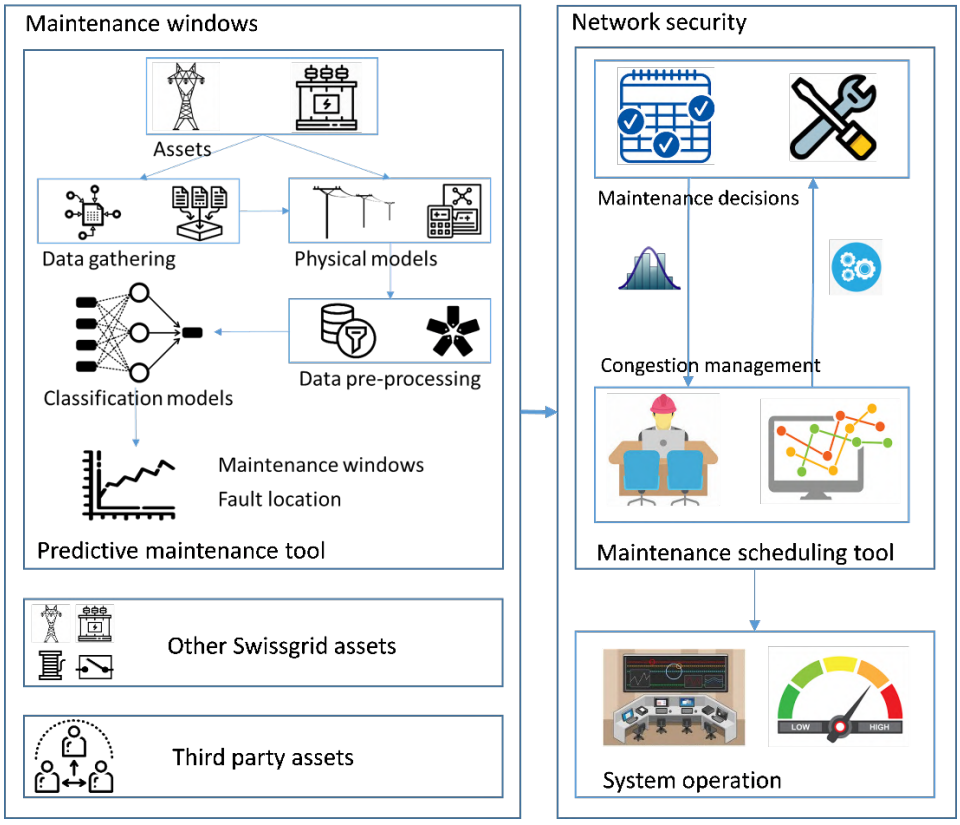
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of the Environment, Transport, Energy and Communications DETEC

Swiss Federal Office of Energy SFOE
Energy Research and Cleantech Division

Final report dated February 3, 2024

IMAGE - Intelligent Maintenance of Transmission Grid Assets



Source: ETHZ-Swissgrid

Date: February 3, 2024

Location: Bern

Publisher:
Swiss Federal Office of Energy SFOE
Energy Research and Cleantech
CH-3003 Bern
www.bfe.admin.ch

Authors:
Laya Das, Reliability and Risk Engineering Laboratory - ETH Zürich, laydas@ethz.ch
Mohammad Hossein Saadat, Reliability and Risk Engineering Laboratory - ETH Zürich, msaadat@ethz.ch
Blazhe Gjorgiev, Reliability and Risk Engineering Laboratory - ETH Zürich, gblazhe@ethz.ch
Giovanni Sansavini, Reliability and Risk Engineering Laboratory - ETH Zürich, sansavig@ethz.ch

SFOE Project coordinators:
Michael Moser, michael.moser@bfe.admin.ch

SFOE Contract number: SI/502073-01

Zusammenfassung

Dieses Projekt zielt darauf ab, i) innovative Methoden für die datengestützte Fehlererkennung und -diagnose von Stromnetzanlagen zu erforschen und ii) die Entwicklung von Algorithmen für die Wartungsplanung zu unterstützen. In diesem Projekt wurden drei Hauptanlagen betrachtet, nämlich Stromleitungen, Isolatoren und Leistungstransformatoren. Das Projekt nutzte eine Vielzahl von Daten (Strommessungen, Luftbilder und Gaskonzentrationsmessungen), die vom Industriepartner gesammelt wurden, sowie verschiedene Modelle zur Musterextraktion. Modernste physikalisch basierte Modellierung, evolutionäre Algorithmen, Deep-Learning-Architekturen, probabilistische Methoden und Transfer-Learning wurden eingesetzt, um die Merkmale realer Daten wie Unvollständigkeit, Unsicherheit, Variabilität und begrenzte Datensatzgrößen zu berücksichtigen. Letzteres stellt eines der Haupthindernisse bei der Anwendung von Machine/Deep Learning für die Diagnose von Anlagen im Stromnetz dar. Die Projektergebnisse haben gezeigt, dass bestehende maschinelle Lernmodelle bei der Durchführung von Analysen auf Komponentenebene sehr genau sein können, vorausgesetzt, es sind ausreichende und qualitativ hochwertige Daten verfügbar. Die wichtigsten Ergebnisse dieses Projekts sind fünf Forschungspapiere und drei Tools, die unserem Projektpartner Swissgrid AG zur Verfügung gestellt wurden.

Résumé

Ce projet vise à i) explorer des méthodes avancées pour la détection des défauts et le diagnostic des composants du réseau électrique à partir de données et ii) soutenir le développement d'algorithmes de planification de la maintenance. Trois composants principaux ont été pris en compte dans ce projet, notamment les lignes électriques, les isolateurs de pylônes et les transformateurs de puissance. Le projet a utilisé une multitude de données (mesures de courant, images aériennes et mesures de concentration de gaz) acquises par le partenaire industriel, ainsi que divers modèles d'extraction de modèles. La modélisation basée sur la physique, les algorithmes évolutionnaires, les architectures d'apprentissage profond (deep learning), les méthodes probabilistes et lapprentissage par transfert (transfer learning) ont été adoptés pour tenir compte des caractéristiques des données du monde réel, telles que l'incomplétude, l'incertitude, la variabilité et la taille limitée des ensembles de données. Ce dernier point constitue l'un des principaux obstacles à l'application de l'apprentissage automatique et de l'apprentissage profond au diagnostic des composants du réseau électrique. Les résultats du projet ont révélé que les modèles d'apprentissage automatique existants peuvent être très précis dans l'exécution d'analyses au niveau des composants, à condition que des données suffisantes et de haute qualité soient disponibles. Les principaux résultats de ce projet sont cinq articles de recherche et trois outils livrés à notre partenaire de projet Swissgrid AG.

Summary

This project aims to i) explore advanced methods for data-driven fault detection and diagnostics of power grid assets and ii) support the development of maintenance scheduling algorithms. Three main assets were considered in this project, namely power lines, tower insulators, and power transformers. The project used a host of data (current measurements, aerial images, and gas concentration measurements) acquired by the industrial partner along with diverse pattern extraction models. State-of-the-art physics-based modeling, evolutionary algorithms, deep learning architectures, probabilistic methods, and transfer learning were adopted to account for the characteristics of real-world data such as incom-

pleteness, uncertainty, variability, and limited dataset sizes. The latter proves to be one of the main obstacles in applying machine and deep learning to power grid asset diagnostics. The project results revealed that existing machine learning models can be highly accurate in performing component-level analyses given that sufficient and high-quality data is available. The main outputs of this project are five research papers and three tools delivered to our project partner Swissgrid AG.

Main findings

Three types of power system assets were considered in the project – transmission lines, transmission tower insulators, and power transformers - that are part of the transmission system of the grid and crucial in ensuring reliable power supply. Machine learning (ML) and deep learning (DL) algorithms were used to assess the state of operation (e.g., healthy, unhealthy, etc.) of the individual components.

1. *Transmission Lines*: Deep neural networks coupled with a physics-based model of transmission lines can be used to detect faulty segments. The physics-based model acts as a digital surrogate of the physical line and the deep neural network performs advanced pattern extraction to detect the operation state accurately. The robustness of the deep neural network models to uncertainty can be improved by adopting probabilistic models, resulting in more reliable models.
2. *Transmission Tower Insulators:* Computer vision-based object detection models can be used to detect healthy and faulty insulators in transmission tower insulators from aerial images captured by drones. The performance of fault detection can be improved with a two-stage approach that first uses an object detection model to identify assets followed by an anomaly detection model that can distinguish between healthy and faulty assets. An anomaly detection model in the second stage additionally offers the ability to perform fault detection in the presence of very few faulty data points.
3. *Power Transformers:* Statistical and machine learning models can be used to detect the state of health of a transformer using various measurements and recorded past events. Dissolved gas analyses performed on Swiss power transformers serve as the main data set for training and testing models. The lack of true labels can be to some extent overcome by developing conventional methods. In particular, conventional methods can be used to identify the type of faults in a transformer. These outputs can be used to train a machine learning model.

Main outcomes

The main outputs of this project are i) five research papers (see Section 8) and ii) three tools were delivered to the project partner Swissgrid AG:

- Object detector model that can be utilized to detect insulators from tower images
- Algorithm that encompasses all of the relevant conventional methods for transformer fault detection and diagnostics from DGA data
- Trained ML model for transformer fault detection and diagnostics from DGA data

Contents

Abbreviations

AC	Alternating Current
ADF	Assumed Density Filtering
AUC	Area Under (receiver operating characteristics) Curve
BND	Bird Nest Dataset
CNN	Convolutional Neural Network
CWT	Continuous Wavelet Transform
DL	Deep Learning
DNN	Deep Neural Network
EPRI	Electric Power Research Institute
Faster RCNN/FRCNN	Faster Region proposal-based Convolutional Neural Network
FCDD	Fully Convolutional Data Description
FCOS	Fully Convolutional One Stage
FFT	Fast Fourier Transform
FNN	Feedforward Neural Network
GA	Genetic Algorithm
GPS	Global Positioning System
IDID	Insulator Defect Image Dataset
LSTM	Long Short-Term Memory
ML	Machine Learning
MS-COCO	Microsoft Common Objects in Context
NLL	Negative Log Likelihood
OOD	Out-Of-Distribution
PDF	Probability Density Function
RGB	Red Green Blue
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
RPN	Region Proposal Network
SSD	Single Shot multibox Detector
STFT	Short Time Fourier Transform
STN-PLAD	Power Line Assets Detection
SVM	Support Vector Machine
TSO	Transmission System Operator
UQ	Uncertainty Quantification
YOLO	You Only Look Once

1 Introduction

1.1 Background information and current situation

The power grid is a longstanding national critical infrastructure, which after going through decades of geographical expansion and technological improvement, is facing the challenge of aging components. Until now, conservative maintenance schemes of the second generation, referred to as preventive maintenance, are used, where assets in the infrastructure are maintained according to a fixed schedule. While aiming to avoid unplanned outages, these maintenance schemes are expensive. Moreover, a fixed schedule of asset maintenance can result in healthy assets being subjected to unwarranted maintenance routines, and unhealthy assets in active operation, risking faults and outages. Budget constraints for owners of transmission and distribution systems along with the need for a more reliable approach to asset maintenance pave the way for third-generation maintenance schemes, referred to as predictive or reliability-centred maintenance schemes. This project aims to develop predictive maintenance solutions for system operators at a component level. The project considers three components; a brief discussion of the state-of-the-art health monitoring techniques for these assets is provided below.

1.1.1 Power transmission lines

Overhead transmission lines are one of the most critical assets in the power transmission infrastructure. They are often subject to fluctuations in the environment such as daily and seasonal variations in humidity and temperature, as well as rain, wind, lightning, and other weather phenomena. These conditions can deteriorate the electrical properties of lines as well as the supporting structures on towers, i.e., insulators, thereby weakening the insulation of the lines and causing a current flow from the high voltage conductor to the ground via the outside surface of the insulator [1]. This so-called leakage current from the lines ultimately leads to short circuits and to the loss of power supply [2]. Detection of such abnormal operating conditions, i.e., *faults* in transmission lines is therefore crucial to ensuring a reliable supply of power and has been extensively studied in the literature [1, 3, 4].

Faults in transmission lines are typically short circuits and are categorized as line-to-ground (short circuit between a line and ground) and line-to-line (short circuit between two or more lines) [5, 6]. Many techniques are proposed in the literature to detect and locate different types of short circuit faults in transmission lines [3, 5]. The conventional methods in the literature broadly employ impedance-based or traveling wave-based techniques. Impedance-based techniques detect and locate a fault on the basis of changes in the line impedance during the fault from current and voltage measurements [7, 8]. Traveling wave-based techniques make use of the traveling wave created by the fault and use time of arrival at measuring stations to locate the fault [9, 10]. In addition to conventional methods, several signal processing and data analysis-based methods have been proposed for fault detection. These methods involve the use of S-transform, Hilbert Transform, and alienation coefficient [11], entropy of signal [12] and curve fitting and principal component analysis [13].

Extensive advancements are made in the utilization of machine learning (ML) methods for prognostic and health monitoring of structures, systems, and components [14, 15]. In particular, there is increasing interest in the development of ML methods for monitoring and fault detection in power grids. Early attempts made use of a support vector machine (SVM) for detecting faults and differentiating them from power swings [16]. The authors in [17] use relevance vector machines as a parsimonious alternative to SVMs with similar performance. Neural networks are applied to predict imminent failures based on the current system state in a smart power grid [18]. Furthermore, deep neural networks consisting of long short-term memory (LSTM) units are used to capture differences in the temporal pattern of measurements during healthy and faulty operation and detect and classify different types of fault [19]. The

work in [20] uses LSTMs to detect, classify, and locate the different types of faults. Hybrid deep neural networks with convolutional neural network (CNN) modules along with LSTM modules have also been proposed for processing of frequency response for fault detection and localization [21]. The authors in [22] used probabilistic neural networks to detect and classify faults and reported significantly less requirement of data with performance comparable to contemporary works. The literature has also proposed hybrid methods that combine signal processing, data analysis, and machine learning for fault detection and localization. For instance, the work in [23] performs wavelet transform of the data and uses the resulting images as inputs to capsule neural networks to detect and classify faults. This approach was modified to process the Gramian angular field with a self-attentive capsule network that allowed for the diagnostics of the fault [24]. The wavelet transform has also been combined with support vector machines [25], extreme learning [26], and neural networks [27] for fault detection and localization. A detailed review of the different methods and algorithms used for fault detection and localization in transmission lines can be found in [5, 6, 28].

The vast majority of the ML in the literature target the detection and localization of short circuits. Such an approach allows the transmission system operator to take corrective maintenance actions after the occurrence of the fault. However, detecting the fault early can help remove the fault before it triggers protection activation and thereby line disconnection. Such a predictive maintenance approach can help in the planned replacement of excessively degraded components, and therefore, avoid unplanned outages. In this regard, the authors in [29] employ the frequency spectrum of the leakage current and the relative magnitude of different harmonics to monitor the lines online. A physics-based model of a real line is developed and used to simulate different faults by introducing different values of capacitance in the line. The harmonic component of the leakage current, computed using the Fast Fourier Transform (FFT) is then used to train a feedforward neural network (FNN) for detection and localization of faults of different magnitudes. The neural network predicts the location of the fault and the associated capacitance. The impact of environmental conditions on the insulation properties of transmission lines is studied in [30] and a method of monitoring the quality of insulation was proposed. A model of the transmission line is proposed to replicate different faults by changing the capacitance, and a comparison of the simulated and measured data allows for the detection of faults at an early stage. The work presented in [31] performs qualitative trend analysis of the harmonics of the leakage current, followed by approximating the trend with polynomial expressions. A Naive Bayes model is then used to detect faults. The results reveal that the approach is able to detect and localize multiple faults present simultaneously in the system.

The presented ML methods for the detection of leakage current for predictive maintenance have at least one of the following drawbacks. First, the lack of detail on the physics-based model in the previous works makes it extremely difficult to replicate such a model. Second, the localization of faults is formulated as a regression problem, and the classification formulation is not explored. The synthetic data generated from the model can only generate faults in discrete segments, not as a continuous location along the line. Furthermore, from a practical perspective, the problem of localization of degraded insulators is discrete because they are located on spatially fixed towers. Therefore, such a problem is inherently a classification problem, and posing it as regression can result in reduced performance. Third, the machine learning model used is a very basic version of neural networks and does not exploit the latest developments in deep learning. Moreover, the work in [29] makes use of only 41 samples to train their neural network, which renders the model prone to overfitting. The work in [31] uses a Naive Bayes classifier, which inherently assumes that the different variables in its input are independent of each other. This assumption cannot be verified with the spectral components of the leakage current as input to the classifier. As a result, the performance of the model can be poor despite a validated physics-based model. Finally, the articles in the literature make use of hand-crafted features, for example, spectral information of the harmonics in leakage current, and do not exploit the powerful automatic feature extraction capability of deep neural networks.

It is also important to note that obtaining data to train DNN models is not always possible. As

a result, physics-based or data-driven models are used to mimic the behavior of the system under different operating conditions and generate synthetic data. However, the resulting DNN models for fault diagnosis are trained with only a digital approximation (digital twin) of the underlying system. Digital twins are developed with a finite amount of data collected from the system and can have uncertainty in their structure and/or parameters. In certain instances, this data can be limited in accurately capturing the dynamics of the system, while in others, data corresponding to different operating states of the system might not be available. This can accentuate the inaccuracies in correctly replicating the system dynamics, causing a mismatch between the data seen by the DNN at the time of training, and the data it encounters at the time of deployment.

The mismatches between the training and deployment data can expose the DNN to out-of-distribution (OOD) samples, resulting in incorrect predictions. This motivates the need to quantify the reliability of the DNN model for the correct interpretation of its predictions. UQ in these situations provides additional information regarding the confidence of DNN's predictions, which can be used to inform downstream decision-making. Specifically, a highly confident prediction of the DNN can be interpreted as being reliable, while poor confidence can point to OOD data encountered by the model. This information is useful in two aspects - (1) it can be used to suggest the end-user exercise caution in downstream decision-making, and (2) such OOD data can be recorded to fine-tune the model further and improve its performance in the future. This article caters to this cause and proposes a system-agnostic framework for UQ of DNNs built from digital models of real-life systems.

1.1.2 Power transmission insulators

Power transmission insulators are an important component of the transmission system and play a key role in ensuring a reliable supply of power [32]. They act as a bridge between transmission lines and transmission towers allowing for physical support, while preventing leakage of current through the tower [1]. Much like other components of the transmission grid, they are subject to day-to-day variations in operation caused by fluctuating demand and generation, as well as extreme environmental conditions such as lightning strikes, wind storms and rain. These factors, in addition to the deposition of dust, rising daily temperatures and rapidly changing weather conditions deteriorate the quality of insulation, thereby allowing current to leak [3, 4]. The leakage current can worsen over time and ultimately result in a protection device activating and disconnecting the line [2]. Therefore, regular insulator inspection and maintenance is important to ensure reliable operation of the system [33].

There is an increasing interest in exploiting powerful deep learning models to perform automated monitoring of the health of insulators. One class of these studies relies on patterns manifested in measurements of current (or derivatives thereof) from multiple towers and makes use of machine learning and deep learning methods for pattern recognition [34]. The second class of studies, and the topic considered in this article, involves the collection of aerial images of insulators and using deep learning models to detect and classify faulty insulators. Such an automated process reduces the reliance on human experts to inspect insulators and the associated costs and human errors. This also increases the safety of the data acquisition process, which involves controlling a drone from a distance instead of physically climbing the tower to collect the images.

Several articles in the literature have developed machine learning and deep learning-based solutions for assessing the state of insulators from aerial images. The authors in [35] use saliency maps for image segmentation with 200 images through static local and global feature extraction. A weighted combination of the local and global saliency maps is used to extract insulators in an image. A Faster RCNN object detection model is applied to detect three different types of insulators in [36]. A dataset with 1000 images for each type of insulator is used to train the model, which is then tested on 500 images of each type of insulator. The authors also carry out fault detection with missing caps of insulators with the model, which is trained with 80 images and tested on 40 images, yielding 92% precision. The Single-Shot multi-box

Detector (SSD) is utilized in [37] to detect healthy and faulty insulators. The model is trained with 385 images and achieves a precision of 92.48% on a test dataset of 100 images. The authors in [38] employ a modified object detector inspired by the highly successful You Only Look Once version 2 (YOLOv2) model to detect missing disks in insulators. The authors compile a dataset of 4031 images and use modified YOLO models to achieve a detection precision of 94.2% for insulators with single missing disk and 98.3% for insulators with multiple missing disks. Similar approaches that modify the model architecture to improve detection performance are proposed in the literature [39, 40, 41, 42]. In [43], the authors use bounding boxes instead of rectangular boxes to closely capture an insulator, thereby allowing a neural network to learn the features relevant to the insulator with minimal interference from the background. The work uses a dataset of 3700 images and trained a Faster RCNN model to detect healthy and faulty insulators (with missing disks) and achieved a precision of $\sim$ 90% for the different subsets of their dataset.

Despite the above advances, key challenges still remain unexplored. *First*, research on differentiating between healthy and unhealthy insulators focuses only on detecting missing caps. However, insulators can also have discolored disks as remnants of electrical flashes and broken disks caused by physical damage. The knowledge of these damages is important for the operator to assess the state of health of insulators and prioritize maintenance activities. Detecting discoloration patches and irregular disks shapes is much more difficult than detecting the presence or absence of the entire cap. This is because the patterns in the former case are finer and confined to a smaller region in the image (on the disk) while the patterns in the latter case are much more pronounced and span a relatively larger portion of the image. The ability of deep neural network models to capture these nuances and detect difficult patterns on insulators is not researched in the literature.

Second, the vast majority of the literature tackles the problem of fault detection from the perspective of model design, and little effort has been made to augment the richness of the data used to train and evaluate these models. While the development of models for traditional computer vision applications has immensely benefited from benchmark datasets such as Imagenet [44] and MS-COCO [45], inspection of insulators from aerial images faces the challenge of limited data availability in the public domain. Consequently, a large number of insulator inspection models are trained with only one dataset, which is also the target dataset intended for deployment of the model, limiting analysis of the generalisability of models. In such scenarios, the network sees only limited variability in terms of the background and features of objects in the foreground (e.g., different colors of insulator disks, different number and density of disks in the insulator, different types of discs/insulators, different positions of the insulators, etc.). The development of a rich dataset that can improve the features learned by the model has not been studied in the literature.

Third, components such as Stockbridge dampers, bird nests, and overgrowing vegetation are important to be inspected from a monitoring and maintenance perspective of the system operator. Although there have been studies to detect these objects, in particular, bird nests [46, 47] and overgrowth of surrounding vegetation [48], the development of a single model capable of detecting a multitude of these objects has not been extensively studied in the literature. Such a multi-object detection model provides a compact way of inspecting multiple assets in an image.

Fourth, class imbalance and size imbalance can be major challenges impeding the detection of smaller and infrequently observed objects, i.e., flashed and broken disks. Thus, while object detection models can simultaneously detect multiple objects in an image, their performance is still limited by the volume of available data. This poses a challenge to automating the inspection process for a well-functioning grid since the vast majority of insulators will be healthy. Thus, this challenge cannot be tackled through the collection of more images and requires methodological improvement.

1.1.3 Power transformers

Power transformers are one of the most essential and costly assets of power systems [49, 50]. Failure of the transformer may lead to high repair costs or, worse, irreversible internal damages [49, 50]. Furthermore, failure can result in power supply interruption, causing loss of profit for plant owners and electrical energy shortage on the demand side [50]. Therefore, an urgent need is given to monitor the transformers' health, detecting and classifying faults at an early stage and taking the the necessary maintenance actions. The academic literature offers multiple approaches and tools for transformer health diagnostics, including conventional methods, Artificial Intelligence (AI), and advanced Machine Learning (ML) methods. Different tests of the transformer are used to collect data and thus detect and classify faults. Dissolved Gas Analysis (DGA), Frequency Response Analysis" (FRA), oil quality test, and "Partial Discharge" (PD) monitoring are among others, examples of commonly used techniques. For DGA fault diagnostics approaches fall into two categories, rule-based methods and ML methods [51]. Since rule-based methods such as the Duval triangle [52] or the Doernenberg [52] ratios method do not guarantee high accuracy, recently, more academic work has focused on ML-based methods [51]. Many different approaches in literature use DGA datasets for the training of ML algorithms, that learn to detect and classify fault types [53, 50, 54].

DGA alone may not be sufficient to evaluate the health of a transformer for various reasons. Some pre-failure conditions do not cause gas production, on the other hand, not every increase in gas concentration indicates a fault [52]. Therefore, more information provided by other tests is needed. ML-based techniques for tests other than DGA are less present in the literature. Those focusing on PD monitoring, aim to distinguish between noise and PD or to classify the type of discharge [55, 56]. However, most work dealing with PD monitoring in combination with ML algorithms uses data from experimental setups instead of real-world data [55, 57] that can be typically obtained by transformer operators. FRA belongs to one of the most well-established diagnostics techniques with high accuracy. In [58], the authors use ML approaches to increase the comprehensibility of the results and thereby decrease the level of expertise costs [58, 59]. In [60], AI is used to determine the electric parameters of a transformer using FRA.

Combining the results of tests enables the assessment of the health condition of the transformer. The main approaches to represent the status are the Health Index (HI) and Remaining Useful Lifetime (RUL). With the HI, the condition of the transformer is usually ranked within five categories from very bad to very good. This might be done in a conventional fashion, by weighting different test results and summing them to a final output that is then ranked within the condition states [61]. More advanced methods that use fuzzy logic or Bayesian networks to determine the HI are presented in [62, 63]. The RUL estimates the investigated transformer has a lifetime until irreversible failure [64, 65].

Several review papers are found in the literature, each focusing on different aspects of a power transformer health assessment. In [66] the authors summarize different conventional methods used to assess a transformer's condition based on the results of a DGA. In [67] the authors review advanced methods for DGS analyses, including fuzzy logic, evolutionary optimization-based approaches, and neural networks. In [68], the authors review several approaches toward predictive maintenance, however, these are general and do not have a focus on power transformers. The majority of the literature focuses on DGA data to detect faults and perform fault diagnostics in transformers. We have reviewed 119 papers with ML-based interpretation of DGA data. Overall, the literature shows that both conventional and ML methods are able to identify patterns in DAG data and achieve acceptable accuracy.

