistically representative of the variables in the cell.

This process also makes possible for simple forecasting. Considering an offset (arbitrary number of rows forward) in the activations, the temporal similarity extraction and computation of outputs results in future values. So, cells may either compute outputs related to the present or future. Nonetheless, this temporal shift might be difficult to achieve if data samples arrive at randomly spaced intervals of time, thus would only be straightforward for systems with time-consistent data acquisition.

The following examples consider that $t = 0$ is associated with the present, and μ represents membership functions.

Self Similarity Having a knowledge base and input variables, the cell can perform intrinsic temporal similarity extraction. Consider the a cell that is characterized by two variables, which are a function of time: $x(t)$ and $y(t)$. Assume that, at a given instant, these are the new cell inputs:

$$x(0) = 1.0 \rightarrow \mu_{x(0)}(x) = \{\, 1 \,, x \in [0.9, 1.1] 0, x \notin [0.9, 1.1] \tag{4.3}$$

$$y(0) = 2.0 \rightarrow \mu_{y(0)}(y) = \{\, 1 \,, y \in [1.8, 2.2] 0, y \notin [1.8, 2.2] \tag{4.4}$$

The membership functions μ are generated considering that x and y are characterized by uniform and symetrical uncertainty, and that the received samples of $x(0)$ and $y(0)$ represent its median.

timestamp	$x(t)$	$\mu_{x(0)}(t)$	$y(t)$	$\mu_{y(0)}(t)$
2023-01-01 00:00:00	0.80	0	1.90	1
2023-01-01 00:01:00	0.85	0	2.00	1
2023-01-01 00:02:00	0.90	1	2.10	1
2023-01-01 00:03:00	0.95	1	2.20	1
2023-01-01 00:04:00	1.00	1	2.10	0
2023-01-01 00:05:00	1.05	1	2.20	0

Bounds of filtered knowledge

Figura 4.6: Visualization of a self similarity extraction example.

With the knowledge base represented in figure **??**, the resulting activations will be 2023-01-01 00:02:00 and 2023-01-01 00:03:00. These are the timestamps of past instances where the cell's variables have values belonging to the set generated from new inputs. Now we can also infer that the actual values of x and y should reside in a set constrained by the filtered historical data ($x(t)$ and $y(t)$). Therefore, the outputs based on self similarity extraction are:

$$x'(0) \in [0.90, 0.95] \tag{4.5}$$

$$y'(0) \in [1.80, 2.20] \tag{4.6}$$

These results confirm that outputs with less uncertainty are achieved, while only depending on intrinsic processes and private data.

Mutual Similarity Extending similarity extraction to multiple cells is a relatively trivial process. If a cell has access to another's activations, at the same rolling timestamp, and for an historical window that intersects its knowledge base, it can use that information to refine the result of intrinsic temporal similarity extraction. Using sets, this can occur by joining intrinsic and extrinsic membership values by aggregation (e.g. sum, multiplication, average, etc). Although simple, this process has some tricky requirements, such has not allowing a time difference between the computation of membership values of different cells to avoid joining incoherent information, and

4.2.6 Connections and Trust

A network could not be formed without the existence of connections. Having neighbours grants the cells perform actions such as mutual similarity extraction. Besides, there can be certain indicators associates with these links, and to take advantage of sharing information with neighbours, there comes a need to have something we named **Trust**.

In order to distinguish cells, each one of them has a unique identifier, generated upon criation and independent from the cell's name. This serves to easily register cells and manage the network. Since interconnections require an agent to keep track of these id's and correctly direct traffic, we introduce a component entitles **Hub** for keeping track of registered cells in the network and their corresponding id's. By having a central component which provides global network visibility, connections are easier to form.

4.2.6.1 Trust computation

...

4.2.7 Events

...

