Fault Detection in Transmission Lines: a Denial Constraint Approach

Nicolas Tamalu[1], Leandro Augusto Ensina[1,2], Eduardo Cunha de Almeida[1], Eduardo Henrique Monteiro Pena[2], Luiz Eduardo Soares de Oliveira[1]

[1]Department of Informatics (DInf) – Federal University of Paraná (UFPR)
Curitiba – PR – Brazil

[2]Federal University of Technology - Paraná (UTFPR)
Toledo, Campo Mourão – PR – Brazil

{nt18,eduardo,lesoliveira}@inf.ufpr.br,

{leandroa,eduardopena}@utfpr.edu.br

Abstract. *This paper introduces an approach for discovering denial constraints (DCs) to identify faults in transmission lines. However, the considerable volume of data in the studied scenario makes traditional DC discovery impractical due to lengthy execution times. We propose an alternative DC discovery approach that uses streaming windows to address this issue. Our experiments demonstrate that the DCs identified in pre-fault windows differ significantly from those in post-fault windows. This valuable insight enables us to detect faults autonomously, eliminating the need for human intervention (i.e., an unsupervised method). The experimental evaluation featuring diverse fault events reveals that our approach achieves fault detection with remarkable 100% accuracy.*

1. Introduction

Electrical power systems rely on transmission lines to deliver electrical power to customers. These lines usually travel long distances and are exposed to several elusive glitches and transient events that can disturb electrical power transmission. These events include, for example, storms and fire under the lines. Unfortunately, such disturbances cause faults that end up interrupting the electrical supply. Generally speaking, a fault can be defined as any abnormal condition in the components of a power system, such as an increase in current flow to one or more phases [Prasad et al. 2018, Furse et al. 2021].

Electrical substations are the interface between the transmission lines and the distribution grid. The substations contain a protection system to guarantee the stability of the network and minimize any possible damage caused by faults [Singh and Vishwakarma 2015]. The goal of the protection system is to accurately and quickly detect the fault and enable the repair and restoration of the faulty line as soon as possible [Aleem et al. 2015]. Such systems contain a protection relay or a digital fault recorder that samples line signals to produce a data stream with sample points. Each point represents information on the distribution of load into three phases (A, B, and C) for the current and voltage signals in the circuit. The sample points are stored in a relational database and can be queried to represent waveforms of changing currents. Figure 1 shows an example of electric current signal waveforms with the incidence of a fault near 0.05 seconds. As can be observed, a failure affects the normal operating condition of the power system, which contains balanced and symmetrical loads, e.g., similar amplitudes of the current signal among the phases during the pre-fault cycles.

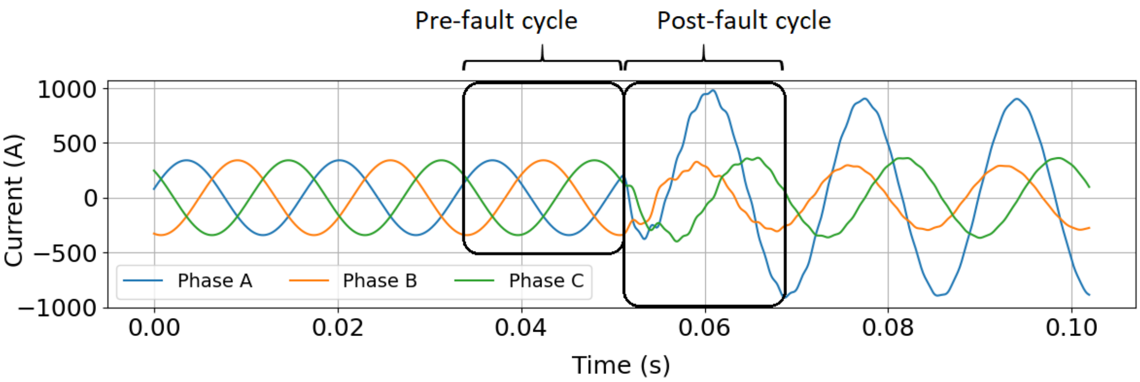

Figure 1. Waveforms of a faulty transmission line, involving the phase A and the ground (AG).

There are three essential tasks in fault diagnosis: detection, classification, and location. Of course, an effective classification and location rely on accurate fault detection. So it is possible to segment the oscillography into pre-fault and post-fault cycles, as represented in Figure 1. Such segmentation is required by several machine learning-based approaches found in the specialized literature. Thus, detecting the fault inception is crucial to allow the execution of the other two tasks since the segmentation, as mentioned above, is intrinsically dependent on this instant for a correct demarcation, which can directly affect the methods' performances for fault classification and location.