1.2 Purpose of the project

The purpose of this project is to explore the field of predictive maintenance for power system applications. Therefore, we aim to develop data-driven component-level algorithms that enable early fault detection and diagnostics. Such research will equip us with knowledge of the methods and the data we need to develop tools for practical application in the domain of power system safety.

1.3 Objectives

The main objective of this project is to develop advanced tools and methods for data-driven fault detection and diagnostics of power lines, insulators, and transformers. The developed tools that show promise for practical applications are to be further tested and used by the TSO operator of Switzerland.

1.3.1 Power transmission lines

We answer the following research questions for power lines: i) can the leakage current be reproduced with a physics-based model; (ii) is the leakage current a useful indicator to accurately locate faulty insulators; and iii) which ML method provides the best accuracy? The contributions are fourfold:

1. **Physics-based model of real power line:** A detailed easy-to-replicate physics-based model of an electric power line is developed. This model is calibrated with data from a real power line between Avegno and Gorduno.
2. **Fault diagnostics with synthetic data:** The physics-based model is used to synthetically generate healthy and faulty operation data for the line. This data is used to train several neural network models and presents a comparison of their performance.
3. **Probabilistic deep learning:** The synthetic nature of the data is acknowledged and probabilistic neural networks are trained to achieve better robustness to uncertainty in the data generating process. Uncertainty quantification is performed to capture the uncertainty and train more robust models
   (a) *aleatoric uncertainty* is quantified with (i) explicit propagation of uncertainty and (ii) implicit prediction of uncertainty. The former approach can be used when uncertainty in training data is available, while the latter can be used without such information.
   (b) *epistemic uncertainty* is quantified with the popular Monte Carlo Dropout approach.

   We calculate the performance of probabilistic and deterministic models on data generated from different instances of the physics-based model. In addition, we evaluate the robustness of the DNN models to out-of-distribution (OOD) samples.

1.3.2 Power transmission insulators

1. **Object detection tasks for asset inspection and incipient fault detection:** Three object detection tasks are formulated with increasing complexity, i.e., detection of insulators, diagnostics of incipient faults, and detection of multiple objects of interest, to evaluate and compare state-of-art object detection models for asset inspection. Diagnosing incipient faults is a much more difficult task than the previously studied in the literature. In addition, it also provides information on the nature of faults, which is important in practice for transmission system operators for system assessment as well as repair and maintenance scheduling.
2. **Preparation of a reference dataset:** A large dataset from multiple sources that have made their respective datasets publicly available is curated. These datasets contain images of transmission

towers with multiple objects of interest, such as insulators, Stockbridge dampers, and bird nests. Such a database with images collected by different agencies in multiple countries inherently exhibits a high richness in terms of foreground and background features.
3. **Models for asset inspection:** Four popular object detection models to perform the three detection tasks with the reference dataset. Fine-tuning of the models to a small target dataset from Swissgrid with transfer learning is performed and comparing the performance with a model trained from scratch only on the target dataset.
4. **Sensitivity analysis:** Different sizes of training datasets were used for fine-tuning a pre-trained model to study how well the models perform under data scarcity.
5. **Anomaly detection:** The detection of flashed and broken disks is posed as an anomaly detection problem and a state-of-the-art explainable one-class classification-based anomaly detector is adopted. This model takes images of disks as input, which can be generated beforehand by an object detection model. The performance of this approach is studied with two datasets of different sizes, i.e., in data-abundant and data-scarce scenarios. The use of data from both datasets to improve the performance of the model in the data-scarce scenario is studied.

1.3.3 Power transformers

For transformers, we answer the following research questions: i) which data is needed to train ML models for fault identification and classification; ii) how ML models perform; iii) how to cope with low data entries and the lack of certain labels.

1. **Preparation of a data set:** The main challenge here is to obtain reliable and high-quality data. Since we focus on DGA data, it is likely that 1) the transformers operated in one country differ from those in other countries, and 2) the measuring accuracy standards may differ. These challenges can be overcome by collecting data from transformers operated in similar conditions and DGA is performed by accredited institutions.
2. **Labeld data set:** Often the labels, i.e., the fault detection (healthy, faulty) and fault diagnostics (partial discharge, thermal faults) are not known. To confirm the correct labels a thorough inspection often needs to be performed by experts. Therefore, reliable labels are often not available. In this study, we face the lack of accurate labels.
3. **Model devlopement:** We develop both conventional approaches and ML models to identify 1) if a transformer is healthy and 2) if not, what is the type of fault. The conventional approaches are used to identify the initial labels, while the ML approaches are trained on such data.

2 Power transmission lines

We adopt two approaches for fault diagnostics of power lines. First, we build a physics-based model of a real-life power line, calibrated with data collected from the system. This model acts as a surrogate model of the line and is used to simulate different operating conditions of the line. We then use the simulated data to train a host of deep learning models that identify the faulty segments of the line. We note that the physics-based model is calibrated with data from a healthy system and used to generate synthetic data for healthy and faulty conditions. Thus, the data generated for faulty conditions can have more uncertainty than those for healthy conditions. In order to account for this uncertainty, we also train probabilistic deep learning models that can learn parameters in an uncertainty-aware manner.

2.1 Methods for fault power lines fault detection and classification

Here, we present the methods we use for fault detection and classification in power lines.

2.1.1 Physics-based modeling and calibration

Figure 1 shows the modeling approach we developed to build a physics-based model of power lines. This approach is applicable to any high-voltage line and herein we utilize it to build a physics-based model of a 220 kV power line that connects the nodes Avegno and Gorduno located in the Swiss extra-high voltage power grid (Figure 2). The length of the transmission line is 26.53 km and it consists of 26 segments (Table 1). The line characteristics per unit length are, respectively, resistance r = 0.072 Ω/km, inductance l = 1.367 mH/km, and capacitance c = 9 nF/km. Furthermore, to account for the insulator resistance R_I a value of 6 MΩ is adopted according to [69]. Table 1 shows the modeled Avegno-Gorduno segments, including the segment length, the number of towers, and the type of insulators.

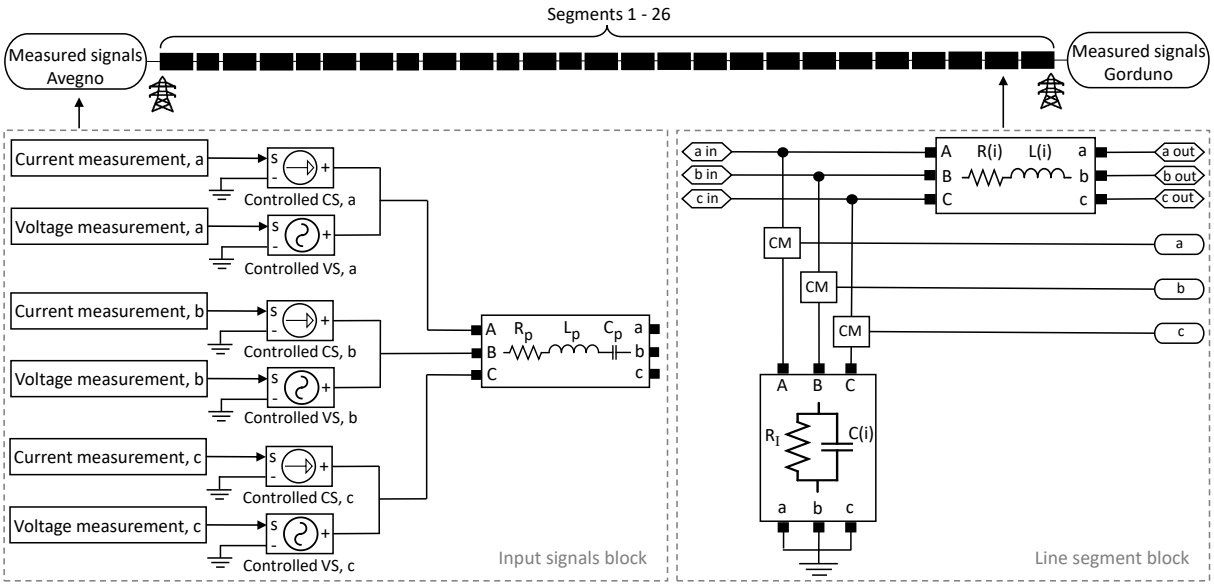


Figure 1: A graphical description of the physics-based model of the Avegno and Gorduno power line. CS stands for current source, VS stands for the voltage source, and CM stands for current measurement.

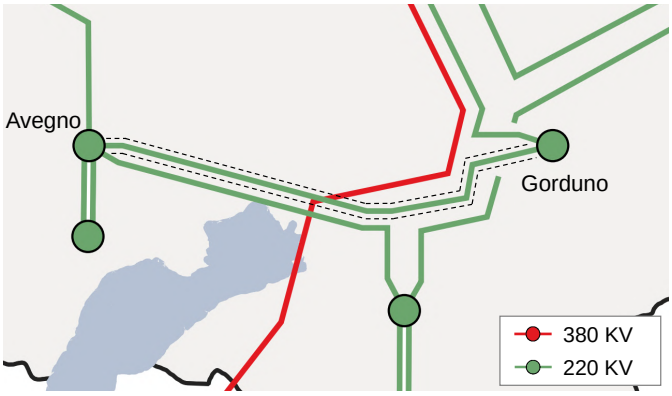


Figure 2: The modeled 220 kV Avegno-Gorduno line [70].

Table 1: The segments of the 220 kV Avegno-Gorduno line.

Segment	Length (km)	Towers	Type of Insulator
1	0.773	2	new chains
2	1.177	5	old chains
3	0.38	2	new chains
4	0.78	1	old chains
5	1.781	6	new chains
6	0.764	3	old chains
7	0.919	2	new chains
8	1.116	4	old chains
9	1.122	2	new chains
10	1.15	3	old chains
11	0.413	1	new chains
12	0.329	1	old chains
13	2.241	6	new chains
14	1.045	3	very new chains
15	1.045	3	very new chains
16	1.045	3	very new chains
17	1.045	3	very new chains
18	1.045	3	very new chains
19	1.045	3	very new chains
20	1.045	3	very new chains
21	1.045	3	very new chains
22	1.045	3	very new chains
23	1.045	3	very new chains
24	1.045	3	very new chains
25	1.045	3	very new chains
26	1.045	3	very new chains

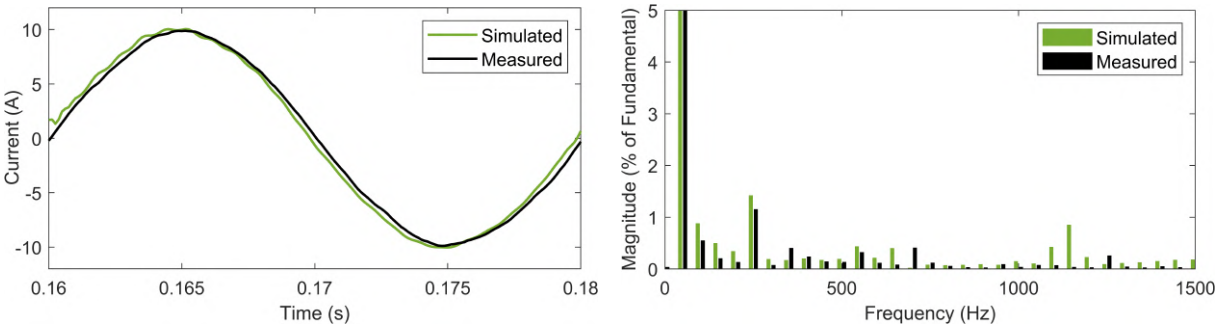
The line model is developed in MATLAB Simulink [71]. Each segment is modeled with two modules, RL branch and RC branch, as shown on the right-hand side of Figure 1 (Line segment block). The RL component consists of resistance $R(i)$ and inductance $L(i)$ connected in series. The RC component, which represents the line insulation, consists of resistance R_l and capacitance $C(i)$ connected in parallel. The $R(i)$, $L(i)$, and $C(i)$ for each segment i (i=1,2,3...,26) are calculated from the parameters of the line, e.g., $R(i) = r * length\_segment(i)$. The leakage current, which is simulated as the flow of current through the RC component to ground, is measured for each phase (a, b, c) via the current measurement (CM) blocks in Figure 1. The total leakage current simulated with the model is calculated as the sum of the simulated leakage currents of all the segments. The change in the values of the RC component at a given segment directly affects the leakage current at that segment. This allows for the creation of a synthetic data set that represents different states of health of the insulators along the power line.

The line model is fed with the time series of the current and voltage at Avegno and Gorduno. The time series are obtained by two GPS-synchronized power system phase measurement units (PMUs), which take the measurements at both ends of the line with a sampling frequency of 8 kHz. Note that there are no measurements at each segment denoted by Figure 1. Since the measurements are subject to errors, the measured data is smoothed to minimize measurement noise. In total, 1 second of measurement data, i.e., 8000 data samples of current and voltage of each phase are separately loaded into the system as shown in the left side of Figure 1 (Input signals block). The current and voltage data sets are connected to controlled current and voltage sources, respectively. These sources convert the input signal data into equivalent currents and voltages. Furthermore, a three-phase series RLC (R_p, L_p, C_p) branch is added to simulate the internal impedance of the source. This also offers additional parameters

for model calibration. The line model is executed in discrete mode, with a sampling time of 0.125 ms (i.e., equivalent to 8 kHz sampling frequency).

In addition to simulating the leakage current with the physics-based model of the line, the measured data is also used to calculate the leakage current, i.e., measured leakage current. This allows for direct comparison and quantification of the accuracy of the developed model. Figure 3(a) shows one period (0.02 seconds) of the simulated and measured leakage currents for one of the phases[1]. Figure 3(b) shows the frequency spectrum of the simulated and measured leakage currents for the same period. Although the waveform of the simulated current is similar to the measured, there is a significant discrepancy between them, which is also reflected in the relative magnitudes of the harmonics where we can observe high mismatch for certain components.

We resort to model calibration to improve the alignment between the simulated and measured leakage current. As calibration parameters, we can use R_s, L_s, C_s, and R_i of each segment and R_p, L_p, C_p of the simulated voltage and current input sources. For the 26-segment line, there are a total of 110 calibration parameters. A sensitivity analysis of these parameters revealed that C_s of all branches and the R_p, L_p, C_p are the most important parameters for model calibration. Thus, we reduce the number of calibration parameters to 32. However, manual calibration of these parameters still remains an impractical option. For that reason, we have coupled the physics-based model of the Avegno-Gorduno line with a genetic algorithm (GA). The purpose of the coupling is to perform optimized calibration of the line model. The execution of the coupled algorithm begins with the creation of a random population of size p, with each candidate in the population containing 32 parameters. Where each candidate represents a potential solution to the optimization problem, which is defined as the minimization of the Euclidean distance between the simulated and the measured leakage currents. The GA is an iterative process, where the *survival of the fittest* is the guiding principle. The algorithm performs selection to create a pool of individuals that are then paired for reproduction. The latter procedure consists of two operators, namely crossover and mutation, to produce a new set of individuals. The old and the new populations are evaluated according to their fitness (Euclidean distance) and the best survive while the rest are discarded. These steps are repeated until a preselected number of iterations is reached. The calibration procedure is described in detail in [72]. The best solution is used to update the parameters of the physics-based model of the Avegno-Gorduno line. Figure 4 shows that the calibrated model produces a leakage current that is in good alignment with the measured data.



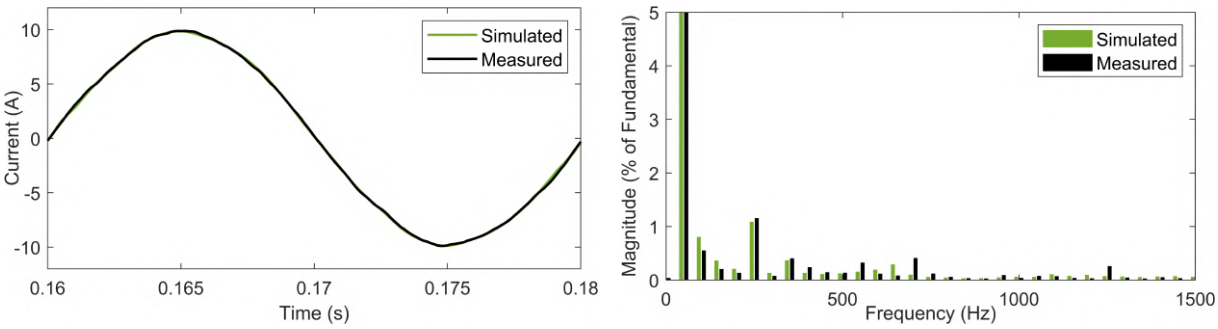
(a) Current (b) FFT

Figure 3: The simulated and measured leakage current without calibration.

Synthetic data generation

After the model is calibrated, we use it to create synthetic data sets, which are then used to train machine learning models. Increasing the capacitance of the model will increase the leakage current, and therefore, it is a suitable parameter to generate faults [29]. We simulate different states of health of

[1] For demonstrative purposes, in this work, we will only show the results for one of the phases.



(a) Current (b) FFT

Figure 4: The simulated and measured leakage current with calibration.

the insulators in a segment by changing the corresponding capacitance while keeping the capacitance of all other segments unchanged. Due to a lack of information on the acceptable amount of leakage current, we assume that all capacitance values up to two times the base values are acceptable, and the corresponding leakage currents are considered as *healthy*. All capacitance values between two and ten times the base values are deemed to cause leakage currents that are of concern to the state of health of the line. These leakage currents are considered as *unhealthy*. Their existence will be used as an indicator of unhealthy insulators that require closer observation and planning for maintenance activities. Furthermore, in order to localize the leaky segment, they are assigned labels according to the index of this segment, i.e., *Seg_01* - *Seg_26*. The data corresponding to healthy lines are labeled a *Seg_00*, which indicates an absence of leaky segments. Therefore, this approach integrates both, fault detection and localization in the same model. It is important to note that the range of accepted leakage current values can be adjusted to reflect the operator's needs and practices for insulator maintenance. To create large data sets, for each segment, we generate a set of random numbers that are uniformly distributed between one and ten and multiply the capacitance of that segment.

2.1.2 Deterministic deep learning methods

One of the highly versatile and powerful techniques in machine learning is the neural network, which has become extremely popular in solving real-life problems in a diverse range of applications. Deep neural networks have been used for healthcare [73], automation [74], agriculture [75] and many more [76]. Furthermore, deep neural networks have been employed for fault detection and diagnostics in smart grids [77] and manufacturing [78]. Their ability to automatically extract features from data and solve classification or regression problems makes them extremely suitable for the detection and diagnosing of abnormal conditions in different systems. We employ four different architectures for addressing the problem of fault detection and localization in transmission lines. These include the simplest feedforward neural networks, recurrent neural networks that are suitable for exploiting temporal relations, convolutional neural networks that are suitable for exploiting spatial relations, and a modified convolutional neural network.

Feedforward neural networks

FNNs are the simplest type of neural network and have neurons in each layer connected to all other neurons in the immediately preceding and succeeding layers. Each of these connections has a weight associated with it, and each neuron has a (typically) nonlinear activation function. The output of the neurons in the i^{th} layer of a feedforward neural network can be expressed as:

$$a^{[i]} = g^{[i]}\left(W^{[i]}a^{[i-1]}\right) \tag{1}$$

where $g^{[i]}(\cdot)$ is the activation function of the neurons in the i^{th} layer, $W^{[i]}$ is the matrix of weights associated with the connections between neurons of i^{th} and $(i-1)^{th}$ layers and $a^{[i-1]}$ is the output of the $(i-1)^{th}$ layer. The weight matrix also optionally contains biases, in which case the output of the preceding layer is concatenated with a column vector of ones before performing the operation in Equation (1). An FNN with L layers performs the above operation recursively on the input to produce the final output. While this is a simple and powerful architecture, it suffers from an exploding number of trainable weights with an increase in a number of layers and size of the input.

We use FNN for the detection and localization of faults in a power line. The FNN architecture consists of an input layer, three fully connected hidden layers with 3000 neurons each, a dropout layer, and a fully connected output layer with 27 neurons, followed by a soft-max layer and a classification layer. The FFT magnitudes of the leakage current signals, generated according to Section 2.1.1, are used as input to the FNN. The FNN training generates a model that can determine the state of health of the insulators in the 26 segments of the Avegno-Gorduno line. If there is no change in the state of health, the model will generate *Seg_00*, and it will generate *Seg_01-26*, denoting the location of the fault.

Recurrent neural networks

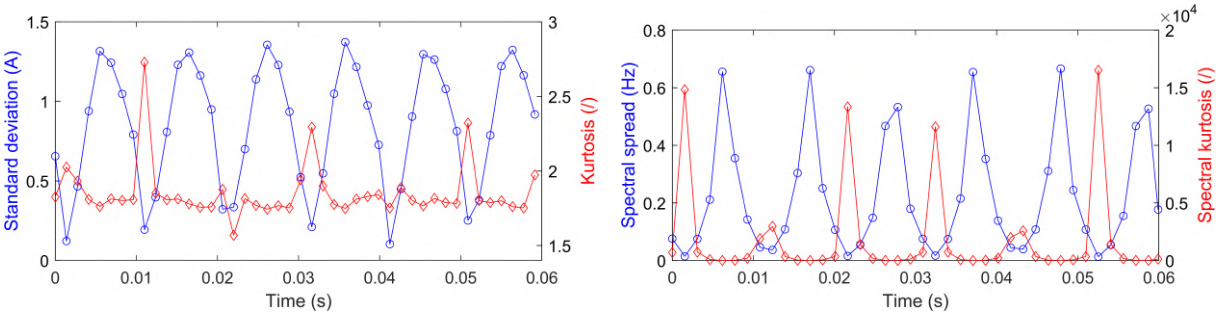
LSTM network is a popular recurrent neural network that is particularly designed to capture temporal relations in the data. LSTM networks are extensively used in language models, time series analysis, and fault detection and diagnostics. The LSTM model consists of several *cells* consisting of inputs, hidden activations, states, and outputs. A cell along with *gates* that perform different computations on the cell quantities constitute a *unit*. The cells of a unit are capable of storing/remembering information over time, while the gates determine the flow of information within and across cells. Several of these units are connected in series to constitute an LSTM model. The different quantities in an LSTM unit are calculated as follows:

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\ \tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\ c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t \\ h_t &= o_t * \sigma_h(c_t) \end{aligned} \tag{2}$$

where i_t, o_t, f_t represent the activations of the input gate, output gate and forget gate of the cell at time t, while h_t, $\tilde{c}_t$ and c_t represent the hidden activation, input activation and state of a cell at time t. The weights and biases are denoted as W_x, U_x, and b_x where x is used to denote the corresponding gate or quantity being computed.

We employ LSTM for the detection and localization of faults in a power line. The LSTM architecture consists of a sequence input layer, one fully connected layer with 200 neurons, one bidirectional LSTM layer, another fully connected layer with 200 neurons, and a fully connected output layer with 27 neurons, followed by a soft-max layer and a classification layer. We use three periods from each of the generated signals and apply summary statistics to extract features from them. This is done either in the time domain or the spectral domain by sliding a window over the signal with length w. In the time domain, at each window, we calculate mean, standard deviation, variance, skewness, and kurtosis, which are then used as inputs. Similarly, in the spectral domain, we calculate spectral spread, spectral centroid, spectral skewness, spectral kurtosis, and spectral flux. Figure 5 shows an example of time and spectral domain features extracted from the same leakage current signal. These features capture different statistical and spectral properties of a signal and can be used to characterize, and hence differentiate different signals. The LSTM models are trained to classify the location of fault by extracting patterns from the evolution of these features over time.

Convolutional neural networks



(a) Time domain (b) Spectral domain

Figure 5: An example of time and spectral domain features for three periods of the leakage current.

CNNs are extremely successful in applications that require identifying patterns in multidimensional data independent of relative location. In addition to this location-agnostic property of CNNs, each layer of such a network consists of several filters of fixed size that contain trainable weights that are shared by the input. Each filter is scanned over the input and a nonlinear activation is performed to obtain the output. While the size of the input can change from less than 10 to thousands, the size of the filters can be chosen according to the desired performance and shared across the data. As a result, such networks can be made extremely deep without a drastic increase in the number of trainable parameters. The output of the neuron in the (j, k) position of the c^{th} channel of the i^{th} layer of a CNN model can be expressed as:

$$a_{j,k}^{[i]} = g^{[i]} \left(\sum_{x=0}^{w-1} \sum_{y=0}^{h-1} w_{x,y}^{[i][c]} a_{j+x,k+y}^{[i-1]} \right) \tag{3}$$

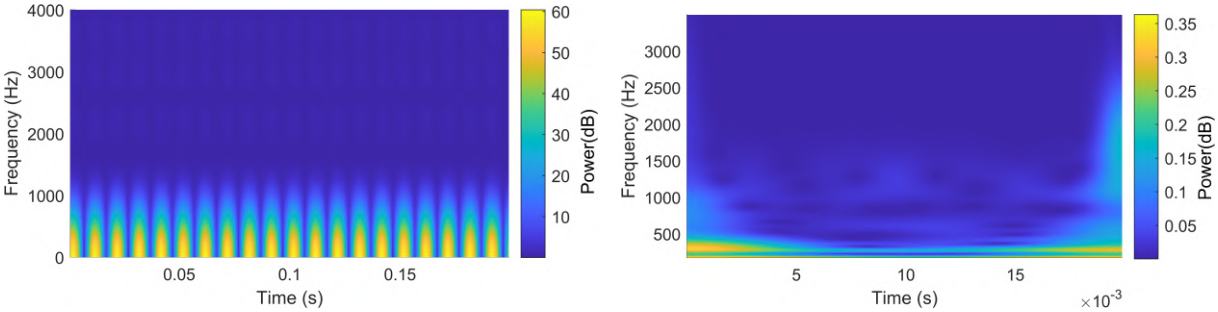
CNNs are extremely popular in computer vision and have been successfully applied for the classification of images, detection of objects in images, and semantic and instance segmentation, among other tasks [79, 80]. These models find applications in diverse fields such as driving [81], robotics [82] and healthcare [83]. CNNs are also often used for fault detection [84, 85], especially when the detection task involves data from multiple sensors. Such networks are capable of extracting the relations between different sensors and efficiently detecting and classifying different patterns according to different types of operating conditions.