4.3 The Hub

The CellTAN tool is supposed to be easily accessible for different assets and owners on any given location. However, due to privacy and security reasons, monitoring equipment and other smart devices (IoT, servers, etc.) present in PV plants and the owner's database's are usually protected from unwanted outside connections. These facts

4.4 Implementation

4.4.1 Programming and Infraestructure

The cell is materialized through a Python module. It was developed using a mix of the OOP (Object Oriented Programing) and FP (functional programing) paradigms and features a structure very familiar with all that has been discussed thus far. Using Python comes with the following advantages:

- Easy to read and write code, requiring less syntax for complex operations compared to low level languages;
- Extensive availability of libraries and tools;
- Deployment ease: doesn't have to compile, only needing an interpreter and dependencies to run;

- Big comunity, with lots of resources (such as documentation) publicly available online.

The infrastructure choice for running both the cells and the hub is Docker containers. It allows running the cell's program as a containerized application, with all the necessary dependencies installed virtually, while also adding a layer of isolation between it and the host machine (increased safety). This makes the "production"environment more predictable and stable, since it acts as a whole separate system built for the sole purpose of running the application, instead of relying in the host's OS (Operating System) and running bare-metal.

4.4.2 Cell configuration and deployment

Configuring cells can be done through a simple configuration file using the YAML protocol (one per cell). It should contain the definition of its variables, database credentials, hub credentials, and all other parameters.

Capítulo 5

CellTAN Application

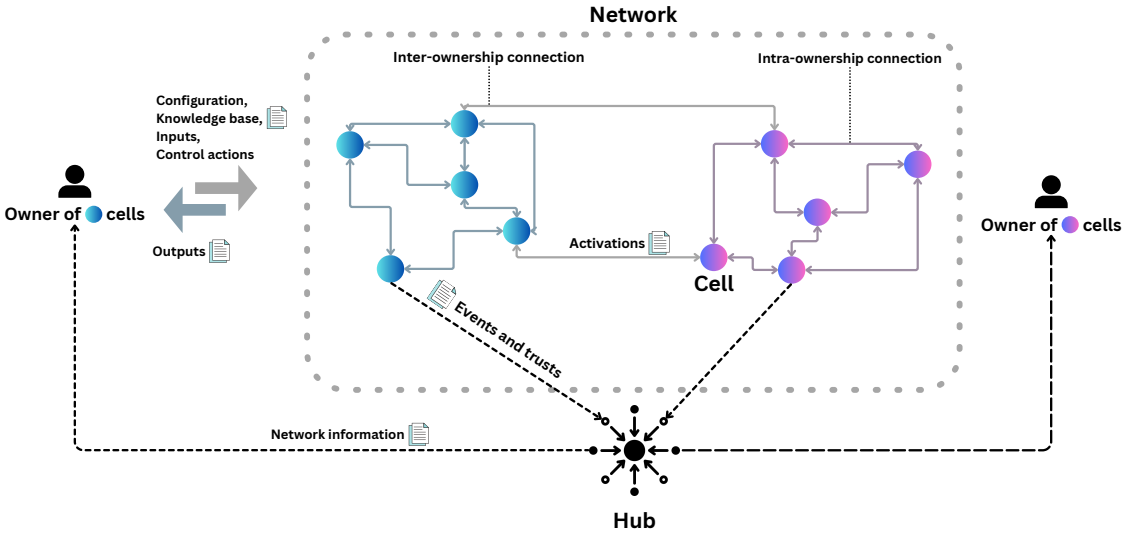

Figura 5.1: Ilustrative overview of a CellTAN.

...

Figure **??** illustrates a simple CellTAN network overview. Regarding the **Hub**, one might infer (correctly) that a central component breaks the non-centralized paradigm. Nevertheless, it is present to solve some real-life implementation challenges and limitations, as will be further discussed in this chapter.

...

Anexo A

Loren Ipsum

Depois das conclusões e antes das referências bibliográficas, apresenta-se neste anexo numerado o texto usado para preencher a dissertação.

A.1 …

…