As highlighted by [Ferreira et al. 2016], some proposals for fault diagnosis assume that the detection phase is accomplished by the protection system itself (e.g., protection relays). Although such an approach works well for online applications, it is not valid for offline applications. It requires an approach to identify the failure inception since a fault record presents pre-fault and post-fault signals, as demonstrated in Figure 1. Developing an integrated tool including the three essential tasks in fault diagnosis is still a challenge. However, treating each task individually enables some advantages, such as high cohesion and independence among them, which can result in better generalization and adaptability capabilities. Due to these advantages, we investigate the fault detection task individually.

We present a novel unsupervised approach to detect faults based on denial constraints (DCs), acting as an offline application. Since DCs are ideal for representing the complex data business rules in databases [Chu et al. 2013], we use the formalism to represent the expected behavior of a transmission line so that it is possible to detect faults when there is a deviation from this behavior. For example, consider the following constraint, often found in transmission lines: "The electric current and voltage among the phases in a transmission line must be similar." A DC capturing such constraint, denoted as φ_1, can be expressed as follows: "There are no two records in the database where the electric current values differ between the phases A and B, and one of the records has a bigger phase A voltage." We defer the formal definition of DCs to Section 4.

Our method differs from traditional approaches based on machine learning algorithms by eliminating the need for feature extraction, as we directly utilize the raw data.

Additionally, our method is unique in that it is the first to detect faults based on an important data management technique, i.e., denial constraint. Thus, our approach enables the direct identification of fault inception from the database for the specific event of interest without loading the entire time series into dedicated systems.

The computational costs of DC discovery are directly influenced by the size of the dataset, specifically, the number of rows and columns it contains [Chu et al. 2013]. As a result, in our context, the sheer amount of data generated by the protection system makes traditional DC discovery impractical due to the long runtimes. In response, we contribute with an approach for discovering DCs in streaming windows. We hypothesize that DCs discovered in pre-fault windows greatly differ from those in post-fault windows and can be used to detect faults. A fault in a transmission line is detected by comparing the ratio of DC violations in each window to an expected threshold.

The main contributions of this work are summarised as follows:

- A DC discovery approach for streaming data;
- An application of DCs and DC discovery for fault detection in electric transmission lines;
- An unsupervised method for fault detection as an offline application;
- An empirical evaluation showing that our approach detects faults in transmission lines for all fault events assessed (100% of accuracy).

This paper is organized as follows. Section 2 describes the related work. Section 3 describes the database used in this study. Section 4 reports the proposed method for fault detection based on DC. Section 5 presents and discusses our results. Finally, Section 6 concludes this work.

2. Related Work

Several methods for fault detection can be found in the literature. Most are primarily based on machine learning. The authors in [Ferreira et al. 2020] used six neural networks for fault detection using voltage and current representation for a single transmission line terminal (bus). [Belagoune et al. 2021] proposed an LSTM-based method for fault diagnosis, including the detection task. [Coban and Tezcan 2021] used Discrete Wavelet Transform (DWT) on the measured single terminal current signals before fault detection, in which the three-level wavelet energy values obtained for each of the three-phase currents were used as input features for the detector based on the SVM algorithm. [Chen et al. 2018] presented a method for fault detection in power transmission lines based on a sparse convolutional autoencoder, automatically learning features from a dataset of voltage and current signals. In turn, [Asadi Majd et al. 2017] used the kNN algorithm with a sliding window with a length of a half cycle moved on the squared normalized current waveform of each phase to detect the fault inception.

On the other hand, [Gilbert and Morrison 1997] discussed about statistical approaches for fault detection, such as calculating the median using a sliding window over voltage or current oscillographies. Other approaches can be found in specialized surveys and reviews [Yadav and Dash 2014, Mishra and Ray 2018, Raza et al. 2020].

The machine learning methods discussed earlier rely on supervised learning, which presents a drawback since they depend on labeled data to train the algorithms. In contrast, our approach overcomes this limitation by operating without supervision, meaning it does not require labeled data. Consequently, it can detect faults autonomously. This paper introduces a novel approach based on denial constraints (DCs), typically employed to represent intricate data relationships in databases. The proposed approach employs DCs for fault detection without employing a traditional machine learning algorithm.

3. Fault Database

The Fault Analysis Database (FADb) is a public dataset with several fault simulations [Ensina et al. 2022]. These events were based on the IEEE 9-bus power system [Høidalen et al. 2019], which approximates a real power system. For this purpose, we used the ATPDraw and ATP (Alternative Transient Program) to model the system and run all simulations, providing representative time series of failures. These tools are widely used in the research community for electrical circuit studies, particularly in power systems considering the investigation about fault analysis and their effects.

We use a transmission line with the following properties: 500 kV, 414 km, and 60 Hz. In special, this specification represents the longest transmission line in a network of a public electric utility company in Brazil (Energy Company of Paraná – COPEL). The available data represent voltage and current signals for each of the three phases at both terminals for a sampling rate of 10 kHz. The oscillography of each simulation starts without failure, which occurs in distinct instants inserted into the same cycle. Also, the fault parameters used in the simulations are as follows:

- Type: AG, BG, CG, AB, AC, BC, ABG, ACG, BCG, ABC;
- Location: 1 to 100% of line extension, with intervals of 1%;
- Resistance: 0.01 to 200 Ω, with intervals of 10 Ω;
- Inception time, in seconds (s): 0.091 s, 0.093 s, 0.095 s, 0.097 s, 0.099 s, 0.101 s, 0.103 s, and 0.105 s.