We utilize CNNs for the detection and localization of faults in a power line. The CNNs are fed with images, which are based on the time-frequency representations of the leakage current signals. We use two types of images, namely spectrograms, and scalograms, as inputs to the CNN. The spectrogram is a visual representation of the time-varying spectral composition of a signal. It consists of the squared magnitudes of the short-time Fourier transform (STFT) coefficients[2] of the signal, plotted as a function of time and frequency. Similarly, the scalogram is a plot of the magnitudes of the continuous wavelet transform (CWT)[3] of a signal, as a function of time and frequency. Both types of images are generated in the RGB color space with a size that depends on the used CNN. These are images with three color channels that make them 3-D tensors, which are suitable for utilization in CNNs. Here, the images are created either from a single period or ten periods of the leakage current. Figure 6 shows an example of a spectrogram and scalogram for ten and one period, respectively. These images capture the energy content at different frequencies in a signal, and the variations in the signal are reflected in different

[2]STFT is obtained by applying the Fourier transform over a signal using a sliding window [86]. In this work, instead of the squared STFT magnitudes we utilize the absolute values of the magnitudes since our tests show better model performance with the latter. However, for convenience we will use the term spectrograms to refer to the absolute values in the remainder of the paper.

[3]The CWT analysis uses complex wavelet, which slides across the entire signal and extracts the phase and amplitude components associated with the signal [87].

colors. This approach allows the use of CNNs to learn patterns that distinguish signals corresponding to healthy or faulty conditions as well as the location of the fault.



(a) Spectrogram (b) Scalogram

Figure 6: An example of spectrogram and scalogram generated for ten and one period of the leakage current, respectively.

Here, we use pretrained deep neural network architectures including AlexNet [88], SqueezeNet [89], ResNet [90], and Xception [91], which have been trained on the ImageNet data set [92]. We start with these pretrained models and use transfer learning on the spectrogram and scalogram images so as to reduce training time and achieve better accuracy with limited data. The pretrained CNN models are designed for the classification of 1000 object categories. Therefore, the classification layer together with the preceding learnable (fully connected or convolutional) layer are replaced with layers with 27 outputs, which makes them suitable for our 27-class classification problem. Furthermore, we replace the last dropout layer with another with a dropout rate of our choosing. If the selected CNN architecture does not comprise a dropout layer (such as ResNet-18 and Xception), we add one just before the last learnable layer.

Convolutional neural networks with custom layer

The regular CNN is designed to receive images as input data. However, a CNN can be modified to accept signals instead. In this work, we insert a custom layer between the image (input) layer and the first convolutional layer. The custom layer is defined as a log spectrogram layer [93]. The objective of the layer is to obtain log spectrograms of the input signals, by calculating the logarithm of the squared STFT magnitudes. The logarithm boosts the amplitudes with small values that may still contain information that is relevant for the learning process. The custom layer does not carry any learnable parameters (weights) and only computes the log STFT of the input. As such the custom layer can be inserted in any custom or existing CNN architecture (e.g., ResNet, Xception). However, when the log spectrogram layer is used, log STFT is directly applied, i.e., no RGB images are created. Therefore, the first convolutional layer consists of only one channel with size [k_1 x k_2 x *1*], where k_1 x k_2 denotes the size of the log STFT matrix. In addition, the input (image) layer is replaced with a layer of size [n x *1* x *1*], where n is the length (number of points) of the signal. Furthermore, we replace the dropout, the last learnable layer, and the classification layer. Since the aforementioned networks are pretrained with RGB images, here we have two options: 1) use the untrained CNN architectures with an appropriate input layer and first convolutional layer, and 2) adjust the first convolutional layer of a pre-trained model by summing the weights of all three channels of the first convolutional layer. The former approach uses only the architecture and trains the model from scratch, while the latter transforms the three channels into one channel, thus making the architecture suitable for training with all the remaining weights unchanged.

The advantages of a CNN with a custom layer are twofold: 1) The preprocessing steps for the creation of spectrograms are not needed. This prevents any ambiguity associated with pre-processing parameters such as window length and results in a consistent data processing pipeline at the time of training and deployment. Therefore, the possibility of using different preprocessing settings while using

the trained model is nonexistent. Furthermore, the custom layer contributes to a self-contained model and simplifies the pipeline for deployment. 2) The creation of spectrograms within the network reduces the memory needed to store all spectrograms since the network allocates memory only for the current batch. However, the custom layer may increase the training time because it computes the spectrogram every time a signal is used during training.

We have generated a data set with 15000 samples according to the steps mentioned in Section 2.1.1. However, we vary the number of samples that are used to study the performance of the utilized ML methods. The preprocessing and/or feature extraction are performed on the data sets and the data is fed into an ML architecture to train a predictive model. The training is performed on an NVIDIA Quadro P4000 GPU with 8GB of VRAM using the MATLAB environment [94]. Each architecture is trained for a predefined number of epochs with the Adaptive Moment Estimation (Adam) optimizer to generate a trained model. An ensemble of 10 such models are generated with different parameter initializations to study the sensitivity of the architectures and the reproducibility of the results.

2.1.3 Probabilistic deep learning approaches

DNN models built for health monitoring and fault diagnostics of systems can be categorized as solving one of the following two problems: (a) regression, when the model predicts a continuous variable indicating the health of the system, e.g., remaining useful lifetime of a battery or (b) classification, when the model predicts a discrete variable representing the state of the system, e.g., healthy or faulty ball bearings. In this section, we exploit well-established UQ techniques in the deep learning literature to develop a system-agnostic and problem-agnostic framework for UQ of DNN-based monitoring and diagnostics models that are built from digital twins.

Consider a system Ξ for which we wish to develop a monitoring or diagnostics model. This can be a dynamic system, e.g., a chemical refinery or a static system e.g., a bridge. Let $f(x;\theta)$ represent a DNN-based monitoring/diagnostics model that is parameterized in θ, and takes $x \in \mathcal{X}$ as input to produce $y \in \mathcal{Y}$ as output. Here, x can represent measurements of variables of the system that are used to predict the health/state y. In the absence of training data available from the real-life system, a digital twin $\hat{\Xi}$ of the system generates synthetic data: $\tilde{x} \in \tilde{\mathcal{X}}$ and $\tilde{y} \in \tilde{\mathcal{Y}}$. After training the parameters θ of the model with data from $(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}})$, the model is deployed on the real-life system to make predictions with data from $\mathcal{X}$.

Let us consider that $f(x;\theta)$ consists of L layers such that the activation a^l of the l^{th} layer can be expressed as:

$$a^l = g_l(a^{l-1};\theta_l) \tag{4}$$

Here, $g_l(\cdot)$ and θ_l represent the non-linear activation and parameters (weights and biases) of the l^{th} layer, respectively, and the parameters θ of the model are decomposed into layer-wise parameters θ_l. The output of the model can be expressed as:

$$\begin{aligned} y &= f(x;\theta) = a^l \\ &= g_L(a^{L-1};\theta_L) \\ &= g_L(\cdot;\theta_L) \circ g_{L-1}(\cdot;\theta_{L-1}) \circ \cdots \circ g_1(\cdot;\theta_1)(x) \end{aligned}$$

Here, $\circ$ represents the composition operator, i.e., $f \circ g(x) = f(g(x))$. The total uncertainty in the prediction $y_i = f(x_i,\theta)$ of the model arises from the uncertainty in x_i (aleatoric uncertainty) and θ (epistemic uncertainty).

Aleatoric uncertainty

The aleatoric uncertainty in $f(x;\theta)$ arises from the fact that x is a random variable drawn from a distribution $p_x(x)$. Quantifying the impact of this source of uncertainty can be achieved in two approaches, as described next.

Explicit propagation of input uncertainty: In this approach, it is assumed that information regarding the uncertainty in inputs x is available beforehand. Specifically, we assume that the mean μ_x and variance Σ_x for any input x can be obtained to train the model. The objective is to predict the mean and variance of outputs, i.e., μ_y and Σ_y respectively. In order to achieve this, we exploit assumed density filtering (ADF) that can be used to (approximately) propagate the uncertainty through each layer of the model.

Let $a^{0:l}$ represent the activations of the model up to the l^{th} layer. Then the joint probability density function (PDF) of all the activations of the model can be expressed as:

$$p(a^{0:L}) = p(a^0, a^1, ..., a^L) \tag{5}$$

$$= p(a^0)\prod_{l=1}^{L} p(a^l|a^{l-1}) \tag{6}$$

Here, $a^0 = x$ is used to simplify the notation. Equation 6 implies that estimating the joint PDF of all activations involves estimating $p(a^l|a^{l-1}), \forall l \in [1, L]$. The objective of ADF is to tractably approximate this PDF, one layer at a time. To that end, let us assume that $x = a^0 \sim \mathcal{N}(\mu_0, \Sigma_0)$. Then let us approximate the activations of subsequent layers as:

$$p(a^l|a^{l-1}) \approx q(a^l) = \prod_j \mathcal{N}(a_j^l; \mu_j^l, v_j^l) \tag{7}$$

where μ_j^l and v_j^l represent the mean and variance of activation of j^{th} neuron in the l^{th} layer. Note that the activations of individual neurons in a layer conditioned on activations of the previous layer are independent of each other, i.e., $a_j^l \perp a_i^l|a^{l-1}, \forall i \neq j$. Thus, the only approximation in Equation (7) is due to the assumption of normality of individual activations. Substituting this into Eq. (6), we obtain the approximate joint PDF as:

$$\tilde{p}(a^{0:L}) = p(a^0)\prod_{l=1}^{L} q(a^l) \tag{8}$$

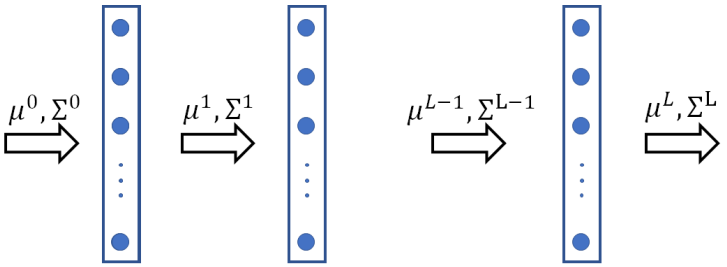
The problem of obtaining the best approximate distribution $q^*(a^l)$ is then posed as minimising the KL-divergence between $\tilde{p}(a^{0:L})$ and $q(a^{0:L})$ [95]. It has been shown that this requires matching the moments of the two PDFs [96], which under the assumption of Gaussian distributions, reduces to estimating the mean and variance as follows:

$$\mu_a^l = \mathbb{E}\left[g_l(a^{l-1};\theta_l)\right] \tag{9}$$

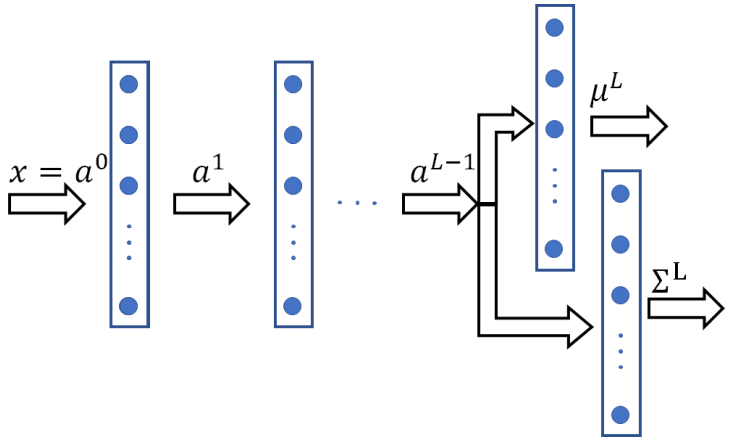
$$v_a^l = \text{var}\left[g_l(a^{l-1};\theta_l)\right] \tag{10}$$

The above approach allows variational uncertainty propagation, i.e., approximating the PDF of activations of each layer in the model and propagating the approximate PDF through all layers. This approach essentially replaces the deterministic layers in the model with their variational counterparts that accept the mean and variance of the inputs and generate the mean and variance of outputs. The variational approximations of common nonlinear activation layers have already been developed and can be found in [95] for regression and classification problems. A more detailed account of ADF for the propagation of input uncertainty can be found in [95].

This approach requires the user to train the network with (μ_x, σ_x^2, y) as data samples. An estimate of the variance of the inputs can be obtained from noise characteristics, sensor resolution, etc. However, in dynamic systems, the variance can change over time and can vary across different entries of a



(a) Explicit propagation of input uncertainty (μ^l and Σ^l are mean and variance of activations of l^{th} layer respectively)



(b) Implicit prediction of output uncertainty (a^l is the activation of l^{th} layer)

Figure 7: Quantification of aleatoric uncertainty (rectangles with filled circles are DNN layers with neurons; arrows are connections between two layers; the layers can be of any type, e.g., fully connected, convolutional)

single sample. In order to address such heteroskedastic signals with time-varying properties, an implicit prediction approach can be useful, which is discussed next.

Implicit prediction of input uncertainty: The impact of aleatoric uncertainty on the outputs of the model can also be quantified by implicitly predicting the variance of outputs. Here, the final layer of the model is redesigned to predict two quantities that are interpreted as the mean and variance of the activations. Specifically, as opposed to a deterministic network that predicts a^L, the probabilistic network predicts μ_a^L and Σ_a^L in the final layer of the model [97]. In the case of regression models, they represent the mean and variance of the continuous-valued predictions. The parameters of the model can be learned by minimizing the negative log-likelihood (NLL), which under the assumption of Gaussian distribution can be expressed as follow [97]:

$$\mathcal{L}_{reg} = \frac{1}{n_y} \sum_{i=1}^{n_y} \frac{1}{2\left(\sigma_{a,i}^L\right)^2} ||y_i - \mu_{a_i^L}||^2 + \frac{1}{2} \log\left(\sigma_{a,i}^L\right)^2 \tag{11}$$

Here, $\mu_{a,i}^L$ and $\left(\sigma_{a,i}^L\right)^2$ represent the predicted mean and variance of activation for the i^{th} node of the final layer respectively, and n_y is the dimension of y.

On the other hand, for classification models, the activations a^L represent the logits of the model that are passed through the softmax function to obtain the class probabilities for the input. In such a scenario, μ_a^L and Σ_a^L represent the mean and variance of logits, with an assumption of Gaussian distribution, one

can draw T samples of logits from $\mathcal{N}(\mu_a^L, \Sigma_a^L)$ as follows:

$$\tilde{a}_t^L = \mu_a^L + \sigma_a^L \cdot \epsilon_t \tag{12}$$

Here, $\tilde{a}_t^L$ represents a sample logit for $t = 1, 2, 3, \ldots, T$ and $\epsilon_t \sim \mathcal{N}(\mathbf{0}, I_{n_y})$. Then, the parameters of the model can be trained to minimize the NLL with the following loss function [97]:

$$\mathcal{L}_{cls} = \frac{1}{T} \sum_{t=1}^{T} \exp\left(\tilde{a}_{t,c}^L - \log \sum_{i=1}^{n_y} \exp(\tilde{a}_{t,i}^L)\right) \tag{13}$$

where $\tilde{a}_{t,i}^L$ represents the i^{th} component of $\tilde{a}_t^L$ and c represents the true class of the input data x. This approach was proposed for predicting output uncertainty in computer vision applications in [97], and has been shown to be a learned loss attenuation formulation of regression and classification problems in the presence of uncertainty. As opposed to ADF, this allows training and deployment of the model without providing any additional information on uncertainty in the inputs.

Epistemic uncertainty

The epistemic uncertainty in $f(x;\theta)$ arises from the fact that the parameters θ are learned from limited training data and are hence probabilistic. Monte Carlo dropout developed in [98] is a widely adopted approach to quantify the epistemic uncertainty of DNNs. This approach performs K forward passes of a test sample x with a random dropout of neurons in the model, and quantifies epistemic uncertainty by estimating the variance of predictions across all K forward passes. This can be achieved as follows:

$$y_k = f(x;\theta^k), k = 1, 2, 3, \ldots, K \tag{14a}$$

$$\mu_y = \frac{1}{K} \sum_{k=1}^{K} y_k \tag{14b}$$

$$\sigma_{ep}^2 = \frac{1}{K-1} \sum_{k=1}^{K} (y_k - \mu_y)^2 \tag{14c}$$

Here, θ^k is obtained by randomly masking neurons in the k^{th} forward pass through the model and σ_{ep}^2 is the prediction variance because of epistemic uncertainty.

Total uncertainty

The calculation of total uncertainty involves summating the aleatoric and epistemic components. However, this calculation can be slightly different depending on the problem solved (regression or classification). In the case of regression, the aleatoric variance can be obtained from explicit propagation or implicit prediction as follows:

$$\sigma_{y|k}^2 = v_{a|k}^L, \qquad \text{or} \qquad \sigma_{y|k}^2 = \left(\sigma_{a|k}^L\right)^2 \tag{15}$$

where $|k$ in the subscript represents the quantities calculated for the k^{th} forward pass. Then, the average aleatoric variance can be obtained as:

$$\sigma_{al}^2 = \frac{1}{K} \sum_{k=1}^{K} \sigma_{y|k}^2 \tag{16}$$

In case of a classification model, we can obtain the aleatoric variance through the explicit and implicit methods as:

$$\sigma_{y|k}^2 = v_{a|k}^L, \qquad \text{or} \qquad \sigma_{y|k}^2 = \text{var}(\text{softmax}(\tilde{a}_k^L)) \tag{17}$$

Note that in the implicit case, every forward pass involves drawing T samples according to Eq. (12).

The epistemic variance for both models can be obtained from Eqs. (14a)-(14c). Finally, the total variance can be obtained as:

$$\sigma^2_{total} = \sigma^2_{al} + \sigma^2_{ep} \tag{18}$$

The framework proposed above is depicted in Figure 8 and is independent of the underlying system and can handle both regression and classification problems.

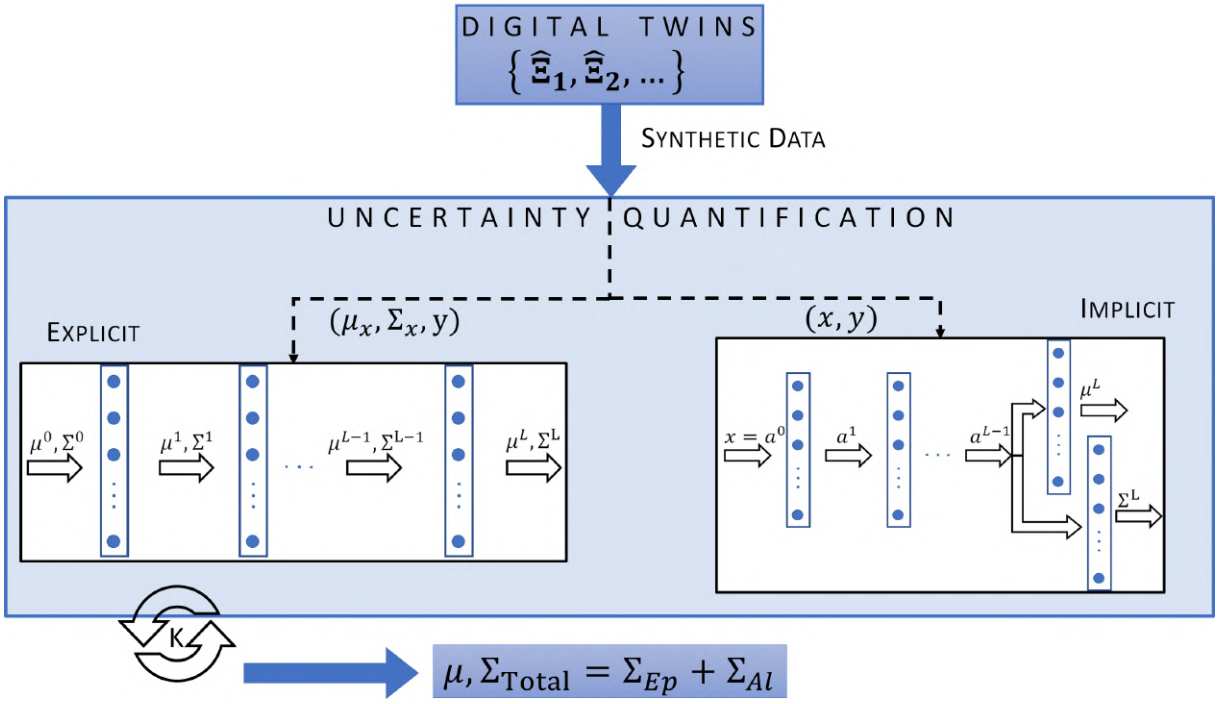


Figure 8: Uncertainty quantification framework for DNN models trained with data generated from digital twins.

2.2 Results for fault power lines fault detection and classification

Here, we present the fault detection and classification results for power lines.

2.2.1 Deterministic machine learning models

Feedforward neural network

The FNN architecture, trained with 5000 samples of the leakage current, shows stable learning behavior, i.e., very low sensitivity to initialization while achieving an accuracy of 95% with a training time of 70 seconds. Increasing the number of samples to 15000 improves the accuracy to 98.6% with a training time of 3.5 minutes. In general, the FNN is easy to train and achieves acceptable accuracy. Figure 9 shows a snapshot of the confusion matrix. The figure reveals that in most misclassifications, the healthy class (*Seg_00*) is misclassified as a faulty class or a fault at a segment is misclassified as a fault at a neighboring segment. In rare cases, a fault at a segment is misclassified as no fault (*Seg_00*). Therefore, the model can be characterized as conservative. We have trained the FNN ten times and observed similar model performance with respect to the training, validation, and test sets for all models.

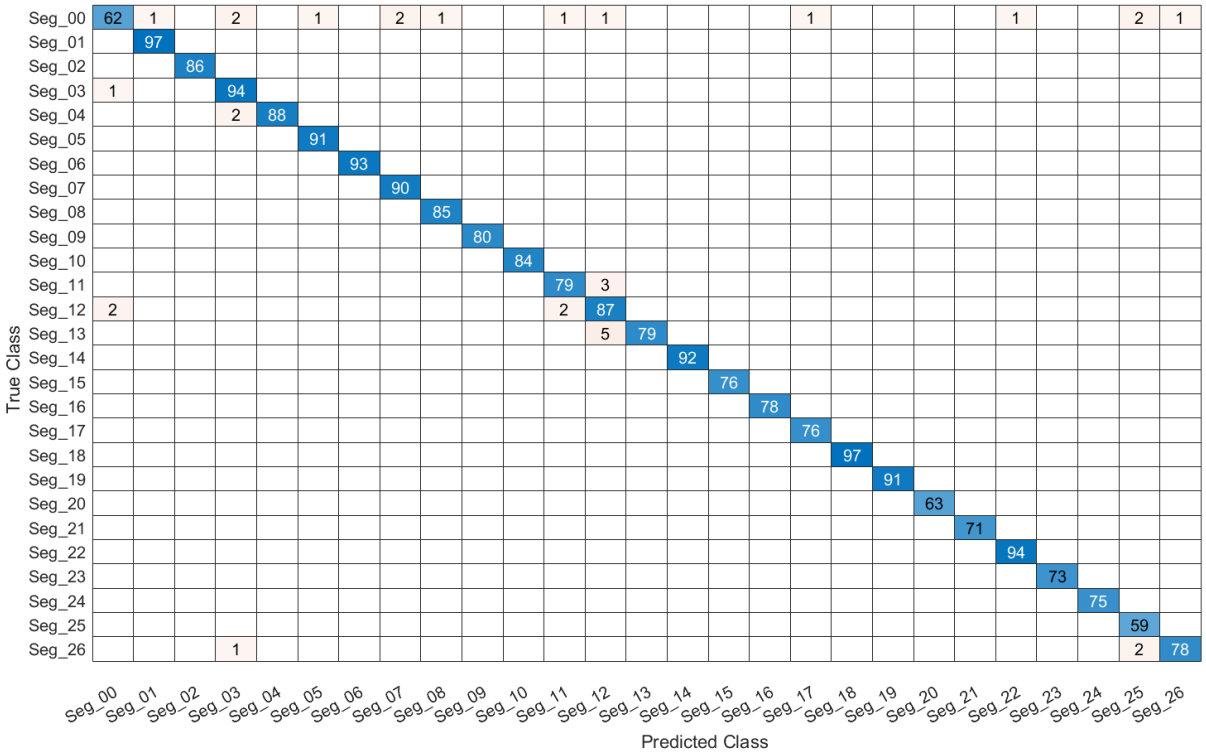


Figure 9: A confusion matrix for the trained FNN model.

Recurrent neural network

The LSTM with the time domain features, when trained with 5000 and 15000 samples, yields an accuracy of 92% and 99% on the test data respectively. When the LSTM is trained with the spectral domain features, the accuracy is smaller, i.e., 88% with 5000 samples and 94.5% with 15000 samples. The training time of the LSTM is substantially higher in comparison to that of FNN. Specifically, on the large data set, the training completes in 174 minutes (3500 epochs) with the time domain features and 295 minutes (8000 epochs) with the spectral domain features. Similarly, as in the case of the FNN, most misclassifications are either a fault at a segment misclassified as a fault at the neighboring segment or a normal state being incorrectly classified as a fault. We have trained the LSTM with time and spectral domain features ten times and observed similar model performance with respect to the training, validation, and test sets for all models.

Convolutional neural network

We evaluated different CNN architectures on the aforementioned sample sizes. We observe that with 5000 samples AlexNet and SqueezeNet obtain accuracy of $\sim$66% and $\sim$80% on the test set, respectively. Furthermore, ResNet-18 and the more complex Xception, although obtain higher accuracy of 95% on the test set, are prone to overfitting (with an accuracy of 100% on the training set). It was observed that larger dropout rates and regularization were not able to address the overfitting challenge. However, the performance and the overfitting are resolved by increasing the training data set size to 15000 samples. We observe an accuracy of 99.6% and 98.3% on the test data set with the spectrograms and scalograms as inputs into the ResNet-18, respectively. The training times are also different; the CNN with spectrograms is trained in 73 minutes (60 epochs) and the CNN with scalograms is trained in 113 minutes (100 epochs). Similar accuracy is achieved by utilizing Xception with significantly higher training times (i.e., 25 hours). It is evident that the spectrograms outperform scalograms for fault detec-

tion and localization with CNN models. Although we have performed numerous tests, it is still possible to improve the outcome by further tuning hyperparameters and CNN architecture selection, as well as by increasing the data set size. In addition to the models with pre-trained weights, we also trained the models from scratch and observed that the training was relatively less stable across different initializations and the accuracy was lower by $\sim$1%. This observation highlights the suitability of pre-trained models combined with transfer learning for fault detection applications. Finally, we have trained the pretrained ResNet-18 with spectrograms and scalograms ten times and observed similar model performance with respect to the training, validation, and test sets for all models.

Convolutional neural network with custom layer

The CNN with custom logarithmic layer, when trained with 5000 samples using ResNet-18, obtains an accuracy of 99.8% on the test set with a training time of 4 minutes. This is similar to the training time of the best FNN model and just a fraction of the training time of the CNN. When trained with 15000 samples, with a training time of 22 minutes, the same architecture achieves an accuracy of 99.9% on the test set, and thus, outperforms all other models. The results, obtained with the pretrained and the untrained CNN architectures are similar. Therefore, the high accuracy achieved with the CNN with a custom layer is not attributed to the transfer learning but instead to the CNN architectures and the log spectrogram layer. We have trained the CNN with a custom layer ten times and observed similar model performance with respect to the training, validation, and test sets for all models.

Figure 10 shows the receiver operating characteristic (ROC) of the best models. The ROC curves are produced with respect to *Seg_00* because most misclassifications are associated with this class. The ROC curves show that the model trained with the CNN with a logarithmic layer outperforms the other models with the highest area under the ROC Curve (AUC). Furthermore, the ROC curves show that all models have similar behavior, i.e., have relatively high true positive rate and low false positive rate for a wide range of decision thresholds.

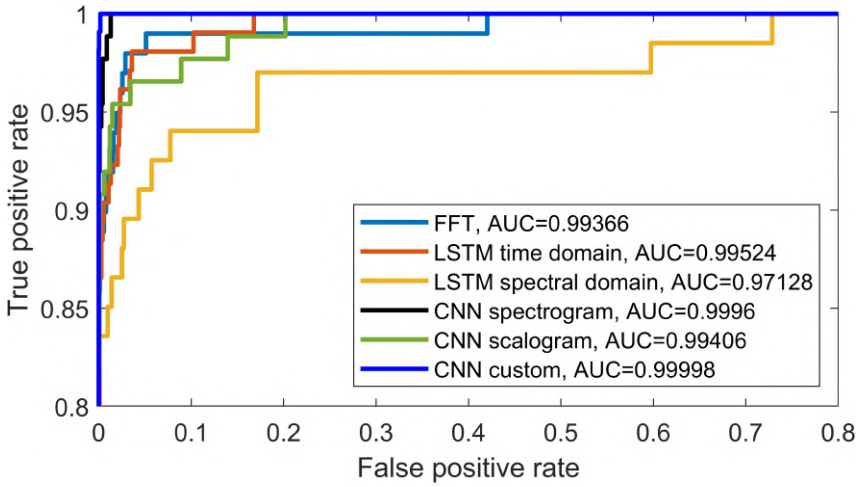


Figure 10: ROC curves for different models.

Discussion

In this section, we briefly discuss some of the challenges and caveats associated with this approach. These challenges pertain specifically to the developed physics-based model:

- Physics-based models are useful in domains where faulty data are not available or are difficult to obtain, which is the case with the application considered in this work. However, building an accurate physics-based model also requires reliable data and accurate measurements that are

not always available. For example, not all substations have GPS-synchronized PMUs that can take measurements with high sampling frequency. Therefore, building such models for every line in a transmission grid may not be possible without significant investments.

- The physics-based model may suffer from instability issues. In this particular case, we have observed instabilities in the current waveforms in the first several periods after a simulation is executed. Therefore, we discard these periods to avoid inaccurate data entering the ML pipeline and hindering the model outcome. Also, the physics-based models are sensitive to the change in parameters and may require additional efforts to calibrate the models after changes in the transmission line.
- We have also observed that the physics-based model is sensitive to the calibration parameters. This results in uncertainty in the data generated by calibrated models, which affects the training of the ML models. Therefore, an ML model trained with the data produced by the physics-based model using one set of calibration parameters might not perform well on the data produced by the physics-based model using a second set of calibration parameters. Here, we perform sensitivity analyses of the best ML model by modifying the calibrated physics-based model parameters. We vary the parameters by different amounts with uniform sampling and then use the new parameters to create a new dataset with the physics-based model. We test the accuracy of the best ML model trained with the original calibration parameters on this new dataset and report the results in Table 2. The table shows that with the increasing degree of changes in parameters, the performance of the model continuously degrades. This is because the new dataset is now drawn from a distribution that is further from the training distribution of the model. Such poor out-of-distribution performance is a universal issue in developing reliable ML models. Therefore, it is very important to investigate this issue in applications that involve coupling a physics-based model with a machine learning model.

  Table 2: Accuracy of the best ML model after random variations in the tuning parameters of the physics-based model.

Variation, v (%)	1	3	5	10
Accuracy (%)	91.59	90.68	88.95	79.16

- We only have current measurements for the reference state of the line, and therefore, we can only calibrate and validate the model for similar states. The synthetic data corresponding to unhealthy states was generated in accordance with the state-of-the-art methods available in the literature. However, these faulty states have not been verified due to a lack of measured data in conjunction with known degraded insulators. This lack of data for faulty scenarios also prevents the training of generative models such as variational autoencoders and generative adversarial networks that could potentially be used to generate more data for different scenarios.

The latter issue can be overcome once the measurements for faulty states and the locations of unhealthy insulators are known. Such data can help in developing more robust physics-based models. Furthermore, such data can also be used to fine-tune (re-train) the deep learning models after they have been trained with synthetic data.

2.2.2 Uncertainty quantification and probabilistic deep learning models

In this section, we present the results of our analyses and highlight the main findings of uncertainty quantification and training probabilistic DNN models.

Performance of uncertainty-aware and deterministic models

The accuracies of the models are summarised in Table 3. The plain and HET models are trained and tested with data generated from one digital twin, specifically $\hat{\Xi}_1$. On the other hand, the ADF models are

trained and tested with mean and variance calculated from 10 digital twins, i.e., $\hat{\Xi}_i, \forall i \in [1, 10]$. Thus, the main difference is that for training plain and HET models, we only use the data of a single instance, while for training the ADF models, we utilize the data from all instances of the twin. It can be observed from Table 3 that for the FC architecture, the deterministic model has the worst accuracy of 79%, followed by the HET model with 87% and ADF model with 98%. The ADF models outperform the Plain and HET models for all architectures. This is an expected outcome since ADF models use more information for training than Plain and HET models. The HET models are sensitive to the type of architecture used, and have the worst performance for convolutional architectures, while outperforming the plain models for FC architectures. It was also observed that the performance of the HET models is very sensitive to model initialization, while the Plain and ADF models are highly robust to initialization. It must be noted here that we compare only the classification accuracy in this section, and do not consider the fact that ADF and HET models provide additional information regarding the confidence of the predictions.

We also observe that although the FC and Conv1d architectures both take raw time series data as input, the latter performs considerably better for the Plain model. This can be attributed to the fact that while the FC architecture looks at relationships between all variables and can be overwhelmed with such relations, the Conv1d architecture extracts patterns relevant to the temporal evolution of the data and can identify important information for classification. The Conv2d architecture, on the other hand, uses spectrograms as input and receives data that has already been processed to extract certain features, i.e., the temporal evolution of spectral content of the signal. As a result, this architecture performs better than the FC and Conv1d architectures for Plain and ADF models.

Table 3: Accuracy of uncertainty-aware and deterministic models

Architecture	Plain	ADF	HET
FC	0.79	0.98	0.87
Conv1d	0.98	0.99	0.45
Conv2d	0.85	0.99	0.72

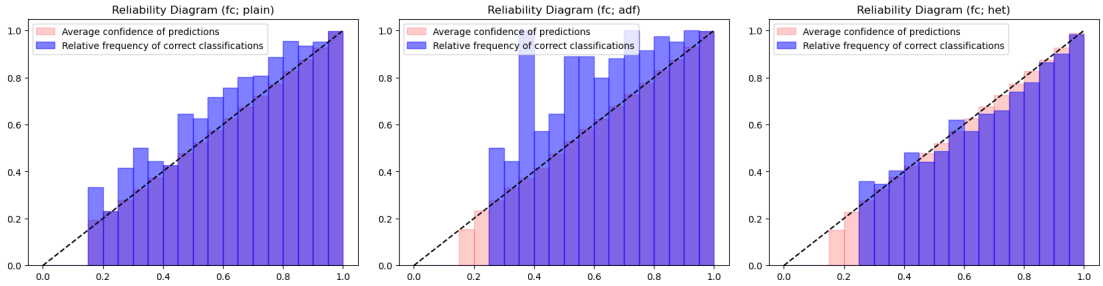
Reliability of uncertainty-aware and deterministic models

The reliability diagrams of all the models are presented in Fig. 11. The reliability diagram compares the predicted probability of a class (pink bars) with the relative frequency of correct classification (blue bars) and is referred to as the calibration of the ML model. A perfectly calibrated model is expected to exhibit a reliability diagram where both the quantities are equal, suggesting that the predictions from the model can reliably be interpreted as class probabilities. A poorly calibrated model, on the other hand, could have confidence greater or less than the accuracy, resulting in over-confident and under-confident models respectively.

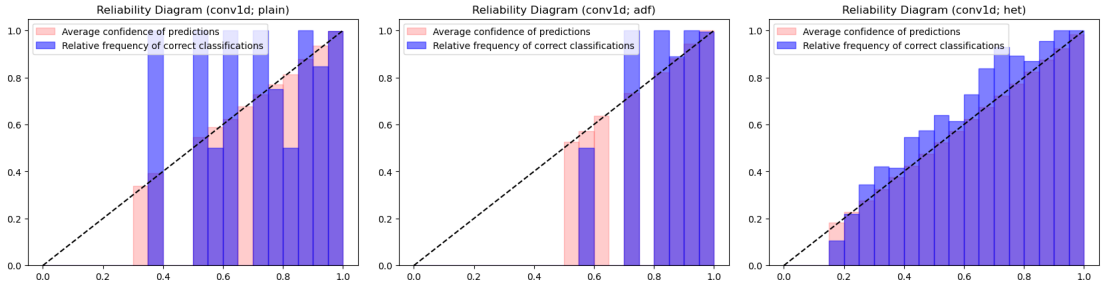
Figure 11 shows that for the FC architecture, the Plain and HET models are relatively well calibrated compared to the ADF model. We also observe that in most of the cases, the ADF model has a lower predicted probability than the frequency of correct classifications, suggesting that it is mostly under-confident. The Conv1d architecture exhibits a poor calibration for Plain and ADF models that are also under-confident in most cases and over-confident in some. This is a useful observation for the case study since it implies that in most cases, the model is not overly confident in its predictions and thus, can be relied upon to make downstream decisions. For the Conv2d architecture, we observe relatively better calibration than the Conv1d counterparts for the Plain and ADF models. The observed miscalibration in the models can be corrected to some extent by employing techniques such as temperature scaling [99] to re-calibrate the models. However, this is beyond the scope of this article and can be explored in future studies to enhance the reliability of the models.

Uncertainty in model predictions

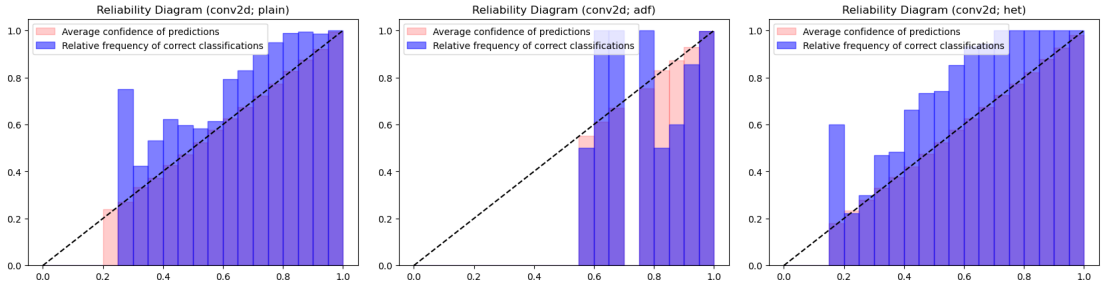
We observed that the epistemic variance of the deterministic models for all architectures did not



(a) FC models



(b) Conv1d models



(c) Conv2d models

Figure 11: Reliability diagrams of plain, ADF and HET models

exhibit any distinct patterns, and were uniformly distributed in a specific range. This can be observed in Fig. 12 which shows the epistemic variance for the Plain model with Conv1d architecture as an illustrative case. A similar observation was also made for the aleatoric variance of the HET models (not shown in a figure). The aleatoric variance of the ADF models exhibited interesting patterns, which are shown in Fig. 13 shows the aleatoric variance of all of the ADF models. Here, the variances are plotted as a function of the scaling factors of capacitance in the digital twin that are used to simulate different healthy and faulty conditions. First, we observe that the variance of the FC model is more than that of the Conv1d model, which is larger than that of the Conv2d model. We also observe that all models exhibit a sudden increase in variance as the scaling factor increases beyond 2, representing faulty data. This suggests that the model is much more confident in the classification of data for the healthy system than for the faulty system. It is noteworthy that the physics-based models are tuned with a healthy system, and hence are expected to exhibit better fidelity in this region compared to the faulty region (scaling factor > 2). Although this information is not provided at the time of training, the ADF model is interestingly able to discover the effect of this difference in the data on its own, and thus reflect it in its predicted variances.

Furthermore, we observe that for the faulty region, the variance of predictions increases with the scaling factor. The model thus becomes less confident as the scaling factor increases. This behavior

is true for the FC and Conv1d models, while the Conv2d model appears to be insensitive to the scaling factor. The ADF models are therefore able to provide the most amount of information regarding their predictions and are also sensitive to the digital twin tuning and data generation processes. We next look at the generalisability of the models to data generated from different digital twins.

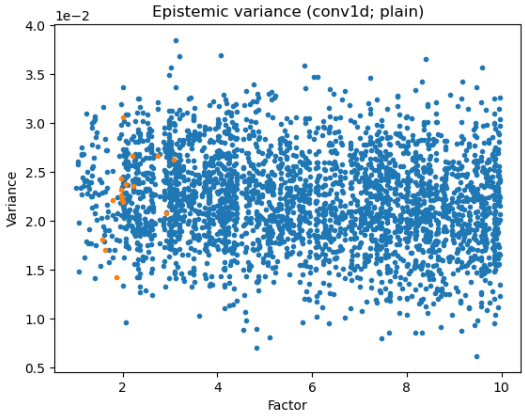
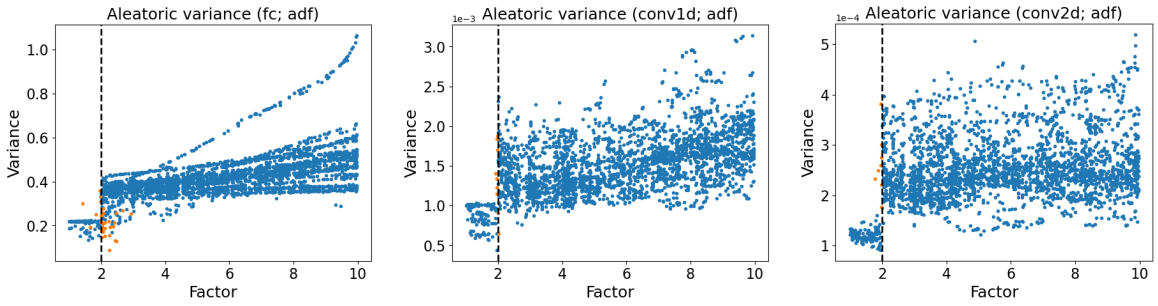


Figure 12: Epistemic variance of Plain Conv1d model (blue points are correctly classified samples; orange points are incorrectly classified samples)



(a) FC models (b) Conv1d models (c) Conv2d models

Figure 13: Aleatoric uncertainty (variance) of ADF models (blue points are correctly classified samples; orange points are incorrectly classified samples; the dashed black line separates healthy samples from faulty ones)

Generalisability of uncertainty-aware and deterministic models

The Plain models trained with data from $\hat{\Xi}_1$, when evaluated on the data generated from any other twin, i.e., $\{\hat{\Xi}\}_{i\neq 1}$ result in a classification accuracy of 0.03 for all architectures. This is equivalent to performing a random guess with 27 classes and indicates no transferability of the features learned by the models to different datasets. It must be noted that all models considered here are small networks with less than $100,000$ parameters, and have dropout layers after all layers except the last prediction layers. As a result, this poor generalisability cannot be attributed to over-fitting by the model and is a direct consequence of the variability in the data generated by different twins. For this case study, training deterministic models on data generated by digital twins is therefore as good as making a naive guess and cannot be relied upon to assess the state of health of the real-life system. The HET models also exhibit similar behavior with a classification accuracy of 0.04 for different datasets.

The ADF models, on the other hand, exhibit better generalisability across different datasets. We trained ADF models with mean and variance calculated from 7 datasets randomly chosen from the 10 available datasets and recorded the performance on the test subset. We then tested this model on the

mean and variance calculated with the remaining 3 datasets. We observed that the absolute performance of the ADF models trained with 7 datasets is the same as that trained with 10 datasets, and is comparable with that reported in Table 3. Furthermore, the difference in their performance on the remaining 3 datasets was found to be in the range $[0.05, 0.15]$, still providing more than 85% classification accuracy. This decline in performance was further observed to be sensitive to different initializations and to the type of architecture used. This highlights the value of providing additional information on input data variance at the time of training and using ADF to train the models. However, real-life implementation of ADF models requires that the variance of inputs be provided at the time of deployment, which might not be available in all applications. We provide further discussion along these lines in the following section.

Discussion

In this section, we discuss some key properties of the probabilistic models and the value of using multiple instances of digital twins.

Information fusion from multiple twins: In the previous section, we train the Plain and HET models with data generation from one digital twin, specifically $\hat{\Xi}_1$, as opposed to the ADF models that are trained with data processed from all digital twins, i.e., $\{\hat{\Xi}_i\}_{i=1}^{10}$. The ADF models therefore see the average (and variance) of the the data generated from $\hat{\Xi}_i$, and it is interesting to investigate if this information fusion from multiple digital twins can also improve the performance of the Plain and HET models. We train Plain and HET models with the mean of data from all digital twins and observe that the models perform comparably to the ADF model for all architectures. This suggests that it is indeed valuable to fuse information from multiple instances of the digital twins to train DNN models to achieve better performance. However, it is also important to provide similarly averaged data at the time of deployment of the model.

Deployment of ADF models in real life: We observe that ADF models provide the best performance as well as insights into the relationship between variance and the data generation process. However, they also require more data to calculate the mean and variance that are used as inputs to the model. While this is possible with digital twins as shown in the previous section, obtaining this information from the real-life system is important to use ADF models in practice. One can utilize measurements over a longer time, divide the data into multiple sets, and estimate the mean and variance from the sets. Furthermore, it is possible to employ sensor noise characteristics to obtain an estimate of the variance in measurements. This is commonly adopted in computer vision applications where the camera resolution can be used to estimate the uncertainty in the pixel values of images.

We tested the performance of ADF models trained with a fixed value of variance = 0.53, which is the average value of the variance across all samples. We observe that the performance of ADF models drops only marginally (by $\sim 2\%$). We also observe that the dependence of the prediction variance on scaling factors used to simulate faulty data also diminishes for the faulty data. On the other hand, the difference between prediction variances for healthy and faulty data is still observed. The models thus lose both performance and some information related to the variance of predictions. Therefore, ensuring that these models are ready for deployment in real-life systems still remains an open question.

3 Power transmission insulators

In this section, we adopt a computer vision-based approach to aerial images collected by drones to detect healthy and faulty components. We rely on two methods to perform fault diagnostics. First, we use a pure object detection-based approach wherein the images are processed by one model, and the different objects of interest are predicted end-to-end. However, this approach can suffer from class

imbalance and size imbalance. Therefore, in the second approach, we use a two-stage approach that includes object detection in the first stage and anomaly detection in the second stage.

3.1 Methods for insulator fault detection and classification

Here, we present the method we use for fault detection and classification of transmission line insulators.

3.1.1 Object detection-based approach

Our primary focus is on incipient fault detection, with an extension to multiple asset inspection. Inspection of insulators can be achieved by (a) locating the insulators and classifying them as healthy or faulty (fault detection), or (b) locating all disks and insulators and identifying healthy and faulty insulators depending on the type of disks, i.e., healthy, flashed, or broken, detected in an insulator (fault detection and diagnostics). We adopt the latter approach that allows us to train multi-object detection models end-to-end while also providing information on the nature of faults. As discussed earlier, the detection of these faults has not been considered in the literature, and to the best of our knowledge this is the first work that addresses this gap. We also investigate if additional objects of interest, specifically Stockbridge dampers and bird nests can be identified in aerial images.

Object detection tasks

We formulate three object detection tasks to achieve the above objective and examine if object detection models can learn to perform these tasks.

1. *Task 1: Insulator detection* – This task involves the detection of insulators and is the simplest of the three tasks considered in this article. This task does not perform fault detection or diagnostics and only serves as a baseline task to compare the performance of different object detection models.
2. *Task 2: Incipient fault detection* – This task involves the detection of the insulator and three types of disks, i.e., healthy, flashed, and broken, resulting in a 4-class object detection problem. In contrast to the 2-class (healthy or faulty insulators) object detection problem, this formulation has two advantages. First, it treats different types of disk damages in a systematic manner instead of aggregating different categories of faults in one class (which can confuse the model). As a result, it provides better performance compared to the 2-class problem. Second, it provides information about the nature of the fault (fault diagnostics) in the insulator, which cannot be obtained with the 2-class formulation.
3. *Task 3: Multiple object detection* – In order to further exploit the potential of object detection models, we perform detection of multiple objects of interest, specifically insulators, disks (healthy, flashed, and broken), Stockbridge dampers, and bird nests, resulting in six object classes. Although this approach can be extended to any number of object classes, we consider only six objects and leave the inclusion of other objects such as corona rings of insulators and spacers for future work.

Object detection models

In order to learn the above three tasks, we consider four types of object detection models popular in computer vision. The first model is Faster RCNN (referred to hereinafter as FRCNN), which is one of the earliest deep neural network-based object detection models and has been used in the early works for the detection of insulators from aerial images in multiple articles [36, 100, 101]. It is a two-stage model that uses a region proposal network (RPN) to identify potential object locations in the first stage, followed by another network to adjust the proposed locations and make final predictions in the second stage.

The FRCNN model suffers from the fact that most of the locations proposed by the RPN do not contain an object, which results in a severe class imbalance problem. To tackle this challenge, the RetinaNet model [102] adjusts the loss function to account for the imbalance explicitly, and is the second model considered in this article. While FRCNN and RetinaNet are two-stage object detection models, the third is a single-stage model, called the Fully Convolutional One-Stage detection model (FCOS). FCOS is a relatively recent detection model [103] and is a proposal-free model that was shown to perform better than its contemporaries. To further increase the exhaustiveness of the study, we include the Single-Shot multi-box Detector (SSD) model, which has been used in the literature for detecting insulators. This article considers the FRCNN, RetinaNet, and FCOS with ResNet50 as the backbone, and SSD with VGG16 as the backbone. The last model is the fifth version of You Only Look Once models, i.e., YOLOv5 which comes from a lineage of highly successful one-stage YOLO models. YOLOv5 performs a series of pre-processing and post-processing steps along with multi-scale detection [104], and has also been shown to outperform several other models like the FRCNN and RetinaNet. These models are very popular and successful in computer vision and domain-specific applications and cover a wide spectrum of modeling approaches including two-stage and one-stage models, imbalance addressing loss function, fully convolutional architecture, and extensive data augmentation models. We do not consider the recent vision transformer-based object detection models with a very different architecture than convolutional models because of the limited size of our datasets.

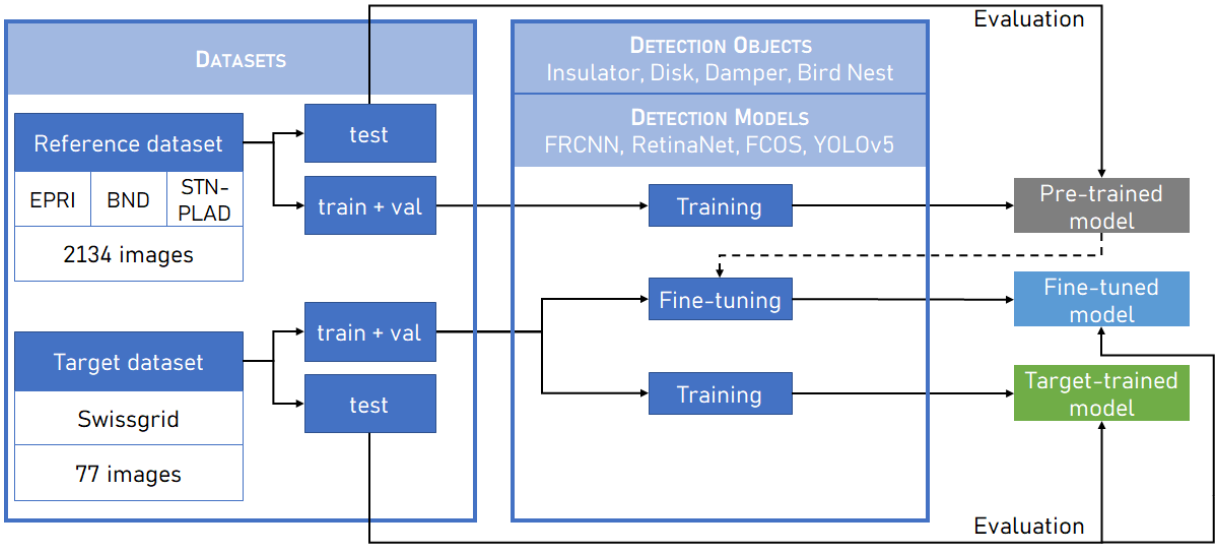


Figure 14: Insulator fault detection and diagnostics: datasets, objects of interest, object detection models, and training methods. Dashed lines represent that the pre-trained models are used as the starting point for fine-tuning. Sensitivity analysis of the fine-tuned model is also performed with different fractions of data used for fine-tuning.