In particular, the letters A, B, and C represent each of the three phases of a transmission line, while the letter G corresponds to the ground action in a fault. The combination of initials indicates faults involving multiple phases or the ground. For example, AG indicates a fault involving the phase A and the ground, as well as AB represents a fault between the phases A and B without the action of the ground. For more technical details, see the following references [Yadav and Dash 2014, Aleem et al. 2015, Grainger et al. 2016].

The FADb repository contains 168,000 fault events combining each of the previously mentioned parameters. The available archives ensure reproducibility of the results and the generation of new fault events considering other values of the parameters. All simulation data are available in our repository[1].

4. Denial Constraint Approach

The goal of a DC is to identify conflicting relationships of combinations of column values with sets of predicates. A DC specifies a conjunction of predicates that cannot be true for any pair of tuples. We use the formalism of a predicate, as p: $t.X\ \theta\ t'.Y$, where X, Y are columns of a table r with schema R; t, t' is a pair of distinct tuples of r; and

[1]https://1drv.ms/u/s!ArMEeMx4MYDNimHVxiDx3b4CI3iL?e=8GfXg7

$\theta \in \{=, \neq, <, \leq, >, \geq\}$ is a comparison operator, see [Chu et al. 2013] for additional DC definitions. We can specify a DC φ as follows:

$$\varphi\colon \forall t, t' \in r, \neg(p_1 \wedge \ldots \wedge p_m)$$

Using the above notation, we express our example rule, as $\varphi_1 : \forall t, t' \in r, \neg(t.IA <= t'.IA \wedge t.IB <= t'.IB \wedge t.VA >= t'.VA)$, where IA is the current at phase A, IB is the current at phase B and VA is the voltage at phase A.

Fault detection with the DC approach is divided into three parts: data streaming and processing windows (Section 4.1), discovery of DCs (Section 4.2) and fault detection using these DCs (Section 4.3).

4.1. Data Streaming and Processing Windows

We use data streaming notation to represent the signals of current and voltage converted by an oscilloscope [Braverman and Ostrovsky 2010]. A data stream is a finite sequence of n observations $T = \{x_0, x_1, \ldots, x_{n-1}\}$ read in an increasing order of the index i, as $0 \leq i < n$. A single data observation x_i contains information of electric current and voltage from the three phases of the transmission line at the i^{th} index.

The transmission line signal is humongous, generating gigabytes of data per second. For example, the dataset used in our experiments stores about 96 GB of data. We propose splitting the stream into small finite sets, called W_j windows, to facilitate DC discovery. We use two different window types: a fixed-size tumbling window and a sliding window. A tumbling window is a non-overlapping batch of the data stream, such as $W_y \cap W_k = \emptyset$ which enables DC discovery with small runtimes. A sliding window traverses a number of tumbling windows to detect faults using the discovered DCs. Sliding windows overlap data observations $W_y \cap W_k = \{x : x \in W_y \wedge x \in W_k\}$.

4.2. Denial Constraint Discovery

DC discovery is one of the most computationally expensive data profiling tasks, so care must be taken with such processes [Chu et al. 2013, Abedjan et al. 2015]. Instead of discovering the DCs using the entire dataset, we discover DCs in tumbling windows (batches with 1,000 samples). We use the state-of-the-art DC discovery algorithm, DCFinder, as provided in [Pena et al. 2019]. Our goal is to discover the DCs and capture the expected behavior of a transmission line without and with faults in pre-fault and post-fault windows, respectively. We selected a subset of fault events to use in discovering DCs. So, we used a subset of 1,000 simulations from FADb dataset (Section 3) for this purpose with the following diversified and representative fault parameters:

- <u>Fault type</u>: AG, BG, CG, AB, AC, BC, ABG, ACG, BCG, ABC;
- <u>Fault location</u>: 1%, 25%, 50%, 75%, and 100% of line extension;
- <u>Fault resistance</u>: 0.01 Ω, 50 Ω, 100 Ω,150 Ω, and 200 Ω;
- <u>Fault inception time</u>: 0.095 s, 0.097 s, 0.103 s, and 0.105 s.