Datasets

We use two aerial image datasets to train object detection models for the above tasks. The first dataset, referred to as the reference dataset, consists of 2134 images collected from three openly available repositories [105, 106, 107]. The second dataset, which we refer to as the target dataset is a very small dataset that consists of 77 images obtained from Swissgrid AG. We perform training on the reference and target datasets separately, which allows us to evaluate the performance of the models in data abundance and data scarcity scenarios. We refer to these models as pre-trained and target-trained models, respectively. However, deep neural network models can be unreliable under data scarcity [108]. Therefore, we also use the model trained with the reference dataset and fine-tune it with the target

dataset. In doing so, the features learned from the reference dataset (which has relatively larger sizes and variations) are used as a starting point to fine-tune the model and adapt it to the target dataset. This allows us to investigate the fine-tuned models' performance with data scarcity, and quantify the value of transfer learning on the models' performance. We also conduct a sensitivity analysis by changing the fraction of the target dataset used to obtain the fine-tuned model. This allows us to study the impact of training dataset size used for transfer learning on the performance of fine-tuned models. Such a scientific approach adopted here examines the value of transfer learning and investigates the sensitivity of the models to dataset size (for insulator fault diagnostics), and is reported for the first time in this article. Figure 14 presents the datasets, objects of interest, models, and training procedures used in the insulator fault detection and diagnostics.

The reference dataset is used to pre-train the object detection models and is a combination of (1) Insulator Defect Image Dataset (IDID) [105, 109], (2) bird nest detection dataset (BND) [106, 110], and (3) Power Line Assets Detection (STN-PLAD) dataset [107]. IDID has 1600 images at varying resolutions and the median resolution of images is 4400×3008. Most images contain only one insulator and a few contain two insulators. IDID shows a considerable variation in the color of insulator disks (brown, black, white, and gray) and in the background and provides the ground truth bounding boxes for insulators, healthy disks, broken disks, and flashed disks. BND consists of 401 images of transmission towers with bird nests and ground truth bounding boxes. The median resolution of the images in this dataset is 5472×3078. All insulators in BND are light green, and the background in most images contains crops or grasslands with a few images having small houses and buildings. In contrast to IDID, the BND has images captured from a relatively greater distance and has two to six insulators in most images. These images also contain insulators and Stockbridge dampers. However, BND does not provide these objects' ground truth bounding boxes. STN-PLAD consists of 133 images of transmission towers with insulators and Stockbridge dampers. The median resolution of images in this dataset is 4048×3040. This dataset has little to no vegetation in the background and consists mostly of dry patches of flatland and mountains. The insulators and dampers are both white in color and consistent in shape. The images in STN-PLAD contain two to four insulators around ten dampers and no bird nests. STN-PLAD also provides the ground truth bounding boxes for the insulators and dampers. All three datasets have images taken with good light exposure and weather conditions and have a single insulator type (ceramic or polymer). However, the combined dataset has a wide variability in the background and foreground.

Owing to the different sources and purposes of the datasets, they contain ground truths for only a subset of all objects of interest in the images. We manually create the ground truth bounding boxes for BND insulators, nests, and dampers with the MATLAB ImageLabeler app [94]. The ground truth for disks has not been generated for BND and STN-PLAD and, as a result, the images from IDID are the only ones with ground truth for the disks. However, since IDID constitutes roughly 75% of the reference dataset, this should not pose a significant threat. This issue of incomplete labels and its impact on performance remains a challenge that can be addressed in the future. The characteristics of the reference dataset are summarised in Table 4, segregated according to the source dataset. It can be observed from Table 4 that the three datasets combined have a considerable number of objects of each class. Specifically, the resulting reference dataset contains 2647 insulators, 13364 healthy disks, 2564 flashed disks, 1180 broken disks, 322 bird nests, and 2536 Stockbridge dampers. In addition, the variations in backgrounds, colors of dampers and insulator disks, distance from the objects, and camera orientation provide a rich dataset for training the object detection models.

The target dataset considered in this work consists of 77 images provided by the Swiss transmission system operator Swissgrid AG. This dataset's background and foreground features differ from those in the reference dataset. In order to perform transfer learning and evaluate the performance of fine-tuned models, we manually label the insulators, disks, and dampers in this dataset with the LabelImg app [111], resulting in 279 insulators, 3706 healthy disks, 64 flashed disks, 76 dampers, and 23 bird nests. This dataset does not contain broken disks.

Table 4: Source-wise properties of images in the reference dataset (median resolution presented as width [px] $\times$ height [px]).

Property	Dataset		
	IDID [105]	BND [106]	STN-PLAD [107]
#Images	1600	401	133
Resolution	4400×3008	5472×3078	4048×3040
#Insulator	1804	531	312
#Healthy Disk	13364	0	0
#Flashed Disk	2564	0	0
#Broken Disk	1180	0	0
#Bird Nest	0	322	0
#Damper	0	1031	1505

Experimental setup

We resize all the images to $1000 \times 1000 \times 3$. The datasets are divided into subsets for training, validation, and testing of the models. We use random horizontal flipping to train the FRCNN, RetinaNet, and FCOS models for data augmentation. We observed that including more augmentation methods such as color jitter and random cropping did not result in a noticeable improvement in performance for the three models. The YOLOv5 model is inherently trained with a host of data augmentation techniques and no additional augmentation methods are included in the experiments. All our experiments are performed with PyTorch. The built-in models for FRCNN, RetinaNet, and FCOS in PyTorch are used, while the official repository of YOLOv5 is used. The total number of FRCNN, RetinaNet, FCOS, and YOLOv5 parameters are 41.29M, 32.16M, 32.06M, and 46.5M, respectively. The number of trainable parameters for pre-training is fixed, while this number varies for fine-tuning depending on the freezing of different modules of the models. All models, i.e., pre-training and fine-tuning, are trained via stochastic gradient descent optimizer with default parameters for 300 epochs. A batch size of 32 is used for training FRCNN, RetinaNet, and FCOS, while YOLOv5 is trained with the default setting of 64 images for backpropagation. The training and evaluation are performed on a workstation with NVIDIA RTX A6000 GPU and 128 GB of RAM. In order to evaluate the performance of the models, the standard metric of mean average precision (mAP) is adopted, which can be obtained as follows:

$$\begin{aligned} \text{AP}_o &= \int \tilde{p}_o(r_o) dr_o \\ \text{mAP} &= \frac{1}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} \text{AP}_o \end{aligned}$$

Here, AP_o is the *average precision* for object o and is equal to the area under the interpolated precision-recall curve $\tilde{p}_o(r_o)$ for object o. The mAP is obtained as the average of AP for all objects in the set $\mathcal{O}$. We report the mAP_{50} for all the models, in which case $\text{AP}_o = \tilde{p}_o(r_o)$ is calculated at a detection threshold of 0.5. For the sake of presentation, we use mAP and mAP_{50} interchangeably in the rest of the article, while always referring to mAP_{50}. We also use mAP to refer to the AP of individual objects.

With the above setup, we first train detection models using only the target dataset. A (train, val, test) split of $(0.7, 0.0, 0.3)$ is used for training FRCNN, RetinaNet, and FCOS models, while $(0.7, 0.1, 0.2)$ is used for training YOLOv5. We then train detection models with the reference dataset with the above split to obtain the pre-trained models. These models are finally fine-tuned with a (train, val, test) split of $(0.3, 0.0, 0.7)$ for FRCNN, RetinaNet, and FCOS, while $(0.3, 0.1, 0.6)$ for YOLOv5 on the target dataset to obtain the fine-tuned models. We adopt this process for all three detection tasks, resulting in 18 models. Finally, we vary the training subset size in the set $\{0.1, 0.3, 0.7\}$ and record the performance of the fine-tuned models.

3.1.2 Object detection and anomaly detection-based approach

We address the problem of insulator fault detection, i.e., identifying the presence of faults in insulators. We consider two types of faults, i.e., flashed and broken disks, which have not been extensively studied in the literature. In recent work, this problem has proved elusive with an object detection-based approach in data-scarce scenarios [112]. Specifically, state-of-the-art object detection models perform very well for detecting insulators (which have high frequency and large size), relatively poorer for healthy disks (very high frequency but small size), and very poor for flashed disks (very low frequency and small size) [112]. As discussed above, the class imbalance challenge cannot be addressed by collecting more images. Further, size imbalance is inherent to the application, and cannot be avoided. This motivates the need for a methodologically different approach to differentiate between healthy and faulty disks reliably. To that end, we investigate the source of poor performance for the flashed disks and observe that classification errors are the primary contributor to the poor performance, rather than localization errors. This implies that the object detection models are capable of accurately localizing disks and only encounter difficulty in correctly distinguishing between the healthy and faulty ones. This naturally leads us to investigate whether a second model dedicated to classifying healthy and faulty disks can benefit the fault detection task.

In this approach, we adopt a two-stage approach that utilizes two models. The first model is an object detector that localizes insulators and disks, and the second model is an anomaly detector that performs fault detection on the extracted disks. Finally, a deterministic post-processing step can identify unhealthy insulators by mapping flashed and broken disks to their source insulators. This approach is presented in Figure 15.

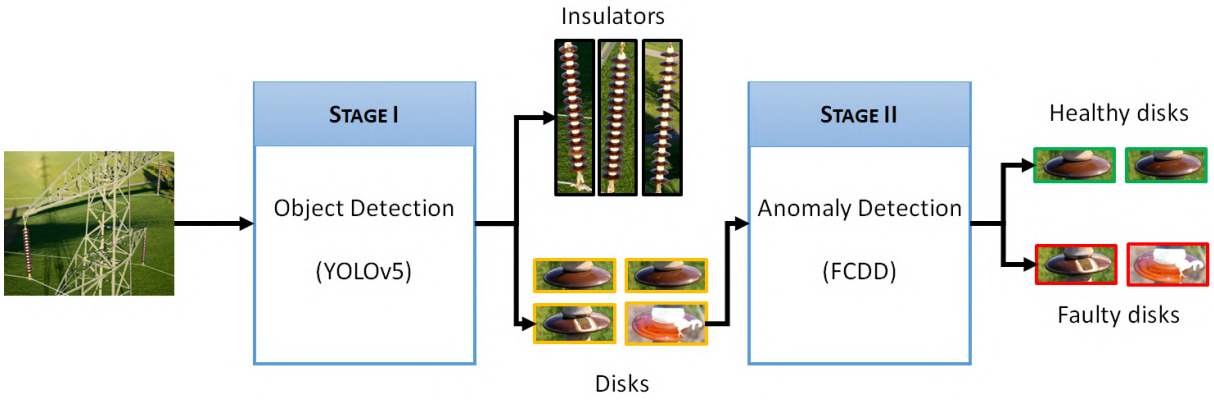


Figure 15: Two-stage insulator fault detection with object detection and anomaly detection. The texts in parentheses represent the particular type of model used in this work.

A two-stage approach divides the fault detection task, and thus, makes it easier for the individual models to perform better. Specifically, in the first stage, a 2-class object detection model will suffice instead of a 4-class model in a purely object detection-based approach. In the second stage, since the samples consist of only extracted images of accurately localized disks, one has to deal with very little background compared to the first. This is evident from Figure 16, which shows sample aerial images of insulators, and extracted images of disks. As a result, one can expect that the two-stage approach will perform better than a one-stage approach.

The first stage of the proposed approach involves training a reliable object detection model for the localization of insulators and disks, which has already been performed in [112]. Therefore, this work focuses only on developing the second stage and building a model to differentiate between different types of disks. This stage will use only images of disks and thus be free from size imbalance. However, the

(a) Aerial images of insulators (b) Extracted images of disks

Figure 16: Aerial images of insulators and extracted images of disks. The extracted images contain very little background compared to aerial images.

class imbalance between healthy and faulty disks persists and must be carefully addressed. The problem associated with the second stage is detecting faulty disks, which occur very infrequently in a dataset with many healthy disks. This naturally lends itself to an anomaly detection problem. Anomaly detection aims to detect rarely occurring data points (anomalies) that deviate significantly from frequently observed normal data points [113]. Anomaly detection is used in several applications such as blockchain networks [114], financial fraud detection [115], and time series analysis [116]. This approach is also adopted for fault detection in marine engineering [117], robotics [118], electrical engineering [119], autonomous driving [120] and photovoltaics [121].

There are several approaches for anomaly detection for different types of data [122]. In this work, we are interested in anomaly detection with images of disks, and thus, adopt a deep learning-based approach to address this problem. A detailed taxonomy and review of the algorithms for anomaly detection using deep learning is presented in [106]. Deep learning-based methods often adopt autoencoders for image [123] and video [124] anomaly detection, which are trained on normal data points. Since such a model only sees normal data points during training, it is expected to exhibit high errors in reconstructing an anomalous data point, thereby allowing anomaly detection by monitoring the reconstruction error of the model. While such models are very popular, they do not offer a way to include any prior information about anomalies during training. However, in the case of detecting faulty disks, one has prior information about the flash-over patterns and shapes of broken disks, which could be leveraged to improve the model's performance. We therefore employ a state-of-the-art anomaly detection method that allows one to use the patterns observed in anomalous data points during training. In addition, the model can also provide explanations for its predictions, which can benefit its adoption by the industry.

Fully convolutional data description

The fully convolutional data description (FCDD) is a state-of-the-art anomaly detection approach [125] that has been demonstrated to provide excellent performance on standard datasets, including ImageNet and MVTec-AD, which is a dataset for detecting defects in manufacturing [126]. FCDD is a one-class classification model, that performs anomaly detection by mapping all normal data points in the vicinity of a center $\mathbf{c}$ in the output space, which results in the anomalies being mapped away from $\mathbf{c}$. FCDD is based on the hypersphere classifier, which employs the following loss function:

$$\mathcal{L}_{HSC} = \frac{1}{N}\sum_{i=1}^{N}\Big[(1-y_i)\,\tilde{h}\left(f\left(x_i;\theta\right)-\mathbf{c}\right) - y_i \log\left(1-\exp\left(-\tilde{h}\left(f\left(x_i;\theta\right)-\mathbf{c}\right)\right)\right)\Big] \tag{19}$$

where, $x_i \in \mathbb{R}^{c\times h\times w}$ represents the i^{th} input, $y_i \in \{1,0\}$ represents the i^{th} target such that $y_i = 1$ for anomalous data points, and $y_i = 0$ for normal data points, N represents the number of data points, $f(\cdot;\theta)$ represents the neural network parameterised in θ, and $\tilde{h}(\cdot)$ represents the pseudo-Huber loss, defined as: $\tilde{h}(x) = \sqrt{||x||_2^2 + 1} - 1$. FCDD uses a fully convolutional architecture for the neural network $f(\cdot;\theta)$, which transforms the input $x_i \in \mathbb{R}^{c\times h\times w}$ to feature $z_i \in \mathbb{R}^{u\times v}$. In addition, the center of the hypersphere

is set to the bias of the last layer of f, so that the FCDD loss can be expressed as:

$$\mathcal{L}_{FCDD} = \frac{1}{N}\sum_{i=1}^{N}\Bigg[(1-y_i)\,\frac{1}{u\cdot v}||\tilde{h}(z_i)\,||_1 \\ - y_i \log\left(1-\exp\left(-\frac{1}{u\cdot v}||\tilde{h}(z_i)\,||_1\right)\right)\Bigg]. \tag{20}$$

The quantity $\tilde{h}(z_i)$ is the pseudo-Huber loss of the features. This loss is summed for all entries of $\tilde{h}(z_i)$ and then normalized with respect to the number of entries, which provides a normalized measure of the deviation of the feature from the center. The above objective minimizes the deviation for normal data points ($y_i = 0$) and maximizes the deviation for anomalous data points ($y_i = 1$). The features of a trained FCDD model can be considered as a heatmap of the input, with anomalous regions exhibiting higher values, and normal regions exhibiting lower values. In general, the features' dimension is lower than that of the input of a convolutional network, i.e., here $u < h, v < w$. In order to map this low-resolution heatmap to the original input image, FCDD further performs a deterministic upsampling of z with a Gaussian kernel [125].

In the above formulation, the model takes only the images and labels, i.e., normal or anomalous, and learns a mapping that highlights the anomalous regions of an image. When anomalous images are used for training in this setting, information about patterns observed in faulty images is implicitly provided to the model. It is also possible to explicitly incorporate prior knowledge at the time of training, which can be achieved by modifying the FCDD loss to allow semi-supervised training as follows [125]:

$$\mathcal{L}^{SS}_{FCDD} = \frac{1}{N}\sum_{i=1}^{N}\Bigg[\frac{1}{w\cdot h}\sum_{j=1}^{w\cdot h}\left(1-y_{i,j}\right)\tilde{z}_{i,j} \\ - \log\left(1-\exp\left(-\frac{1}{w\cdot h}\sum_{j=1}^{w\cdot h} y_{i,j}\tilde{z}_{i,j}\right)\right)\Bigg] \tag{21}$$

where, $y_i \in \{1,0\}^{h\times w}$ is the ground truth map for the i^{th} image, and contains 1 for all pixels exhibiting an anomaly, and $\tilde{z} \in \mathbb{R}^{h\times w}$ represents the up-sampled features. The above loss function has been shown to result in better performing models [125].

In this work, we use FCDD in both formulations (Eqs. (20) and (21)) to differentiate between healthy and faulty disks. In the first formulation, learning from only normal samples, i.e., without including any anomalies at the time of training is possible. It is also possible to use images of any object other than disks as anomalous samples, or employ an outlier exposure algorithm to synthetically generate anomalies, both of which improve the model's performance [125]. Finally, it is possible to include real anomalies (without any ground truth anomaly maps) to minimize $\mathcal{L}_{FCDD}$. In all of these scenarios, either no *real anomalies* are used during training, or the ground truth anomaly maps of real anomalies are not provided, resulting in unsupervised training. We perform unsupervised training with (i) no faulty disks and (ii) real faulty disks without anomaly maps. In addition, we also perform semi-supervised training by minimizing $\mathcal{L}^{SS}_{FCDD}$, wherein we provide the ground truth anomaly masks during training.

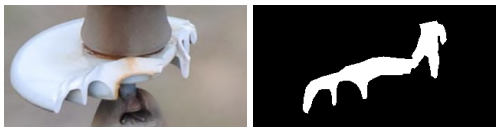
Datasets

We investigate the performance of FCDD with two datasets. The first dataset is an openly available dataset curated by the Electric Power Research Institute and is referred to as the Insulator Defect Image Dataset (IDID) [105]. This dataset consists of 1600 images of insulators with various disk colors. Each image in this dataset has four orientations – original, diagonally flipped, horizontally flipped, and vertically flipped. In this work, we use only the images with original orientations, which results in 400 images with 3286 healthy disks, 716 flashed disks, and 282 broken disks. This dataset is used to study anomaly

detection performance in data-abundant scenarios. The second dataset (SG) is a collection of 77 aerial images collected by Swissgrid AG, with 2429 healthy disks, only 53 flashed disks, and no broken disks. This dataset is used as the data-scarce scenario in this study. Since semi-supervised training requires the ground truth anomaly maps, these datasets are manually labeled with the “labelme" tool [127]. This involves generating polygonal segmentation masks for flash-over patches and regions of broken disks. The segmentation masks are then converted to $\{0,1\}^{h\times w}$ to obtain the ground truth map, which is shown in Figure 17 for healthy, flashed, and broken disks. Note that the maps for healthy disks are completely black, and can be generated automatically. This reduces the manual burden of labeling only faulty disks.

(a) Healthy disk and its map

(b) Flashed disk and its map



(c) Broken disk and its map

Figure 17: Sample disks (healthy, flashed and broken), and their corresponding ground truth maps for semi-supervised training.

Experimental setup

We first study the performance of the FCDD model trained individually on IDID and SG datasets and compare the performance in data-abundant and data-scarce scenarios. Then, we investigate whether data from a different dataset can be used to improve the performance in a data-scarce scenario. To that end, we include data from both IDID and SG datasets to train a third FCDD model and examine if it performs better compared to the model trained only on SG dataset. Note that while IDID has two types of faulty disks (flashed and broken), the SG dataset has only flashed disks. Thus, we train an FCDD model for the IDID dataset with both types of faulty disks for IDID and use only flashed disks of IDID to augment the performance of (the third) FCDD for the SG dataset. The number of training and testing images used for different experiments are listed in Table 5. As an illustrative case, for unsupervised training with no real anomalies for IDID, we use 2586 healthy disks, while for semi-supervised training, we additionally use 200 faulty disks and 100 broken disks.

Table 5: Number of samples used for training and testing of FCDD for insulator fault detection. In unsupervised training, we use all healthy samples along with 1) no anomalous samples or 2) anomalous samples without ground truth maps. In semi-supervised training, we utilize all healthy and anomalous samples with the ground truth maps. ((F): Flashed disks, (B): Broken disks).

Dataset	#Healthy Samples		#Anomalous Samples	
	Train	Test	Train	Test
SG	2379	50	6	47
IDID	2586	700	200 (F) + 100 (B)	516 (F) + 182 (B)
SG + IDID	2379 0	50 0	6 716	47 0

All experiments are performed with the FCDD package [125]. The default settings of FCDD have been employed, of which the notable settings are an input image size of $3 \times 224 \times 224$, the VGG-11-BN-based deep convolutional model [128], a batch size of 128, 200 epochs for training and stochastic

gradient descent optimizer. All experiments are performed on a Windows workstation with one NVIDIA RTX 3070 GPU and 128 GB of RAM. In order to evaluate and compare the performance of the models, we employ the standard area under the receiver operating characteristics curve (AUC) metric. The AUC can be expressed as:

$$AUC = \int_{\mathcal{T}} R_{TP} dR_{FP}, \tag{22}$$

where R_{TP} and R_{FP} are the true positive rate and false positive rate, respectively, and are obtained as:

$$\begin{aligned} R_{TP} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ R_{FP} &= \frac{\text{FP}}{\text{FP} + \text{TN}} \end{aligned}$$

where, TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. $\mathcal{T}$ represents the set of all decision thresholds used to predict the positives and negatives. First, different values of true and false positive rates are obtained for different classification thresholds, and AUC is then calculated according to Equation 22. The optimal decision threshold can be obtained from the receiver operating characteristics by identifying the top-left point of the curve, which can then be used to evaluate the classification accuracy of the model. In order to account for uncertainty during training, FCDD trains 5 different instances of a model by default. We report the average performance (AUC and optimal accuracy) of the five models in each experiment.

3.2 Results for insulator fault detection and classification

3.2.1 Object detection-based approach

In this section, we present the findings of the computational experiments for the three tasks identified in Section 3.1.1. We compare different models and training approaches for each task and highlight the combination that delivers the best performance in detecting the objects of interest.

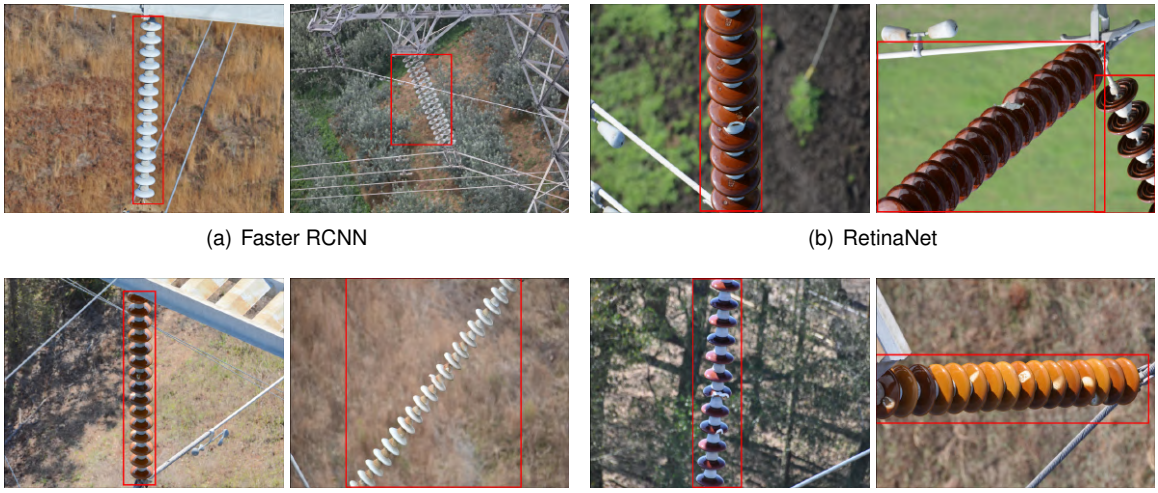
Task 1: Insulator detection

The first task (Insulator detection) is a one-class detection problem with fairly large object sizes compared to the other two tasks. The mAP_{50} of pre-trained models on the test images of the reference dataset are 0.86, 0.88, 0.80, and 0.97 for FRCNN, RetinaNet, FCOS, and YOLOv5, respectively. The YOLOv5 model performs the best, followed by RetinaNet, FRCNN, and FCOS. Some illustrative examples of predictions with pre-trained models are shown in Fig. 18 and the performance of target-trained and fine-tuned models are summarised in Table 6. YOLOv5 outperforms the other models for both types of training. The FRCNN models exhibit the second-best performance, followed by FCOS and RetinaNet. The fine-tuned YOLOv5 model performs much better than the others and is only 14 points short of the model trained from scratch, compared to the others that are at least 30 points apart. This is in stark contrast to other models that exhibit a sharp difference in performance between the fine-tuned and target-tuned models.

YOLOv5 consistently outperforms the other three models for all the tasks and training procedures, i.e., training from scratch and fine-tuning. This can be attributed to the data augmentation-heavy pipeline of YOLOv5, which provides a significant advantage in the low data regime.