4.3. Fault Detection

We use the term $(t, t') \not\models \varphi$ to denote that the pair of tuples t, t' violate the DC φ. Our fault detection mechanism is based on the degree of approximation shown in Equation 1, which measures the ratio of the number of tuple pairs violating a DC divided by all the tuple pairs in a batch [Chu et al. 2013].

$$g(\varphi, r) = \frac{|\{(t, t') \in r \mid (t, t') \not\models \varphi\}|}{|r| \cdot (|r| - 1)} \tag{1}$$

We calculate the degree of approximation of the batch samples with the DCs previously discovered for each sliding window in previous steps. Then, we compare the degree of approximation with a threshold value to determine the occurrence or not of a fault. A fault is detected if the degree of approximation of a window is higher than the threshold value. In this scenario, there are two parameters to be considered: the window length and the threshold of the degree of approximation. The values evaluated in this work are presented as follows:

- Window length (w): 50, 100, 200, 400, and 800;
- Degree of approximation threshold (DAT): 0, 1x10^{-5}, 1x10^{-4}, 5x10^{-4}, 1x10^{-3}, 5x10^{-3}, 1x10^{-2}, 2x10^{-2}, 4x10^{-2}, and 6x10^{-2}.

To measure the effectiveness of our approach, we use the precision (Equation 2) and F1-Score (Equation 3) as evaluation metrics. We calculated each of these metrics in every window of every simulation.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$F1 - Score = \frac{TP}{TP + \frac{FN+FP}{2}} \tag{3}$$

In the above equations, TP, FP, and FN represent, respectively, the values of True Positive, False Positive, and False Negative. Figure 2 depicts an overview of the approach. We observed the sliding window depicted in blue color when running across the batches without faults in white color and purple color when running across the batches with faults in orange color. We also observe the degree of approximation and the evaluation by batch with DAT of 0.2.

5. Results and Discussion

We present in this section the results for the discovery of denial constraints and evaluation of the performance of these DCs for fault detection, acting as an offline application.

5.1. Discovery of Denial Constraints

During the initial phase, we executed DCFinder on the simulations without employing batching or partitioning techniques. This led to the identification of 140,879 unique DCs, with an average of approximately 1,300 DCs per simulation. However, the substantial

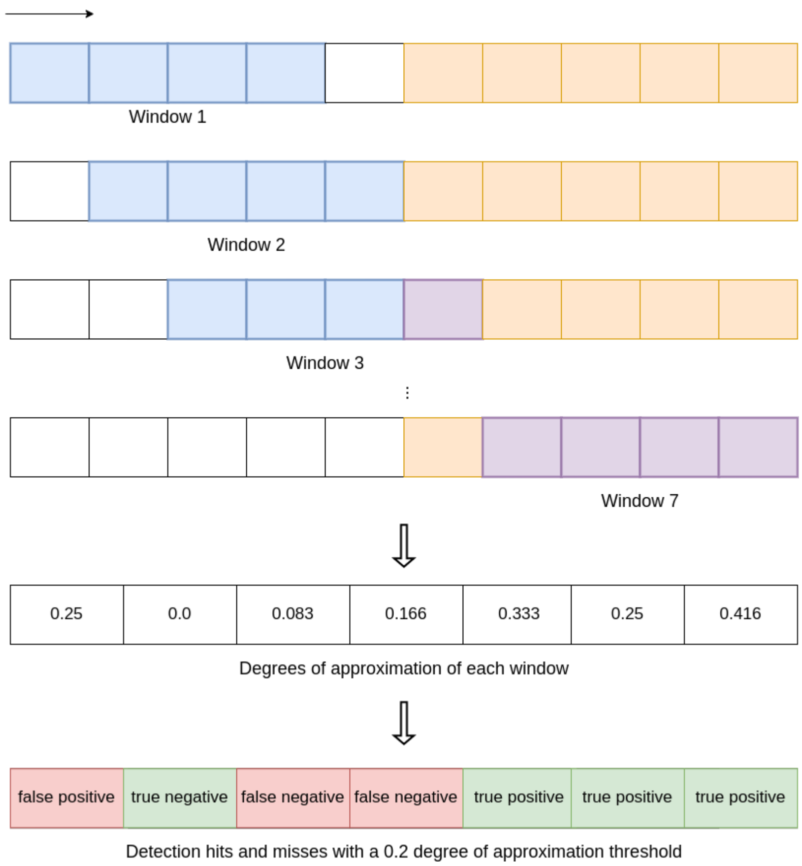

Figure 2. Fault detection with sliding windows and DC violations by batch.

number of DCs posed a challenge as it rendered subsequent stages of the process, such as obtaining coverage metrics and evaluating the constraints, impractical due to the extensive time required. Moreover, without data batching, it was not feasible to differentiate constraints discovered in fault-free samples, which are crucial for fault detection.

By batching selected simulations in groups of 1,000 samples, 153,052 unique DCs were discovered, resulting in an average of around 555 DCs per batch. This high number of discovered DCs allowed for the distinction between constraints present in samples without faults and those with faults, overcoming previous limitations.

Finally, the application of batching and partitioning (use of voltage and current samples only from a single terminal of the transmission line) reverberates in a drastic reduction in the number of DCs, finding 695 unique DCs and an average of approximately 18 DCs per batch, which demonstrates the relevance of the number of attributes, as it determines the size of the predicate space. Even more important, the partitioning process enables the discovery of constraints with predicates with attributes of a single bus. So, fault detection with these DCs does not require synchronization of data collected on both transmission line terminals, which is more consistent with a real operating environment.

From the 695 constraints found, 100 were found in batches with samples without fault presence, and 694 were found in batches with failure samples, i.e., only one DC was found exclusively in batches without failure, while the other 99 constraints of the batches without fault also appeared in batches with failure samples. This proportion in the number of times a DC was found in batches with and without fault varied a lot; for example, we obtained a constraint that was found in all 500 batches without fault and which was also found in only 62 batches with failure, since another DC that was found in all batches with no failure was also found in 3,386 batches with fault. The mean coverage of each of the 100 DCs found in batches without fault also varied. Trivial constraints with only one predicate obtained an average coverage of 1.0, other trivial ones with only two predicates had an average coverage of approximately 0.5, while non-trivial constraints varied between approximately 0.666 and 0.833.

The lengths of these 100 constraints ranged between two and six predicates, being the vast majority (79%) with five or six predicates. Disregarding the most trivial constraints (with one and two predicates), the shortest length was four; thus, the majority succinctness was 1×10^5, 0.666, and 0.833. Analyzing which types of dependency the constraints found represented, we noticed that we obtained a large amount of bidirectional order dependencies, such as: $\varphi_e : \forall t_x, t_y \in r, \neg(t_x.IC <= t_y.IC \wedge t_x.IA <= t_y.IA \wedge t_x.IB <= t_y.IB)$

which is coherent since the three phases have symmetric and balanced current values during the normal operating state. So, the ordering by the current of one of the phases must also order the values of another phase, even if in a different direction (ascending or descending). We also had unique combinations of trivial columns: $\varphi_f : \forall t_x, t_y \in r, \neg(t_x.VA = t_y.VA \wedge t_x.IC = t_y.IC)$

In turn, other dependencies did not fit any known definition, but presented a pattern where there is a predicate with the equality operator and two others with the inequality operators, such as: $\varphi_g : \forall t_x, t_y \in r, \neg(t_x.IC <= t_y.IC \wedge t_x.VA <= t_y.VA \wedge t_x.IA = t_y.IA)$. The complete set with all DCs found is available as a supplementary material[2].

5.2. Performance of the Proposed Approach

We used different simulations for the discovery of DCs and fault detection. The performance of our approach, considering the F1-Score measure, is shown in Figure 3. The performance of our approach increases for higher values of the window length together with lower values of the degree of approximation threshold. This suggests that DCs discovered at larger window sizes better capture the behavior of a transmission line and perform better in detecting DC violations.

We did not present performance results for window sizes larger than 800 due to the increase in the computational cost for larger windows, at the same time that there is no significant performance gain compared to $w = 800$. We did some additional experiments and observed that no performance gain was achieved for these cases. On the other hand, smaller window sizes demonstrate worse results, which would be even lower for $w < 50$. It is also possible to observe in Figure 3 that values higher than 6×10^{-2} for the DAT indicate a considerable performance loss for all combinations with window sizes.

In general, the pair of parameters composed of $w = 800$ and $DAT = 0$ initially seems to be the best configuration. However, if we analyze the precision measure (Figure 4), we identify that this pair is the only one that presented FP suggesting possible overfitting. The occurrence of FPs is a problem in the fault detection context since it in-