Task 2: Incipient fault detection

The second task (Incipient fault detection) detects healthy and faulty disks in the insulators and can be used to assess the health of the insulators. This is a 4-class detection problem as discussed in



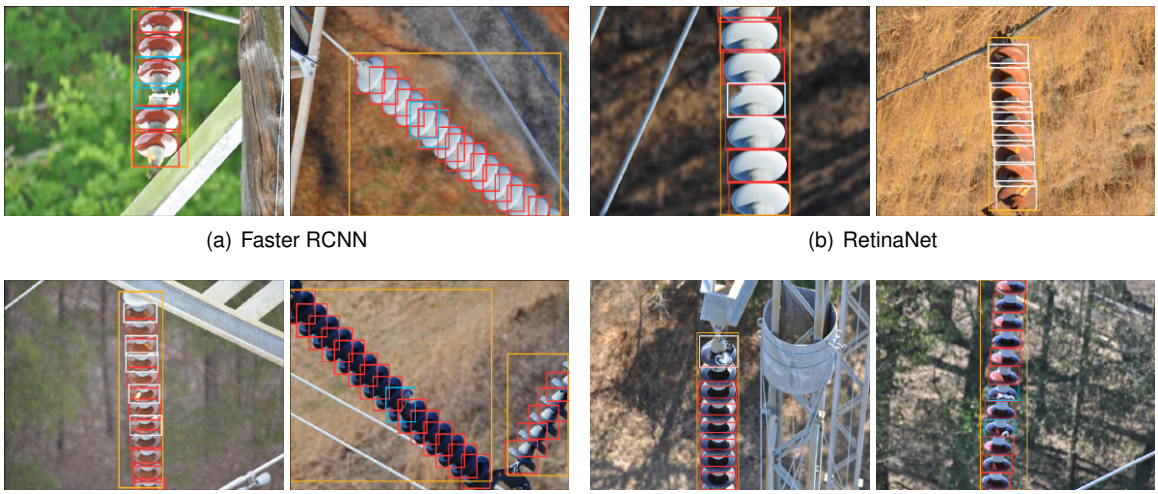
(a) Faster RCNN (b) RetinaNet

(c) FCOS (d) YOLOv5

Figure 18: Predictions of pre-trained models on Task 1 (insulator detection) on test images of the reference dataset.

Table 6: mAP_{50} for Task 1: Insulator detection, on target dataset.

Model	Trained from scratch	Fine-tuned
FRCNN	0.76	0.45
RetinaNet	0.58	0.15
FCOS	0.69	0.18
YOLOv5	0.87	0.73



(a) Faster RCNN (b) RetinaNet

(c) FCOS (d) YOLOv5

Figure 19: Predictions of pre-trained models trained on Task 2: Insulator and disk detection, on test images of the reference dataset (orange: insulator, red: healthy disks, blue: broken disks, white: flashed disks).

Section 3.1.1. Since each insulator has approximately ten disks, there are always ten times as many disks as insulators. Moreover, the number of healthy disks is also five to six times the number of flashed and broken disks in the reference dataset. Thus, this is a more challenging detection task with varying

sizes of the objects and frequencies of their occurrence. The pre-trained YOLOv5 model has an mAP of 0.98 on the test images of the reference dataset, while the FRCNN, RetinaNet, and FCOS have mAPs of 0.89, 0.89 and 0.85, respectively. These performance scores are comparable to the one-class detection models, with approximately two to three points better mAP on the 4-class detection task. Therefore, the models are able to capture the differences in patterns of healthy, flashed, and broken disks. Some exemplary predictions of the models on the reference dataset are illustrated in Fig. 19.

The target dataset also exhibits class imbalance across the disks and insulators. The number of healthy disks is about 58 and 13 times the number of flashed disks and insulators. This dataset does not have any broken disks. Table 7 presents the mAP of the pre-trained and fine-tuned YOLOv5 models on the test images of the target dataset for the four object types. The models have the worst performance for flashed disks, which are also the least frequent objects in the dataset. The fine-tuned model performs poorer than the target-trained model as also observed for Task 1, while the difference in overall mAP is smaller (nine points) compared to the one-class detection model.

Table 7: mAP_{50} of YOLOv5 models for Task 2: Insulator and disk detection, on the target dataset.

Object	Trained from scratch	Fine-tuned
Insulator	0.87	0.74
Disk (H)	0.76	0.72
Disk (F)	0.22	0.13
Disk (B)	–	–
Overall	0.62	0.53

Task 3: Multiple object detection

The third task (Multiple object detection) involves the detection of six objects and is the most difficult. Both the reference and target datasets have severe class imbalance for bird nests in addition to the flashed and broken disks. The Stockbridge dampers are also the smallest objects as well as up to five times less frequent than the healthy disks. The mAP of the pre-trained models on the test images of the reference dataset are 0.85, 0.85, 0.79, and 0.85 for the FRCNN, RetinaNet, FCOS, and YOLOv5 models. These scores are also comparable to the one-class and four-class models, allowing for the detection of multiple assets from the aerial images.

Table 8 summarises the mAP of the target-trained and fine-tuned YOLOv5 models. The difference in performance between the two models is much more severe than the one-class and four-class models. Specifically, the overall mAP of the fine-tuned models is 26 percent lower than that of the target-trained model, the insulators and bird nests being the biggest contributors to this difference. This may be due to the nature of these objects. Specifically, the bird nest is always occluded by structures of the tower and might be difficult to detect - even by humans. Conversely, the insulator is the largest object and stands out from the other objects that are comparably small. This difference in size compared to all other classes together with the class imbalance might potentially lead the model to emphasize object prediction at a smaller scale. This shortcoming could be addressed by appropriately weighing the contributions of the objects to the loss function. Figure 20 shows the predictions of the model on two sample test images of the target dataset. The bounding boxes predicted by the model for different objects are shown in different colors. The model is able to detect all the healthy disks, dampers, insulators, and a bird nest. However, in the right panel of Fig. 20, the model misclassifies a flashed disk as healthy, which explains its poor performance in detecting flashed disks on the target dataset (see Table 8).

Finally, we observe that the fine-tuned models perform significantly worse than the target-trained models for the majority of the tasks, as seen in Table 6, 7 and 8, thus discouraging the use of pre-training and fine-tuning. In order to further investigate this observation, we perform a sensitivity analysis of the fine-tuned models.

Figure 20: Predictions of YOLOv5 model trained from scratch on Task 3: Multiple object detection, on test images of the target dataset (orange: insulator, red: healthy disks, green: Stockbridge dampers, yellow: bird nest).

Table 8: mAP_{50} of YOLOv5 models for Task 3: Multiple object detection, on the target dataset

Object	Trained from scratch	Fine-tuned
Insulator	0.90	0.67
Disk (H)	0.77	0.70
Disk (F)	0.18	0.14
Disk (B)	–	–
Nest	0.87	0.48
Damper	0.71	0.50
Overall	0.68	0.50

Sensitivity analysis

In the previous results, we present the performance of the fine-tuned models that use 30% of the target dataset for training. On the other hand, the model trained from scratch uses 70% of the target dataset. It is common practice to use a smaller portion of the target dataset for fine-tuning and this is one of the key advantages of using a pre-trained model. We perform a sensitivity analysis to examine whether the amount of training data has any impact on the performance of the fine-tuned models. We use 10%, 30%, and 70% of the target dataset for fine-tuning and recording the performance of the models on the remaining images. Table 9 lists the object-wise mAP of these fine-tuned YOLOv5 models and shows that even with 10% of the data, i.e., the eight images used for fine-tuning, the model has mAPs of 0.63 and 0.7 for the insulator and healthy disk, respectively. This can be attributed to the large size of insulators and the high frequency of healthy disks. The detection of rare classes is unsurprisingly poor for small fractions of the target dataset used for fine-tuning but improves significantly with the addition of more images. The last model in Table 9 uses the same number of images for fine-tuning as the model trained from scratch. These two models differ only in the initial parameters before training, i.e., while the former starts with parameters learned from the reference dataset, the latter starts with random initialization. A comparison of the two models (Table 8 and Table 9) reveals that the fine-tuned model has poor performance compared to the target-trained model even when trained on the same images. We observe the opposite behavior for the Faster RCNN, RetinaNet, and FCOS models across all tasks, with approximately five points of improvement in mAP compared to the target-trained model. This difference in behavior may be due to the different data processing pipeline of YOLOv5.

Figure 21 compares the performance of fine-tuned FRCNN and YOLOv5 models for Task 3 with

Table 9: mAP_{50} of target-trained and fine-tuned YOLOv5 models trained with different fractions of the target dataset for Task 3: Multiple object detection (TT: target-trained, FT: fine-tuned).

Object	0.1		0.3		0.7	
	TT	FT	TT	FT	TT	FT
Insulator	0.74	0.63	0.70	0.67	0.90	0.82
Disk (H)	0.62	0.70	0.77	0.70	0.77	0.60
Disk (F)	0.04	0.08	0.07	0.14	0.18	0.19
Disk (B)	–	–	–	–	–	–
Nest	0.11	0.29	0.47	0.48	0.87	0.66
Damper	0.06	0.07	0.83	0.50	0.71	0.58
Overall	0.31	0.36	0.57	0.50	0.68	0.57

different fractions of the target dataset. YOLOv5 performs much better in the extremely low-data part of the plot. This may be attributed to the data augmentation-heavy pipeline of YOLOv5, which adds significant value with 40% and smaller fractions of datasets used for training. On the other hand, FRCNN has very poor performance in the extremely low-data part and performs comparably with the YOLOv5 model when more data is available.

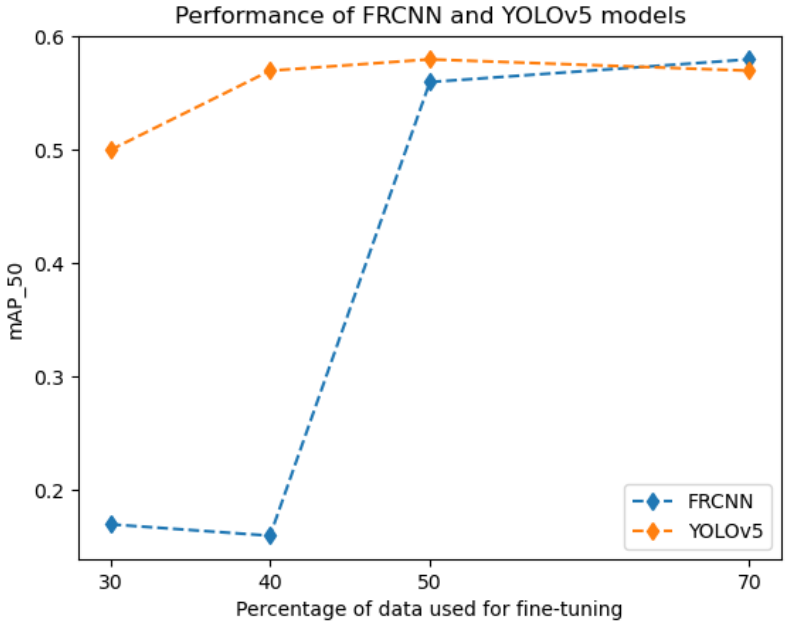


Figure 21: Test performance of FRCNN and YOLOv5 models (Task 3) fine-tuned with different fractions of the target dataset.

3.2.2 Object detection and anomaly detection-based approach

In this section, we present an evaluation and comparison of FCDD models trained for data-abundant and data-scarce scenarios, with different formulations. We then present the explanations provided by the semi-supervised FCDD model for its predictions, followed by a discussion of the results and future work.

Fault detection with data abundance

Unsupervised training of the FCDD model without any faulty samples provides an AUC of 0.5923

for IDID, corresponding to a classification accuracy of 0.5735 for the optimal decision threshold. The AUC and optimal accuracy increase to 0.7920 and 0.7172, respectively with the inclusion of only flashed disks as anomalous data points during unsupervised training, i.e., without any ground truth anomaly maps. This is a large improvement in performance, achieved with only 200 training images of flashed disks, and underscores the significance of using real anomalies at the time of training. We also trained models with both broken and flashed disks (without any ground truth anomaly maps), and observed the AUC to be 0.7491, corresponding to an accuracy of 0.6889. This further demonstrates that FCDD can learn from multiple types of anomalies in the training dataset, and deliver performance comparable with learning from only one type of anomaly.

In the semi-supervised formulation, the FCDD model provides an AUC of 0.9091, and an accuracy of 0.8292, when trained and tested with only flashed disks as anomalous data points. This demonstrates the advantage of using ground truth anomaly maps to inform the training process about the patterns observed in faulty data – in a more explicit manner. Similar to the unsupervised formulation, we performed semi-supervised training with both broken and flashed disks and obtained an AUC of 0.9040, corresponding to an accuracy of 0.8266. This again demonstrates the capability of the FCDD model to deliver comparable performance with multiple types of anomalies. It must be noted, however, that the model can only differentiate between normal (healthy) and anomalous (faulty) data points, and cannot identify the nature of an anomaly (flashed or broken).

Fault detection with data scarcity

In the data-scarce application, the unsupervised FCDD model trained without any real anomalies provides an AUC of 0.7387 and an accuracy of 0.7051, which is considerably higher than the performance in a data-abundant scenario. This difference may partly be attributed to the larger number of images used to evaluate the models in data-abundant scenarios, which exposes the model to more variations in the images. The model trained with flashed disks as anomalies provides an AUC of 0.8495 and an accuracy of 0.7608. This suggests that FCDD can leverage information about the anomalies in an efficient manner, i.e., with only 6 anomalies, and provide a considerable improvement in the performance of fault detection.

In the semi-supervised formulation, the FCDD model delivers an AUC of 0.8830, corresponding to an optimal accuracy of 0.8061. This further highlights the value of using ground truth anomaly maps to inform the training process about the specific patterns of faults.

Improving anomaly detection in data scarce scenario

We adopt two different methods of increasing the training dataset size to improve the performance in data-scarce scenarios. In the first approach, we increase the number of training anomalies to cover half of the total number of available anomalies. We perform this experiment in both the unsupervised and semi-supervised settings. The AUCs of the models trained with 6, 14, 20, and 26 anomalies are shown in Figure 22. The AUC of the unsupervised model without any training anomalies is also shown for reference. We observe that both models provide a consistent improvement in performance with more training anomalies. The semi-supervised models can also be seen to perform better than the unsupervised models for all the cases. The model trained with 26 anomalies provides an AUC of 0.9193 and an optimal accuracy of 0.8337 with unsupervised training. The corresponding semi-supervised model provides an AUC of 0.9363 and an accuracy of 0.8597.

In the above approach, the number of anomalies used for testing the model is different across the scenarios. In order to maintain the same test images, and increase the number of training anomalies, we further leverage data from IDID. Specifically, we use all images of flashed disks in IDID, in addition to the 6 training anomalies, and train unsupervised and semi-supervised FCDD models. The unsupervised model provides an AUC of 0.7940 with an accuracy of 0.7237, which is poorer than the corresponding model trained with only 6 anomalies of SG dataset. The semi-supervised model provides an AUC

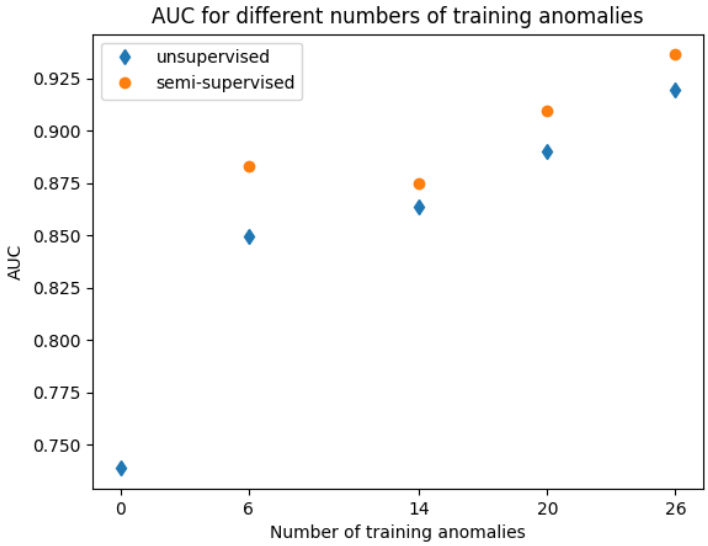


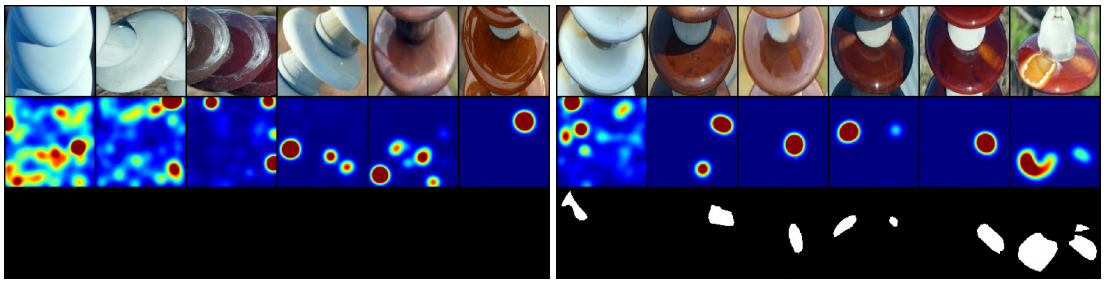
Figure 22: AUC of FCDD models trained on SG dataset with different numbers of training anomalies.

of 0.8860 and an accuracy of 0.8123, which is only marginally better than the corresponding model trained with only SG dataset. These observations suggest that in an unsupervised formulation, including training anomalies from a different dataset hampers the learning process, while in the semi-supervised formulation, the advantage gained from such anomalies is minimal. Moreover, we observe that the patterns of flash-over of disks are similar for both datasets, suggesting that the poor performance might be partly due to the high variation of colors in disks of IDID, compared to the SG dataset.

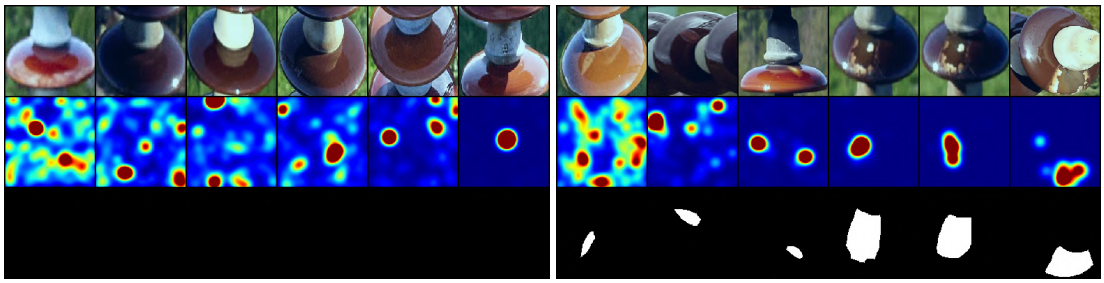
Explanations of FCDD

The explanations from an FCDD model are obtained by up-sampling the model's output features (z) to match the original image size, through a deterministic non-trainable Gaussian kernel-based approach [125]. The up-sampled features ($\tilde{z}$) are then represented as a heat map to highlight the anomalous regions predicted by the model. Figure 23 shows the explanations of the semi-supervised models on sample test images for both datasets. For each dataset, Figure 23 shows six sample test images (healthy in the left panel and faulty in the right panel), their explanations produced by the model, and the corresponding ground truth anomaly maps. It can be observed that the explanations for the faulty images match closely with the ground truth anomaly maps for the images from IDID. This suggests that the model is able to localize patches of discoloration on the disks with very good accuracy. The model also predicts anomalous regions on the disks that are not aligned with the ground truth anomaly maps. The explanations for the healthy images, on the other hand, consistently contain anomalous regions predicted by the model.

The explanations for the faulty images of the SG dataset are also well aligned with the ground truth maps, although they exhibit slightly more deviation from the ground truth anomaly maps compared to IDID. This can also be observed for the healthy images in this dataset. The qualitatively poor explanations for the SG dataset compared to IDID can be corroborated by the difference in AUC and optimal accuracy of the two models trained with these datasets. Although both models make false predictions, i.e., spurious anomalous regions (on disks, as well as in the background), these predictions are aggregated to decide whether the image is normal or anomalous. This aggregation and classification based on the optimal threshold finally provides a very good accuracy.



(a) Explanations for IDID (left: healthy, right: faulty).



(b) Explanations for SG dataset (left: healthy, right: faulty).

Figure 23: Explanations of FCDD models for IDID (data abundant) and SG dataset (data scarce), trained in a semi-supervised manner. In each panel, the top, middle, and bottom rows show the original image, explanations provided by the FCDD model and ground truth anomaly maps, respectively. The explanations are shown as heat maps, with higher values (shown in red) representing a faulty pattern and lower values (shown in blue) representing a healthy pattern.

4 Power transformers

This work focuses on the Dissolved Gas Analyses (DGA) data for transformer fault detection and diagnostics. The DGA analyzes the composition and concentration of gases dissolved in the transformer's insulating oil. The presence and relative quantities of specific gases, such as Hydrogen (H2) methane (CH4), ethane (C2H6), ethylene (C2H4), and acetylene (C2H2). Furthermore, the analyses often use gasses such as carbon monoxide (CO) and Carbon dioxide(CO2).

4.1 Methods for power transformer fault detection and classification

To train and validate a machine learning model or utilize any conventional method, we need a sufficient amount of data and labeled data. We have obtained a Swissgrid data set of transformers with 84 samples. Since this data is small and insufficient to train models, we have requested data from Fachkommission für Hochspannungsfragen (FKH). The company has granted us a large data set of over 10,000 DGA samples. The data includes power transformers and other components, including current and voltage measurement transformers, bushings, etc. After cleaning, we obtained a dataset of 3500 DGA samples of Swiss power transformers. Unfortunately, the data does not contain all the labels and when labels are provided, they are not based on expert validation of the possible faults. This is a major problem in the application of ML. In this work, we utilize conventional methods to obtain initial labels. In the first step, we have developed most of the known conventional algorithms and compared the obtained diagnostics results, i.e., labels. In the second step, we utilize these labels to train various ML methods. Under labels, here we refer to the type of faults commonly defined for transformers:

- Partial discharges (PD): Discharges of the cold plasma (corona) type in gas bubbles or voids, with the possible formation of X-wax in the paper.
- Discharges of low energy (D1): Discharges in paper or oil, with power follow-through, resulting in extensive damage to paper or large formation of carbon particles in the oil, metal fusion, tripping of the equipment, and gas alarms.
- Thermal fault, >700 ℃ (T3): high-temperature fault without carbonization, >700 ℃.
- Thermal faults without carbonization, >700 ℃ (T3-H): Extensive formation of carbon particles in oil, metal coloration (800 ℃) or metal fusion (>1000 ℃).
- Thermal faults with carbonization, >700 ℃ (T3-C): confirmation of high-temperature thermal issue (above 700 ℃) with paper involvement in the fault (carbonization).
- Thermal faul without carbonization, 300<T<700 ℃ (T2-0): confirmation of thermal issue, with a temperature of 300 to 700 ℃ but unlikely to involve solid insulation or paper carbonization.
- Thermal fault with carbonization, 300<T<700 ℃ (T2-C): Confirmation of thermal issue with temperature in the range 300 to 700 ℃, with a high likelihood of paper involvement (probability of 80 %, based on data from transformers showing faults in internal inspection).
- Thermal fault without carbonization, <300 ℃ (T1-0): Thermal issue with expected temperature <300 ∘C but without carbonization of solid insulation.
- Thermal fault with carbonization, <300 ℃ (T1-C): Confirmation of thermal issue with expected temperature <300 ℃ but now with likely involvement of paper, showing carbonization.
- Stray gassing S of mineral oil at 120 and 200 ℃ in the laboratory (S).

4.1.1 Conventional methods for power transformer fault detection and classification

The conventional methods use statistics to determine from a DGA to 1) identify the fault detection (Status) and 2) perform fault diagnostics, i.e., identify the type of fault.

Fault detection

In [129], the Status of a transformer is defined as:

- **Status 1:** Low gas levels and no indication of gassing (Unexceptional DGA or healthy)
- **Status 2:** Intermediate gas levels and/or possible gassing (Possibly suspicious DGA)
- **Status 3:** High gas levels and/or probable active gassing (Probably suspicious DGA or faulty)

To identify the status of a transformer using DGA samples, we utilize reference values (limits) that are statistically determined. In other words, we reference the 90th and 95th percentile of the gas concentrations in a dataset. More advanced approaches utilize additional data, i.e., the IEEE Std.C57.104-2019 [129] uses 95th percentile values for absolute level change between successive laboratory DGA samples (delta) and 95th percentile values from multi-point (3-6 points) rate analysis of laboratory DGA samples (rate). Here, we have the 90th percentile limits provided by FKH available. In addition, we calculate the 95th percentile for the 3000 DGA samples of Swiss transformers provided by FKH. Based on these limits and the limits calculated (with more than 1 million DGA samples) in IEEE Std.C57.104-2019 [129], we developed 4 methods to assess the state of health (Status) of a transformer:

- **Method 1:** uses only the 90th percentile from Fachkommission für Hochspannungsfragen (FKH) data.
- **Method 2:** uses the 90th and 95th percentile from FKH (95th is calculate by us)
- **Method 3:** uses the 90th and 95th percentile from FKH and the delta and rate according to IEEE Std. C57.104-2019
- **Method 4:** uses all limits according to IEEE Std.C57.104-2019 [129].

Methods 1 and 2 use only limits based on FKH percentiles. Method 3 is a hybrid and thus uses

the 90th and 95th percentiles from FKH and the delta and the rate from IEEE Std.C57.104-2019 [129]. Method 4 uses only the IEEE Std.C57.104-2019 [129] algorithm for detecting the transformer Status (Status1, Status2, Status 3).