[2]https://github.com/leandroensina/FaultDetection_DC_SBBD

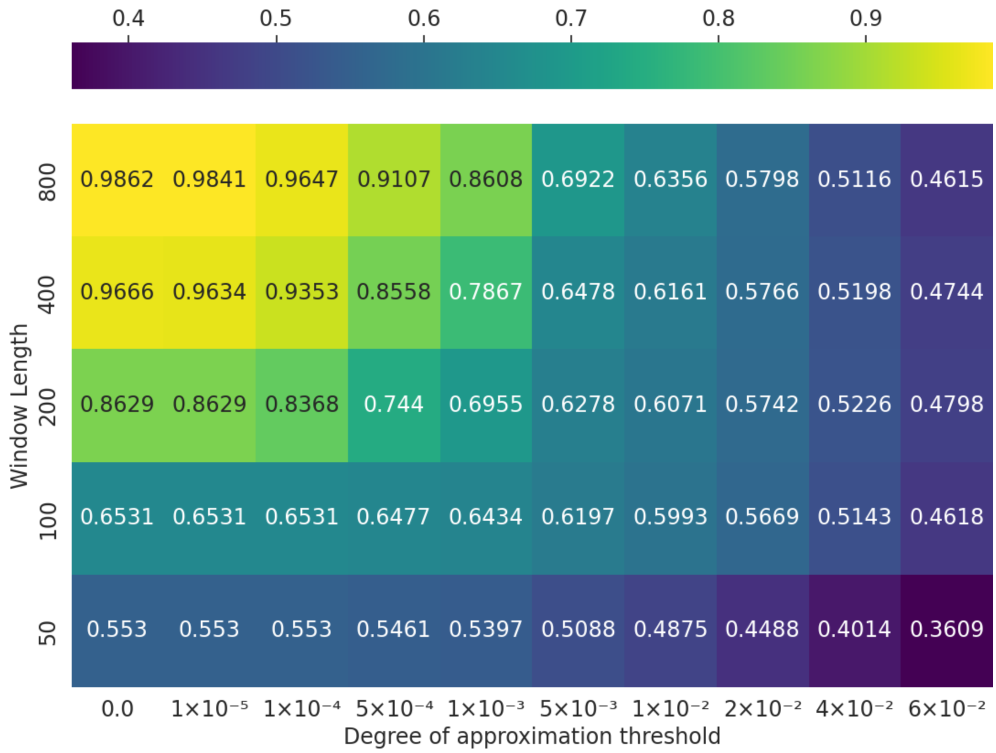

Figure 3. F1-Score for fault detection for each pair of parameters windows length and degree of approximation.

dicates that a window without electric fault data contains an anomaly. Thus, the absence of FP is essential to guarantee the accuracy of data for posterior fault classification and location tasks by machine learning approaches. This can impact in a performance loss for fault classification and location tasks, which these activities are directly related to the instant of the fault inception, determined by the fault detection task in an oscillography. Thus, it is essential the absence of FP in order to the data segmentation for a posterior fault classification and location be as accurate as possible.

In response, we consider $w = 800$ and $DAT = 1 \times 10^{-5}$ as the best pair of parameters for our approach. This pair of parameters does not present any FP and holds an F1-Score of 98.41%. It is crucial to mention that our method identified the fault incidence for all test cases (accuracy of 100%), but for some events our approach did not recognize the initial batch that contained the first failure samples as faulty, which penalized the performance and justifies the F1-Score of 98.41%. Thus, our method detected the fault incidence for all test events, but not necessarily in the first batch that the failure began.

Also, this pair of parameters requires only about 20 samples to determine the fault inception after its real beginning, as can be observed in Figure 5, where the smaller the value, the better the performance. We discovered 695 DCs in all the simulation data, where 100 were discovered in pre-fault windows and 694 in post-fault windows. DCs discovered in pre-fault windows significantly differ from those in post-fault windows and perform best in discovering faults.

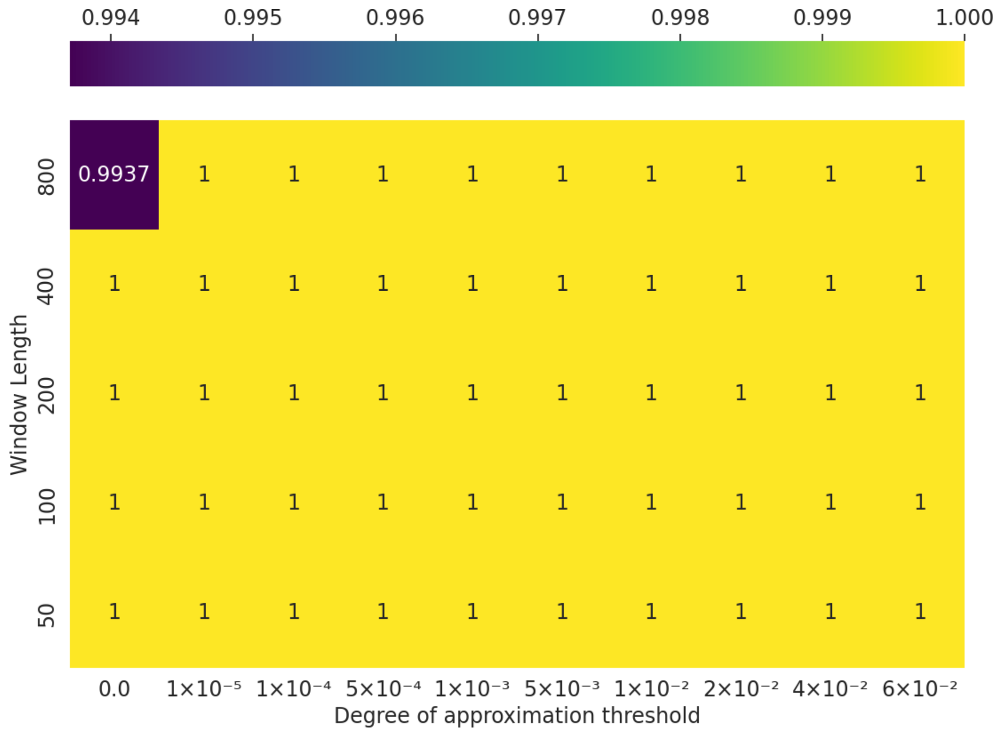

Figure 4. Precision for fault detection for each pair of parameters windows length and degree of approximation.