Fault diagnostics (classification)

When Status 3 is identified we utilize a method for fault diagnostics (i.e., identify the type of fault). We have developed the algorithms for the following methods:

- **Duval triangle:** The Duval triangle method is widely used for interpreting Dissolved Gas Analysis (DGA) results in power transformers. It involves plotting the concentrations of different gases, such as methane, ethane, and ethylene, on a triangular diagram to identify potential faults within the transformer oil. Each triangle corner represents a specific fault category, allowing analysts to visually assess the type and severity of potential issues, such as overheating, partial discharge, or electrical arcing [130].
- **Duval pentagons:** The Duval pentagon is an extension of the Duval triangle, i.e., the relative percentages of the five leading hydrocarbon gases analyzed by DGA are first calculated. There are several variations of the method (Pentagon 1, Pentagon 2, and Pentagon 1+2), where some variations include more fault types. Triangles and pentagons are very popular tools for DGA [131].
- **Doernenburg ratios:** The Doernenburg ratios method in DGA involves evaluating specific gas ratios within transformer oil to assess potential faults. Named after Dr. Heinz Doernenburg, this method focuses on the ratios of certain key gases, such as the methane-to-ethylene ratio, to identify and diagnose transformer-related issues. By analyzing these ratios, experts can gain insights into the type and severity of faults, aiding in the proactive maintenance of power transformers and enhancing the reliability of electrical systems [129].
- **Rogers Ratios:** The Rogers method is similar to the Doernburg ratios method. It uses three gas ratios indicating five different types (cases) of faults, depending on the values of the ratios. The limitation of the Rogers Ratios Method is that it cannot identify faults in a relatively large number of DGA results (typically 35%) because they do not correspond to any of the ratios defined by the method [129].
- **IEC method:** The IEC (International Electrotechnical Commission) method for DGA is a standardized approach used to assess the condition of power transformers. It involves measuring and analyzing the concentrations of various gases in the transformer oil, such as methane, ethane, and ethylene. Interpreting these gas levels, established ratios, and diagnostic criteria outlined in IEC standards help identify potential faults, such as overheating, partial discharge, or electrical arcing, allowing for timely maintenance and preventing catastrophic failures [132].

Methods 1 and 2 can be used for transformers and other components such as measurement transformers and bushings. However, Methods 3 and 4 are only applicable to transformers, since the method is based on IEEE Std.C57.104-2019 [129], which is developed specifically for power transformers.

4.1.2 ML models for power transformer fault detection and classification

Besides the fully connected feedforward neural networks (see Section 2.1.2), we have tested a large set of machine learning methods: Decision trees, Naive Bayes Classifiers, Support Vector Machines, Ensemble classifiers (Boosted Trees, Bagged Trees, RUSBoosted Trees).

4.2 Results for power transformer fault detection and classification

In this section, we provide the results from the transformer fault detection and diagnostics obtained with the conventional methods (Section 4.2.1) and ML models (Section 4.2.2).

4.2.1 Conventional methods

As described in Section 4 we are utilizing most of the conventional techniques to identify the state of health of a transformer. First, we identify the Status of the transformer (healthy, suspicious, faulty) with one of the four methods (Method 1, 2, 3, 4), and then for the potentially faulty transformers (Status 3) we perform diagnostics (partial discharge, thermal fault, etc.). Here we will only show the fault detection results from Method 1 and Method 4, and fault diagnostics results from Duval Triangle and Duval Petagon 1+2 for both the FKH and the Swissgrid datasets. The results for Method 2 and Method 3 are not presented since these methods are similar with respect to the utilized percentiles and obtained results with Method 1 and Method 4, respectively.

Results obtained for the FKH data: Method 1

Here, Method 1 is used to determine potentially faulty transformers, after which we apply the fault diagnostics methods. Figures 24 and 25 show the classification of the faults according to the Duval Triangle and Duval Penatgon 1+2, respectively. We observe that with both methods, the faults are spread across all classification types, with a higher concentration of high-temperature thermal faults and partial discharge. It is worth noting that Method 1, which uses the 90 percentile to determine which transformers are potentially healthy (Status 1) and which are potentially faulty (Status 3), is a conservative approach. It is possible that some of the transformers are actually suspicious (Status 2) and only need more frequent observations.

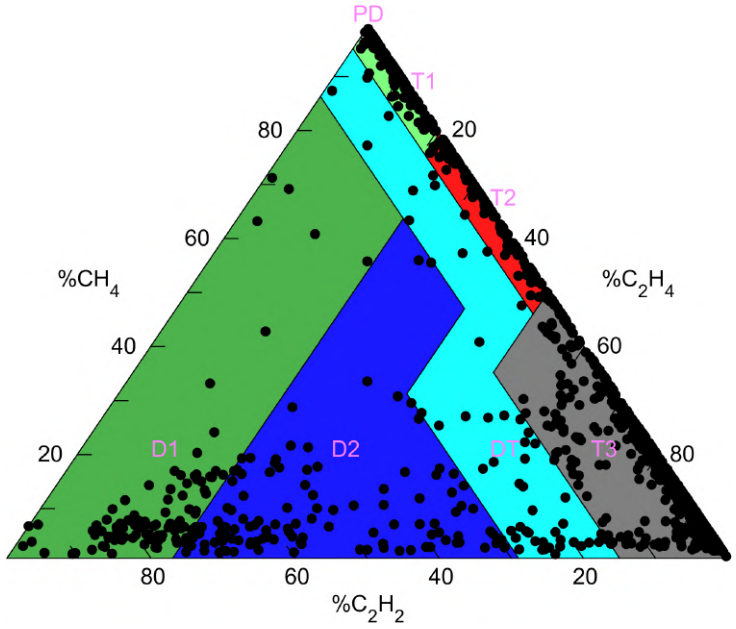


Figure 24: FKH transformer fault diagnostics for Method 1 (potentially faulty transformers are determined using the 90 percentile from the FKH data).

Results obtained for the FKH data: Method 4

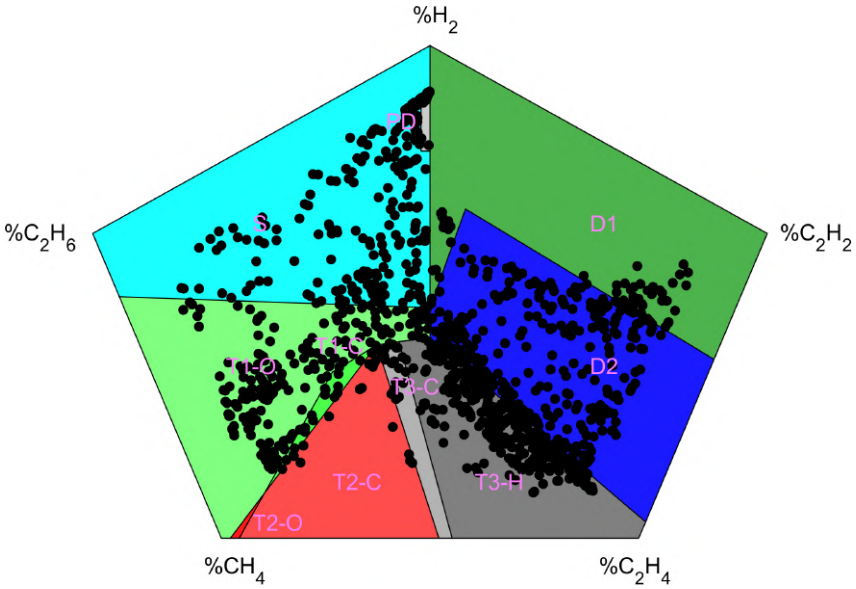


Figure 25: FKH transformer fault diagnostics for Method 1 (potentially faulty transformers are determined using the percentile from the FKH data).

Here, Method 4 is used to determine potentially faulty transformers, after which we apply the fault diagnostics methods. Figures 26 and 27 show the classification of the faults according to the Duval Triangle and Duval Penatgon 1+2, respectively. We observe similar behavior as in the case of Method 1. The main difference is the number of data points classified as faulty, particularly in the case of label T1-O (low-temperature thermal fault with oxidation). In other words, some data points in Method 4 are classified as suspicious (Status 3), and Method 1 classifies them as faulty, which, in most cases, belongs to the faulty class T1-O. Not that the 90 percentiles in Method 1 and Method 4 are different. The Method 1 percentiles are calculated from the FKH data, which comprises only Swiss transformers. The Method 4 percentiles are calculated using a large DGA dataset from North America (over 1 million samples). Therefore, the percentiles are different, which affects the identification of faults. Unfortunately, we do not know the true labels of the Swiss transformers, and therefore, we cannot check which method performs better.

Results obtained for the Swissgrid data: Method 1

Here, Method 1 is used to determine potentially faulty transformers, after which we apply the fault diagnostics (classification) methods on the Swissgrid dataset. Figures 28 and 29 show the classification of the faults according to the Duval Triangle and Duval Penatgon 1+2, respectively. The Duval triangle and Duval pentagon classifications show that the faults are concentrated in T1 (T1-O) and T3 (T3-H) fault classes. These results indicate that there is a possibility of thermal faults without paper carbonization in a small set of transformers. However, further examinations are required to confirm these findings.

Results obtained for the Swissgrid data: Method 4

Here, Method 4 is used to determine potentially faulty transformers, after which we apply the fault diagnostics methods on the Swissgrid dataset. Figures 30 and 31 show the classification of the faults according to the Duval Triangle and Duval Penatgon 1+2, respectively. According to Method 1 and Method 4, 19 and 20 samples out of the 84 available, respectively, are determined to be potentially faulty. In both methods, the Duval classification shows the faults are concentrated in T1 (T1-O) and T3 (T3-H) fault classes. Although, very similar, the outcome of Method 4 is another 24 data points considered suspicious (Status 3). In addition, we have analyzed the result for Method 2 (90 and 95 percentile

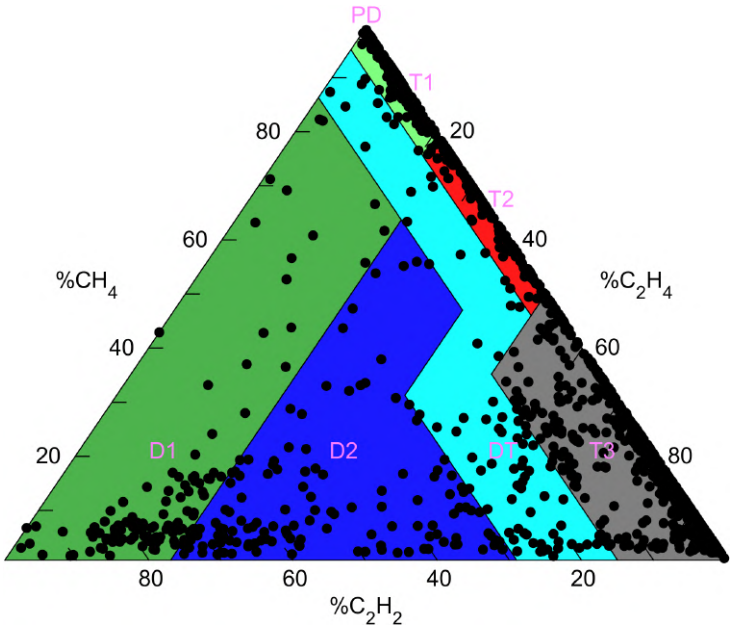


Figure 26: FKH transformer fault diagnostics for Method 4 (potentially faulty transformers are determined using the percentiles from the IEEE Std.C57.104-2019 [129]).

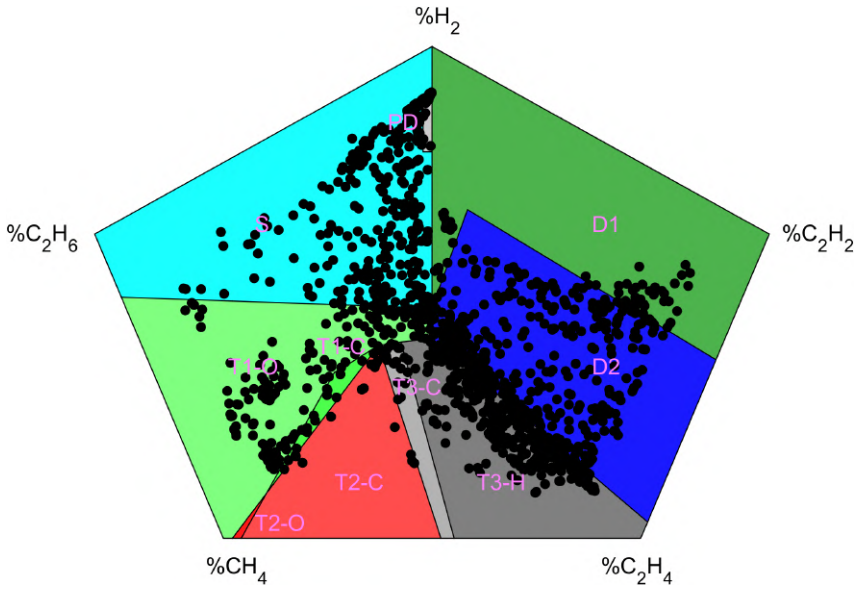


Figure 27: FKH transformer fault diagnostics for Method 4 (potentially faulty transformers are determined using the percentiles from the IEEE Std.C57.104-2019 [129]).

from FKH) and found that only two samples are identified as potentially faulty, and 17 are suspicious. Overall, these analyses show a significant difference between the fault detection results obtained with the percentiles from Swiss transformers (FKH data) and those from North American transformers (IEEE Std.C57.104-2019 [129]).

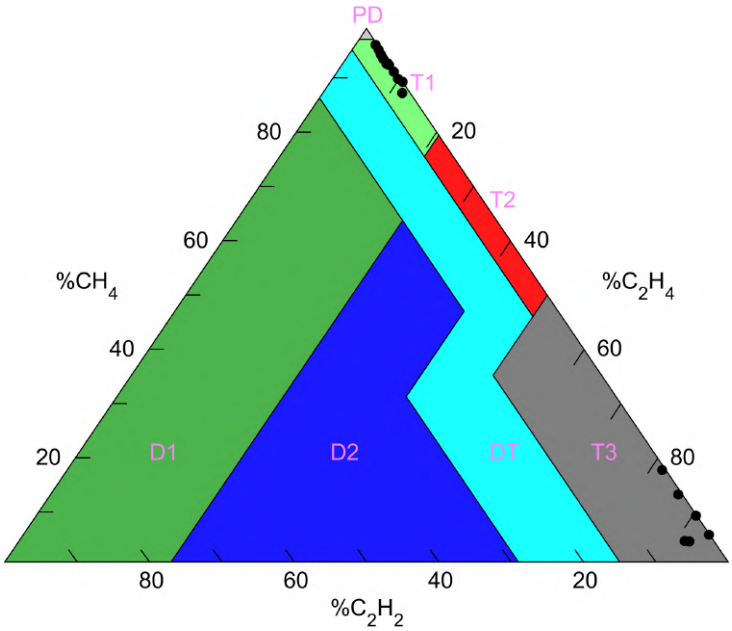


Figure 28: Swissgrid transformers fault diagnostics for Method 1 (potentially faulty transformers are determined using the 90 percentile from the FKH data).

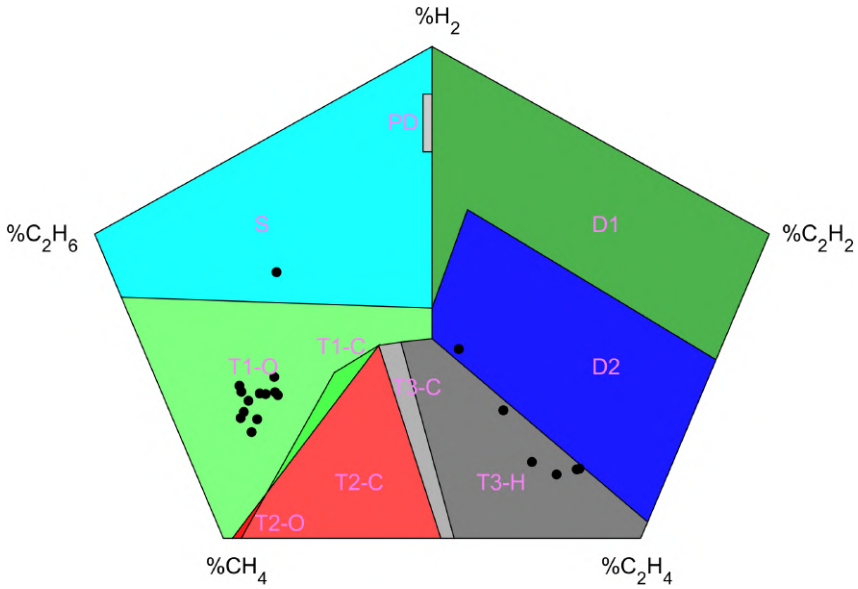


Figure 29: Swissgrid transformers fault diagnostics for Method 1 (potentially faulty transformers are determined using the 90 percentile from the FKH data).

4.2.2 ML models

The literature denotes the Duval triangle as the most accurate classical method for fault diagnostics [132, 133, 134]. Therefore, we use the labels obtained with the Duval triangle to train ML models. In fact, we use Feedforward neural networks, Decision trees, Naive Bayes classifiers, Support vector machines, and Ensemble classifiers (Boosted trees, Bagged trees, RUSBoosted trees). The best (high accuracy) models are obtained with an Optimizable ensemble of Bagged trees using the Classification Learner in

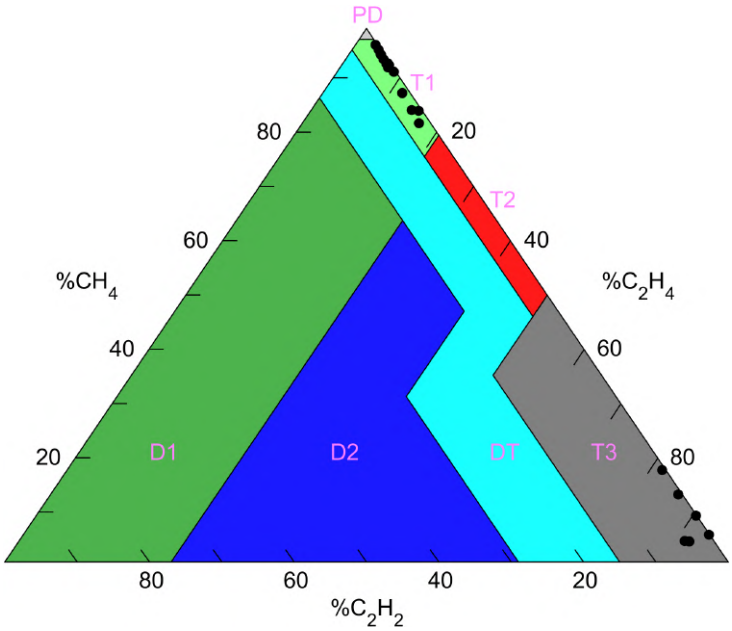


Figure 30: Swissgrid transformers fault diagnostics for Method 4 (potentially faulty transformers are determined using the percentiles from the IEEE Std.C57.104-2019 [129]).

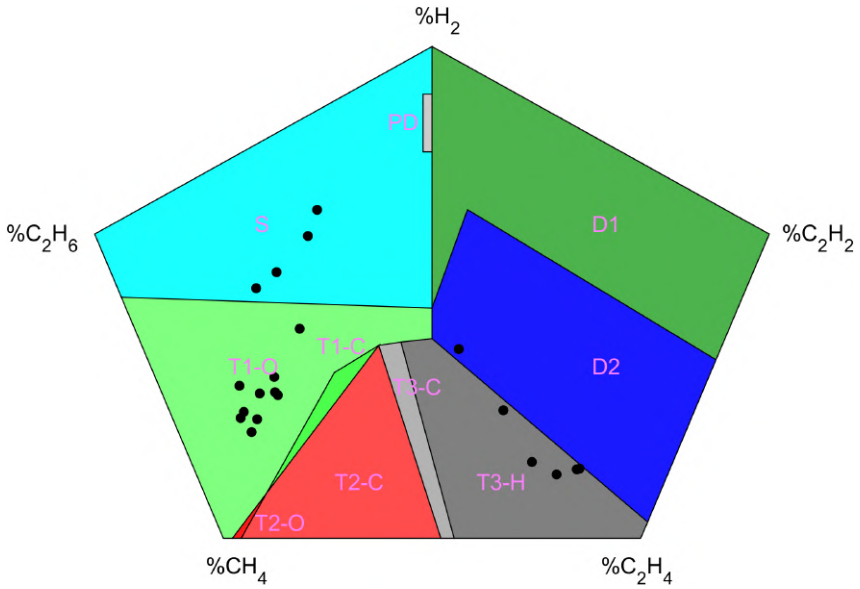
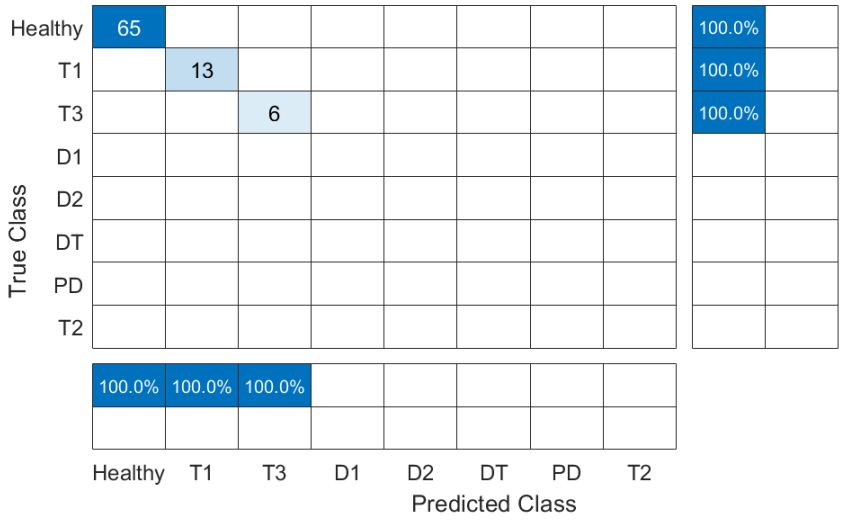


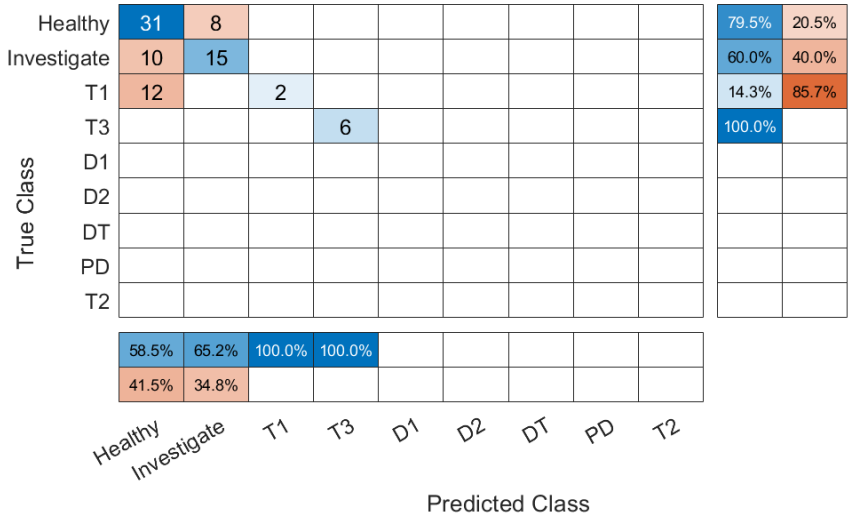
Figure 31: Swissgrid transformers fault diagnostics for Method 4 (potentially faulty transformers are determined using the percentiles from the IEEE Std.C57.104-2019 [129]).

MATLAB [94]. We have trained four models, i.e., a separate model (Model 1, 2, 3, and 4) with the labels obtained with each conventional fault detection method (Methods 1, 2, 3, and 4). The models are trained on the FKH data set and applied to the Swissgrid data set for validation. The classification accuracy of Model 1 and 2 is 100 %, while Model 3 and Model 4 underperform, with accuracy of 77 % and 64 %, respectively. Figire 32 shows the confusion matrices for Model 1 and Model 4, respectively. We observe that the healthy class and the thermal faults 1 and 3 are potential labels for the unhealthy samples in both models. However, Model 4 has many misclassified samples revolving around the healthy class. On the one hand, 8 healthy samples are misclassified as suspicious (Investigate). On the other hand, 10

and 12 suspicious and thermal fault 1 samples, respectively, are misclassified as healthy. Therefore, we recommend using only Models 1 and 2 on DGA data from Swiss transformers. Nevertheless, further investigation, including comparison with the true labels is necessary for all models.



(a) Model 1



(b) Model 4

Figure 32: Confusion matrices for Model 1 and Model 4.

We suspect several reasons for the underperformance of Models 3 and 4:

- The Swissgrid data does not have transformer age and N2 and O2 measurements, which can have a significant impact on the selection of the percentiles (look at Table 1 and Table 2 for the IEEE Std.C57.104-2019 [129]).
- The labeling using the percentiles given by IEEE Std.C57.104-2019 [129] introduces variability. Since our training data is small, it is likely that it does not have enough diversity compared to the data used in the standard to derive these limits (more than 1 million samples). Therefore, our models may not be able to discover all of the patterns.

- There is no past data in the Swissgrid dataset; therefore, the IEEE method's full potential (Method 4) is not utilized.

5 Conclusions

This project is aimed at component-level fault detection and diagnostics. Three components are considered in this project, and state-of-the-art machine learning and deep learning methods were used to perform fault diagnostics. The project used a host of data (current measurements, aerial images, and gas concentration measurements) acquired by the industrial partner, other contributors, and open-source data. State-of-the-art physics-based modeling, evolutionary algorithms, deep learning architectures, probabilistic methods, and transfer learning were adopted to account for the characteristics of real-world data such as incompleteness, uncertainty, variability, and limited dataset sizes.

The project results revealed that existing machine learning models can perform component-level analyses accurately. Physics-based models that act as digital surrogates of the real-life system and generate synthetic data for fault diagnostics were built. Popular neural network architectures such as FNN, CNN, LSTM, and customized architectures were trained on the synthetic data and a comparative study revealed that customized architectures could provide better performance. Uncertainty-aware models could account for multiple sources of uncertainty, resulting in more robust models with better deployment readiness. Computer vision-based object detection models were able to identify multiple assets of interest in aerial images, and anomaly detection models were able to provide explanations of regions that are considered to be faulty by the model. Finally, we applied conventional methods and ML models for transformer fault detection and diagnostics. In the first step, we developed the most known conventional algorithms and compared the obtained diagnostics results, i.e., labels. In the second step, we use these labels to train ML models, which we applied to a test dataset. However, we could not validate the results due to the lack of verified labels for the state of health of the transformers.