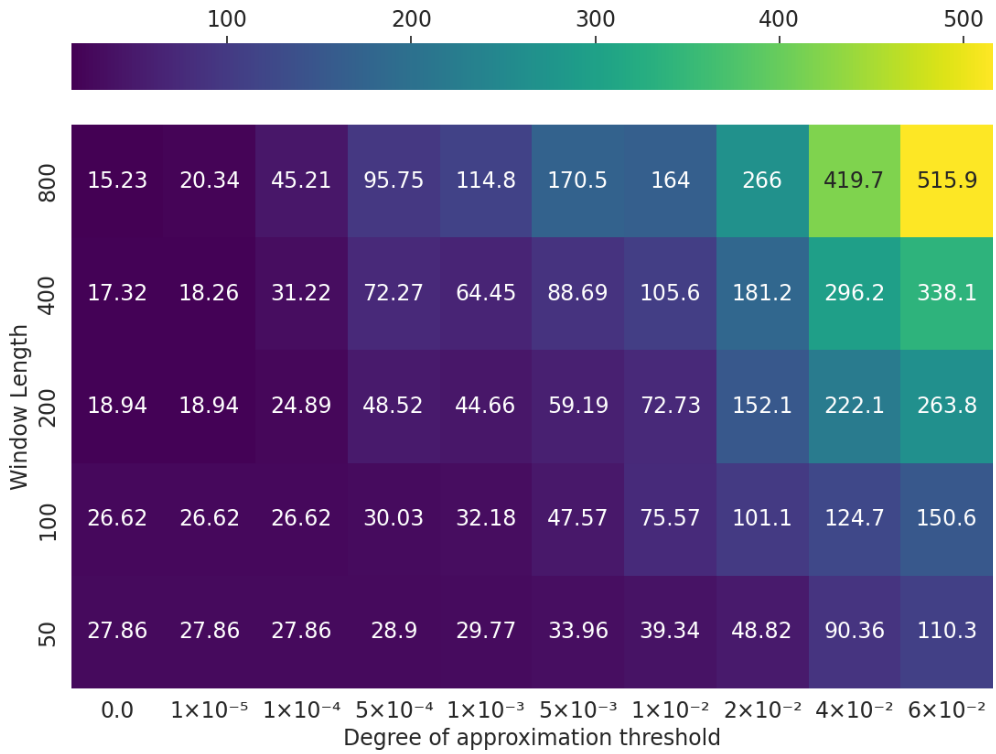

Figure 5. Average of samples for fault detection after the real fault inception for each pair of parameters.

The degree of approximation takes into account the number of pairs of tuples that violate a DC, but there are other possible metrics that we let for future work, such as the amount of tuples involved in violations and the minimum amount of tuples that need to be removed for the constraint not to be violated.

We also compared the performance of our method against two related works previously described in Section 1 [Asadi Majd et al. 2017, Coban and Tezcan 2021]. It is worth mentioning that we replicated both methods using the same dataset used to evaluate our approach. The results demonstrate the both approaches based on supervised machine learning algorithms also detected the fault for all test events (accuracy of 100%) just like the proposed unsupervised approach.

However, our method does not require supervision (i.e., labeled data) to infer the fault (anomaly) incidence, in other words, a human specialist delimiting if each sample or data window represents a failure for the algorithm training. This aspect can also reverberate in a better generalization of the proposed approach compared to supervised methods for fault detection. The utilization of supervised machine learning methods might necessitate separate training for each supervised transmission line. This is due to the variations in voltage and current signal amplitudes observed during the operation of each transmission line, influenced by factors such as generator power fluctuations, transmission line length, and voltage [Ensina et al. 2022]. Consequently, it is likely that these approaches will require an individual model for each supervised line.

On the other hand, our approach can indiscriminately identify faults despite these factors since the transmission lines present predominantly properties like symmetrical and balanced voltages/currents until the inception of a fault. Thus, the DCs found in this work are valid to use in other transmission lines (e.g., other datasets) for fault detection. The DCs are based on the correlation among the behavior of the voltage and current waveforms, according to the properties previously mentioned, and not by features extracted from the signals or predefined threshold values (e.g., constants).

Methods relying on feature extraction, such as those based on machine learning algorithms, may encounter challenges when confronted with varying feature values from other transmission lines exhibiting different amplitudes of voltage and current signals. This can lead to a lack of generalization capacity, as mentioned in [Ensina et al. 2022]. Instead, the proposed approach aims to eliminate the need for feature extraction, seeking to improve the method's capacity to adapt effectively to new, previously unseen data originating from other power systems (different datasets). Further exploration of this analysis is planned for future investigations.

The primary limitation of our approach lies in the computational expense associated with initially discovering the Denial Constraints (DCs). However, once these DCs are identified, the average time required for the approach, using a window length of $w = 800$, to calculate the degree of approximation and perform classification amounts to approximately 331.54 milliseconds (ms). It is noteworthy that the runtime cost exhibits a linear pattern concerning the window length, with times ranging from 20.75 ms for $w = 50$, 40.39 ms for $w = 100$, 78.37 ms for $w = 200$, 159.17 ms for $w = 400$, up to 331.54 ms for $w = 800$.

6. Conclusion and Future Work

This paper presented a DC-based approach for accurate fault detection in electric transmission lines. Since DCs capture complex rules in databases, we used them to represent the behavior of transmission lines with and without faults. We showed that the DCs discovered in pre-fault windows significantly differ from those in post-fault windows and can be used to detect faults. The results demonstrated accuracy and precision of 100% for this task, requiring only about 20 samples to determine the inception of the fault with a window size of 800 and a degree of approximation of 1×10^{-5}.

Future works include (1) the evaluation of the method using more failure events from the FADb dataset and examples of real fault data. Considering the DC algorithm employed in our approach, future works also include (2) testing the C-FASTDC algorithm [Chu et al. 2013] as it allows discovering DCs with constant values not covered by the DCFinder; (3) applying our DC approach in other data streaming applications to detect elusive events, like IoT and stock market; (4) assessing the predicates of the DCs for the fault classification task.

References

Abedjan, Z., Golab, L., and Naumann, F. (2015). Profiling relational data: A survey. *The VLDB Journal*, 24(4):557–581.

Aleem, S. A., Shahid, N., and Naqvi, I. H. (2015). Methodologies in power systems fault detection and diagnosis. *Energy Systems*, 6(1):85–108.

Asadi Majd, A., Samet, H., and Ghanbari, T. (2017). k-nn based fault detection and classification methods for power transmission systems. *Protection and Control of Modern Power Systems*, 2(32):1–11.

Belagoune, S., Bali, N., Bakdi, A., Baadji, B., and Atif, K. (2021). Deep learning through lstm classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems. *Measurement*, 177:109330.

Braverman, V. and Ostrovsky, R. (2010). Effective computations on sliding windows. *SIAM J. Comput.*, 39(6):2113–2131.

Chen, K., Hu, J., and He, J. (2018). Detection and classification of transmission line faults based on unsupervised feature learning and convolutional sparse autoencoder. *IEEE Transactions on Smart Grid*, 9(3):1748–1758.

Chu, X., Ilyas, I., and Papotti, P. (2013). Discovering denial constraints. *PVLDB*, 6:1498–1509.

Coban, M. and Tezcan, S. S. (2021). Detection and classification of short-circuit faults on a transmission line using current signal. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 69(4).

Ensina, L. A., Oliveira, L. E. S., Almeida, E. C., Santos, S. L. F., and Bernardino, L. S. (2022). Fault classification in transmission lines with generalization competence. In *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6.

Ferreira, V. H., Zanghi, R., Fortes, M. Z., Gomes, S., and Alves da Silva, A. P. (2020). Probabilistic transmission line fault diagnosis using autonomous neural models. *Electric Power Systems Research*, 185:106360.

Ferreira, V. H., Zanghi, R., Fortes, M. Z., Sotelo, G. G., da Boa Morte Silva, R., de Souza, J. C. S., Guimarães, C. H. C., and Júnior, S. G. (2016). A survey on intelligent system application to fault diagnosis in electric power system transmission lines. *Electric Power Systems Research*, 136:135–153.

Furse, C. M., Kafal, M., Razzaghi, R., and Shin, Y.-J. (2021). Fault diagnosis for electrical systems and power networks: A review. *IEEE Sensors Journal*, 21(2):888–906.

Gilbert, D. and Morrison, I. (1997). A statistical method for the detection of power system faults. *International Journal of Electrical Power & Energy Systems*, 19(4):269–275.

Grainger, J. J., Stevenson, W. D., and Chang, G. W. (2016). *Power System Analysis*. McGraw-Hill Education, 2 edition.

Høidalen, H. K., Prikler, L., and Peñaloza, F. (2019). *ATPDraw version 7.0 for Windows - Users' Manual*. ATPDraw.

Mishra, D. and Ray, P. (2018). Fault detection, location and classification of a transmission line. *Neural Computing and Applications*, 30:1377–1424.

Pena, E. H. M., de Almeida, E. C., and Naumann, F. (2019). Discovery of approximate (and exact) denial constraints. *PVLDB*, 13(3):266–278.

Prasad, A., Belwin Edward, J., and Ravi, K. (2018). A review on fault classification methodologies in power transmission systems: Part—i. *Journal of Electrical Systems and Information Technology*, 5(1):48–60.

Raza, A., Benrabah, A., Alquthami, T., and Akmal, M. (2020). A review of fault diagnosing methods in power transmission systems. *Applied Sciences*, 10(4):1–27.

Singh, S. and Vishwakarma, D. N. (2015). Intelligent techniques for fault diagnosis in transmission lines — an overview. In *International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE)*, pages 280–285.

Yadav, A. and Dash, Y. (2014). An overview of transmission line protection by artificial neural network: Fault detection, fault classification, fault location, and fault direction discrimination. *Advances in Artificial Neural Systems*, 2014:1–20.