We have delivered three of the developed tools to the project partner Swissgrid AG: 1) Object detector model that can be utilized to detect insulators from tower images; 2) Algorithm that encompasses all of the relevant conventional methods for transformer fault detection and diagnostics from DGA data; and 3) Trained ML model for transformer fault detection and diagnostics from DGA data. These tools are expected to be integrated into Swissgrid's internal fault detection and diagnostics platforms.

6 Outlook and next steps

6.1 Power transmission lines

1. **Model development and validation:** The current models are developed by coupling physics-based and deep learning methods. The physics-based model of a power line is developed and calibrated using measured data. The deep learning models were developed using different architectures, which are trained with healthy and faulty data (synthetic) simulated with the physics-based model. It is clear that the uncertainty in the physics-based model propagates into the trained models. This issue can be, to some extent, mitigated in the development of a robust physics-based model, which needs to be developed and validated using not only healthy data but also data that clearly represent fault intensity and location. Similarly, the training of the ML models can be performed with a combination of synthetic data (produced by the physics-based model) and measured

data that represent faulty conditions.
2. **Deployment of ADF models:** The uncertainty-aware ADF (assumed density filtering) models were observed to deliver the best performance in fault diagnosis. However, their deployment in real life requires one to provide mean and variance as input to the model. With the lack of operational data for different conditions available during the project, this was not possible to perform. However, future studies that target this gap can be performed to evaluate the deployment readiness of ADF models in the industrial sector.

6.2 Power transmission insulators

The current work has demonstrated the advantage of a purely object detection-based approach and a two-stage (object detection and anomaly detection) approach. The following gaps still exist in the current work, which can be addressed in future research:

1. **Data completion:** The object detection-based strategy has used openly available data for pre-training the models. Some of the labels in this data are incomplete, which can affect the features learnt by the models. In the future, the labels can be completed by manual efforts to improve the quality of the data and hence, the performance of the resulting models.
2. **Model updation:** The current work has used the state-of-the-art models that were available at the time of the study. However, the field of computer vision is one of the fastest evolving fields, with better models rapidly appearing in the literature. For example, while we use YOLOv5 to train the models, the latest version of this line of models, i.e., YOLOv8 has been shown to outperform its predecessors. The current work has proved that these models can be used to perform multiple object detection, and better models in the future can improve the performance in the future.
3. **Inclusion of multiple assets:** In addition to the insulator, the current work has aimed at detecting bird nests and Stockbridge dampers. However, several other assets of interest, such as tower bars exhibiting corrosion, vegetation encroaching into towers/lines and many more can, in principle, be detected with the adopted approach. However, this will require the collection and labelling of data to train the models. The current work has provided the motivation to conduct such an investigation to automatically detect multiple assets from images, which can be explored in the future.
4. **Domain adaptation:** The current work has adopted transfer learning to leverage the learnt knowledge from the reference dataset to the target dataset and found that this results in poor performance. However, several other approaches, under the umbrella of domain adaptation, exist that transfer the knowledge from one dataset to another. Future work can be directed towards exploiting domain adaptation to improve the performance of the models. This will translate to the model trained on images collected in, for example, the summer (with coloured backgrounds) to be adapted to images collected in the winter (with snow-covered, white backgrounds).

6.3 Power transformers

We have developed both conventional methods and ML models for fault identification and classification in power transformers. The conventional methods are used to identify the state of health of the transformed data. These labels are then used to train the ML models since the transformer data lack the true labels. It is evident that the lack of validation data is a problem that needs further development. Future works should focus on: 1) creating a dataset of Swiss transformers where the true labels are known; and 2) accounting for the uncertainty in the measured DAG data and uncertainty in the labels when training ML models. For the latter part, our preliminary research shows that methodological developments may be needed.

7 National and international cooperation

The project was carried out in cooperation with Swissgrid AG. They have provided the data for modeling the power line from Avegno to Gorduno, which includes measurements of current at the endpoints and parameters of the line. They have also provided aerial images of towers with insulators and concentrations of gases in power transformers for fault diagnosis. Furthermore, we have established cooperation on the gas concentration data with Fachkommission für Hochspannungsfragen (FKH) and discussed relevant transformer fault detection and diagnostics methods.

The following members were involved in the research team at the Risk and Reliability Engineering Lab, ETHZ:

1. Giovanni Sansavini (Associate Professor)
2. Blazhe Gjorgiev (Senior Scientist)
3. Laya Das (Postdoctoral Researcher)
4. Mohammad Hossein Saadat (Postdoctoral Researcher)
5. Ambra Van Liedekerke (Masters Student, ETH and Swissgrid)
6. Selina Merkel (Masters Student)
7. Jan-Simon Görtzen (Semester Student)
8. Manon Prarie (Student Assistant)
9. Tyler Anderson (Student Assistant)
10. Athina Nisioti (Student Assistant)
11. Javier Orive Soto (Student Assistant)
12. Adrien Mellot (Student Assistant)

The following members were involved from Swissgrid AG and FKH:

1. Etienne Auger (Head of Procurement Management Team, Swissgrid)
2. Evangelos Vrettos (Research and Development Manager, Swissgrid)
3. Martina Rohrer (Asset Portfolio Engineer, Swissgrid)
4. Marcelo Paiva Rodrigues (Asset Portfolio Engineer, Swissgrid)
5. Martina Kolpondinos (Research and Digitalisation Manager, Swissgrid)
6. Thomas Heizmann (FKH)

In addition to exploring ML models for fault detection and diagnostics, this project had synergies with an activity at our project partner Swisshrid, where they are developing maintenance scheduling tools. In fact, we supported the development of an optimal topological changes algorithm to be applied to the very high-voltage Swiss power grid. This part of the work has been performed within Swissgrid AG by an MSc student co-supervised by ETH Zurich. The work is published as an MSc thesis in the ETH research collection [135].

8 Publications

1. Gjorgiev, B., Das, L., Merkel, S., Rohrer, M., Auger, E., & Sansavini, G. (2023). Simulation-driven deep learning for locating faulty insulators in a power line. Reliability Engineering & System Safety, 231, 108989.
2. Das, L., Gjorgiev, B., & Sansavini, G. (2023). Uncertainty-aware deep learning for digital twin-driven monitoring: Application to fault detection in power lines. arXiv preprint arXiv:2303.10954.
3. Das, L., Saadat, M. H., Gjorgiev, B., Auger, E., & Sansavini, G. (2022). Object detection-based

inspection of power line insulators: Incipient fault detection in the low data-regime. arXiv preprint arXiv:2212.11017.

4. Das, L., Gjorgiev, B., & Sansavini, G. (2023). Anomaly detection for automated inspection of power line insulators. under preparation.
5. Saadat, M. H., Gjorgiev, B., Das, L., Sansavini, G. (2022). Neural tangent kernel analysis of PINN for advection-diffusion equation. arXiv preprint arXiv:2211.11716.

9 References

[1] Marcelo Martins Werneck, Daniel Moreira dos Santos, Cesar Cosenza de Carvalho, Fábio Vieira Batista de Nazaré, and Regina Celia da Silva Barros Allil. Detection and monitoring of leakage currents in power transmission insulators. *IEEE sensors journal*, 15(3):1338–1346, 2014.

[2] J.Y. Li, C.X. Sun, W.X. Sima, and Q. Yang. Stage pre-warning based on leakage current characteristics before contamination flashover of porcelain and glass insulators. *IET Generation, Transmission & Distribution*, 3:605–615(10), July 2009.

[3] Jingyan Li, Wenxia Sima, Caixin Sun, and Stephen A. Sebo. Use of leakage currents of insulators to determine the stage characteristics of the flashover process and contamination level prediction. *IEEE Transactions on Dielectrics and Electrical Insulation*, 17(2):490–501, 2010.

[4] Isaias Ramirez, Ramiro Hernandez, and Gerardo Montoya. Measurement of leakage current for monitoring the performance of outdoor insulators in polluted environments. *IEEE Electrical Insulation Magazine*, 28(4):29–34, 2012.

[5] VH Ferreira, R Zanghi, MZ Fortes, GG Sotelo, RBM Silva, JCS Souza, CHC Guimarães, and S Gomes Jr. A survey on intelligent system application to fault diagnosis in electric power system transmission lines. *Electric Power Systems Research*, 136:135–153, 2016.

[6] Alok Mukherjee, Palash Kumar Kundu, and Arabinda Das. Transmission line faults in power system and the different algorithms for identification, classification and localization: a brief review of methods. *Journal of The Institution of Engineers (India): Series B*, pages 1–23, 2021.

[7] CA Apostolopoulos and GN Korres. Accurate fault location algorithm for double-circuit series compensated lines using a limited number of two-end synchronized measurements. *International Journal of Electrical Power & Energy Systems*, 42(1):495–507, 2012.

[8] Ying Zhang, Jun Liang, Zhihao Yun, and Xiaoming Dong. A new fault-location algorithm for series-compensated double-circuit transmission lines based on the distributed parameter model. *IEEE Transactions on Power Delivery*, 32(6):2398–2407, 2016.

[9] Biswajit Sahoo and Subhransu Ranjan Samantaray. An enhanced fault detection and location estimation method for TCSC compensated line connecting wind farm. *International Journal of Electrical Power & Energy Systems*, 96:432–441, 2018.

[10] Dong Wang, Mengqian Hou, and Yifei Guo. Travelling wave fault location of HVAC transmission line based on frequency-dependent characteristic. *IEEE Transactions on Power Delivery*, 2020.

[11] Abhishek Gupta, Ramesh Kumar Pachar, Baseem Khan, Om Prakash Mahela, Sanjeevikumar Padmanaban, and Fellow IET. A multivariable transmission line protection scheme using signal processing techniques. *IET Generation, Transmission & Distribution*, 15(22):3115–3137, 2021.

[12] Alok Mukherjee, Palash Kumar Kundu, and Arabinda Das. Classification and fast detection of transmission line faults using signal entropy. *Journal of The Institution of Engineers (India): Series B*, pages 1–16, 2021.

[13] Alok Mukherjee, Palash Kumar Kundu, and Arabinda Das. Classification and localization of transmission line faults using curve fitting technique with principal component analysis features. *Electrical Engineering*, pages 1–16, 2021.

[14] Enrico Zio. Prognostics and health management (phm): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering & System Safety*, 218:108119, 2022.

[15] Zhaoyi Xu and Joseph Homer Saleh. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. *Reliability Engineering & System Safety*, 211:107530, 2021.

[16] K Seethalekshmi, SN Singh, and SC Srivastava. A classification approach using support vector machines to prevent distance relay maloperation under power swing and voltage instability. *IEEE Transactions on Power Delivery*, 27(3):1124–1133, 2012.

[17] Ujjaval Patel, Nilesh Chothani, and Praghnesh Bhatt. Supervised relevance vector machine based dynamic disturbance classifier for series compensated transmission line. *International Transactions on Electrical Energy Systems*, 31(10):e12663, 2021.

[18] Koosha Marashi, Sahra Sedigh Sarvestani, and Ali R. Hurson. Identification of interdependencies and prediction of fault propagation for cyber–physical systems. *Reliability Engineering & System Safety*, 215:107787, 2021.

[19] Fezan Rafique, Ling Fu, and Ruikun Mai. End to end machine learning for fault detection and classification in power transmission lines. *Electric Power Systems Research*, 199:107430, 2021.

[20] Soufiane Belagoune, Noureddine Bali, Azzeddine Bakdi, Bousaadia Baadji, and Karim Atif. 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, 2021.

[21] Arash Moradzadeh, Hamid Teimourzadeh, Behnam Mohammadi-Ivatloo, and Kazem Pourhossein. Hybrid CNN-LSTM approaches for identification of type and locations of transmission line faults. *International Journal of Electrical Power & Energy Systems*, 135:107563, 2022.

[22] Alok Mukherjee, Kingshuk Chatterjee, Palash Kumar Kundu, and Arabinda Das. Probabilistic neural network-aided fast classification of transmission line faults using differencing of current signal. *Journal of The Institution of Engineers (India): Series B*, pages 1–14, 2021.

[23] Shahriar Rahman Fahim, Subrata K Sarker, SM Muyeen, Sajal K Das, and Innocent Kamwa. A deep learning based intelligent approach in detection and classification of transmission line faults. *International Journal of Electrical Power & Energy Systems*, 133:107102, 2021.

[24] Shahriar Rahman Fahim, Subrata Kumar Sarker, SM Muyeen, Md Rafiqul Islam Sheikh, Sajal K Das, and Marcelo Godoy Simoes. A robust self-attentive capsule network for fault diagnosis of series-compensated transmission line. *IEEE Transactions on Power Delivery*, 2021.

[25] Hanif Livani and C Yaman Evrenosoglu. A machine learning and wavelet-based fault location method for hybrid transmission lines. *IEEE Transactions on Smart Grid*, 5(1):51–59, 2013.

[26] Yann Qi Chen, Olga Fink, and Giovanni Sansavini. Combined fault location and classification for power transmission lines fault diagnosis with integrated feature extraction. *IEEE Transactions on Industrial Electronics*, 65(1):561–569, 2017.

[27] KM Silva, Benemar A Souza, and Nubia SD Brito. Fault detection and classification in transmission lines based on wavelet transform and ann. *IEEE Transactions on Power Delivery*, 21(4):2058–2063, 2006.

[28] Kunjin Chen, Caowei Huang, and Jinliang He. Fault detection, classification and location for transmission lines and distribution systems: a review on the methods. *High voltage*, 1(1):25–33, 2016.

[29] Paula Renatha N da Silva, Martin Max LC Negrão, Petrônio Vieira Junior, and Miguel A Sanz-Bobi. A new methodology of fault location for predictive maintenance of transmission lines. *International Journal of Electrical Power & Energy Systems*, 42(1):568–574, 2012.

[30] Martin Max LC Negrão, Paula Renatha N da Silva, Cristiane R Gomes, Hermínio S Gomes, Petrônio Vieira Junior, and Miguel A Sanz-Bobi. Mcho–a new indicator for insulation conditions in transmission lines. *International Journal of Electrical Power & Energy Systems*, 53:733–741, 2013.

[31] Paula Renatha Nunes da Silva, Hossam A Gabbar, Petrônio Vieira Junior, and Carlos Tavares da Costa Junior. A new methodology for multiple incipient fault diagnosis in transmission lines using QTA and Naïve Bayes classifier. *International Journal of Electrical Power & Energy Systems*, 103:326–346, 2018.

[32] Xuefeng Kong and Jun Yang. Reliability analysis of composite insulators subject to multiple dependent competing failure processes with shock duration and shock damage self-recovery. *Reliability Engineering & System Safety*, 204:107166, 2020.

[33] Mehdi Akbari Moghadam, Sajad Bagheri, Amir Hosein Salemi, and Mohammad Bagher Tavakoli. Long-term maintenance planning of medium voltage overhead lines considering the uncertainties and reasons for interruption in a real distribution network. *Reliability Engineering & System Safety*, page 109089, 2023.

[34] Blazhe Gjorgiev, Laya Das, Seline Merkel, Martina Rohrer, Etienne Auger, and Giovanni Sansavini. Simulation-driven deep learning for locating faulty insulators in a power line. *Reliability Engineering & System Safety*, 231:108989, 2023.

[35] Li Junfeng, Li Min, and Wang Qinruo. A novel insulator detection method for aerial images. In *Proceedings of the 9th International Conference on Computer and Automation Engineering*, pages 141–144, 2017.

[36] Xinyu Liu, Hao Jiang, Jing Chen, Junjie Chen, Shengbin Zhuang, and Xiren Miao. Insulator detection in aerial images based on faster regions with convolutional neural network. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, pages 1082–1086. IEEE, 2018.

[37] Hao Jiang, Xiaojie Qiu, Jing Chen, Xinyu Liu, Xiren Miao, and Shengbin Zhuang. Insulator fault detection in aerial images based on ensemble learning with multi-level perception. *IEEE Access*, 7:61797–61810, 2019.

[38] Jiaming Han, Zhong Yang, Qiuyan Zhang, Cong Chen, Hongchen Li, Shangxiang Lai, Guoxiong Hu, Changliang Xu, Hao Xu, Di Wang, et al. A method of insulator faults detection in aerial images for high-voltage transmission lines inspection. *Applied Sciences*, 9(10):2009, 2019.

[39] Chuanyang Liu, Yiquan Wu, Jingjing Liu, Zuo Sun, and Huajie Xu. Insulator faults detection in aerial images from high-voltage transmission lines based on deep learning model. *Applied Sciences*, 11(10):4647, 2021.

[40] Chuanyang Liu, Yiquan Wu, Jingjing Liu, and Zuo Sun. Improved yolov3 network for insulator detection in aerial images with diverse background interference. *Electronics*, 10(7):771, 2021.

[41] Haiyang Xia, Baohua Yang, Yunlong Li, and Bing Wang. An improved centernet model for insulator defect detection using aerial imagery. *Sensors*, 22(8):2850, 2022.

[42] Jiaming Han, Zhong Yang, Hao Xu, Guoxiong Hu, Chi Zhang, Hongchen Li, Shangxiang Lai, and Huarong Zeng. Search like an eagle: A cascaded model for insulator missing faults detection in aerial images. *Energies*, 13(3):713, 2020.

[43] Chunxue Shi and Yaping Huang. Cap-count guided weakly supervised insulator cap missing detection in aerial images. *IEEE Sensors Journal*, 21(1):685–691, 2020.

[44] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[46] Fan Li, Jianbo Xin, Tian Chen, Lijie Xin, Zixiang Wei, Yanglin Li, Yu Zhang, Hua Jin, Youping Tu, Xuguang Zhou, et al. An automatic detection method of bird's nest on transmission line tower based on faster_rcnn. *IEEE Access*, 8:164214–164221, 2020.

[47] MengYing Chen and Chen Xu. Bird's nest detection method on electricity transmission line tower based on deeply convolutional neural networks. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 2309–2312. IEEE, 2020.

[48] Shuaiang Rong and Lina He. A joint faster rcnn and stereovision algorithm for vegetation encroachment detection in power line corridors. In *2020 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2020.

[49] Lucas D. Simoes, Bruna L. Souza, Hagi J.D. Costa, Rodrigo P. De Medeiros, V. S. Orivaldo, and Flavio B. Costa. A Power Transformer Event Classification Technique Based on Support Vector Machine. *2020 Workshop on Communication Networks and Power Systems, WCNPS 2020*, (Wcnps), 2020.

[50] Lilia Tightiz, Morteza Azimi Nasab, Hyosik Yang, and Abdoljalil Addeh. An intelligent system based on optimized ANFIS and association rules for power transformer fault diagnosis. *ISA Transactions*, 103:63–74, 2020.

[51] Sunuwe Kim, Soo-ho Jo, Wongon Kim, Jongmin Park, Jingyo Jeong, Yeongmin Han, Daeil Kim, and Byeng Dong Youn. A Semi-Supervised Autoencoder With an Auxiliary Task (SAAT) for Power Transformer Fault Diagnosis Using Dissolved Gas Analysis. *IEEE Access*, 8:178295–178310, 2020.

[52] Transformers Committee, Ieee Power, and Energy Society. *IEEE Guide for the Interpretation*, volume 2019. 2019.

[53] Incipient fault diagnosis in power transformers by DGA using a machine learning ANN - Mean shift approach. *2019 IEEE International Autumn Meeting on Power, Electronics and Computing, ROPEC 2019*, (Ropec), 2019.

[54] Abdolrahman Peimankar, Stephen John Weddell, Thahirah Jalal, and Andrew Craig Lapthorn. Evolutionary multi-objective fault diagnosis of power transformers. *Swarm and Evolutionary Computation*, 36(March):62–75, 2017.

[55] R Sarathi, I. P.Merin Sheema, and R. Abirami. Partial discharge source classification by support vector machine. In *2013 IEEE 1st International Conference on Condition Assessment Techniques in Electrical Systems, IEEE CATCON 2013 - Proceedings*, pages 255–258. IEEE, 2013.

[56] Joao Bartolo Gomes, Hai-long Nguyen, Min Wu, Jianneng Cao, and Shonali Krishnaswamy. Active Learning for On-Line Partial Discharge Monitoring in Noisy Environments. pages 37–42, 2016.

[57] The Duong Do, Vo Nguyen Tuyet-Doan, Yong Sung Cho, Jong Ho Sun, and Yong Hwa Kim. Convolutional-Neural-Network-Based Partial Discharge Diagnosis for Power Transformer Using UHF Sensor. *IEEE Access*, 8:207377–207388, 2020.

[58] Venera Nurmanova, Yerbol Akhmetov, Mehdi Bagheri, Amin Zollanvari, Gevork B. Gharehpetian, and Toan Phung. A New Transformer FRA Test Setup for Advanced Interpretation and Winding Short-circuit Prediction. In *Proceedings - 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe, EEEIC / I and CPS Europe 2020*. Institute of Electrical and Electronics Engineers Inc., jun 2020.

[59] Ali Reza Abbasi, Mohammad Reza Mahmoudi, and Zakieh Avazzadeh. Diagnosis and clustering of power transformer winding fault types by crosscorrelation and clustering analysis of FRA results. *IET Generation, Transmission and Distribution*, 12(19):4301–4309, oct 2018.

[60] A. Abu-Siada, Mohamed I. Mosaad, Dowon Kim, and Mohamed F. El-Naggar. Estimating power transformer high frequency model parameters using frequency response analysis. *IEEE Transactions on Power Delivery*, 35(3):1267–1277, 2020.

[61] Ali Naderian Jahromi, Ray Piercy, Stephen Cress, Jim R.R. Service, and Wang Fan. An approach to power transformer asset management using health index. *IEEE Electrical Insulation Magazine*, 25(2):20–34, 2009.

[62] Shuaibing Li, Guangning Wu, Haiying Dong, Lei Yang, and Xiaofei Zhen. Probabilistic health index-based apparent age estimation for power transformers. *IEEE Access*, 8:9692–9701, 2020.

[63] Ricardo Manuel Arias Velásquez, Jennifer Vanessa Mejía Lara, and Andres Melgar. Converting data into knowledge for preventing failures in power transformers. *Engineering Failure Analysis*, 101(January):215–229, 2019.

[64] V M Catterson. Prognostic modeling of transformer aging using Bayesian particle filtering. In *2014 IEEE Conference on Electrical Insulation and Dielectric Phenomena, CEIDP 2014*, pages 413–416. IEEE, 2014.

[65] Masoud Pourali and Ali Mosleh. A Bayesian approach to online system health monitoring. In *Proceedings - Annual Reliability and Maintainability Symposium*, pages 2–7. IEEE, 2013.

[66] Haroldo De Faria, João Gabriel Spir Costa, and Jose Luis Mejia Olivas. A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis, 2015.

[67] Huo Ching Sun, Yann Chang Huang, and Chao Ming Huang. Fault diagnosis of power transformers using computational intelligence: A review. *Energy Procedia*, 14:1226–1231, 2012.

[68] Juan José, Montero Jimenez, Sébastien Schwartz, Rob Vingerhoeds, Bernard Grabot, and Michel Salaün. Towards multi-model approaches to predictive maintenance : A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems*, 56(July):539–557, 2020.

[69] Saraa I Khalel, Mohd Fadli Rahmat, and Mohd Wazir Bin Mustafa. Sensoring leakage current to predict pollution levels to improve transmission line model via ANN. *International Journal of Electrical & Computer Engineering (2088-8708)*, 7(1), 2017.

[70] Jared Garrison, Blazhe Gjorgiev, Xuejiao Han, Renger H. van Nieuwkoop, Elena Raycheva, Marius Schwarz, Xuqian Yan, Turhan Demiray, Gabriela Hug, Giovanni Sansavini, and Christian Schaffner. Nexus-e: Input data and system setup. Technical report, Bern, 2020-11-27.

[71] Matlab-Simulink. Simulation and model-based design - Simscape/Electrical/Specialized power systems, 2022.

[72] Blazhe Gjorgiev, Bing Li, and Giovanni Sansavini. Calibration of cascading failure simulation models for power system risk assessment. In *Proceedings of the 28th International European Safety and Reliability Conference, ESREL 2019*, page 6, 2019.

[73] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[74] Benjamin Maschler and Michael Weyrich. Deep transfer learning for industrial automation: a review and discussion of new techniques for data-driven machine learning. *arXiv preprint arXiv:2012.03301*, 2020.

[75] Kirtan Jha, Aalap Doshi, Poojan Patel, and Manan Shah. A comprehensive review on automation in agriculture using artificial intelligence. *Artificial Intelligence in Agriculture*, 2:1–12, 2019.

[76] Pramila P Shinde and Seema Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.

[77] Dongxia Zhang, Xiaoqing Han, and Chunyu Deng. Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE Journal of Power and Energy Systems*, 4(3):362–370, 2018.

[78] Bo Luo, Haoting Wang, Hongqi Liu, Bin Li, and Fangyu Peng. Early fault detection of machine tools based on deep learning and dynamic identification. *IEEE Transactions on Industrial Electronics*, 66(1):509–518, 2018.

[79] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[80] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 817–825, 2016.

[81] Chiao-Ling Kuo and Ming-Hua Tsai. Road characteristics detection based on joint convolutional neural networks with adaptive squares. *ISPRS International Journal of Geo-Information*, 10(6):377, 2021.

[82] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 769–776. IEEE, 2017.

[83] Hang Yu, Laurence T Yang, Qingchen Zhang, David Armstrong, and M Jamal Deen. Convolutional neural networks for medical image analysis: state-of-the-art, comparisons, improvement and perspectives. *Neurocomputing*, 444:92–110, 2021.

[84] Andrei-Alexandru Tulbure, Adrian-Alexandru Tulbure, and Eva-Henrietta Dulf. A review on modern defect detection models using dcnns–deep convolutional neural networks. *Journal of Advanced Research*, 35:33–48, 2022.

[85] Sandeep Sony, Kyle Dunphy, Ayan Sadhu, and Miriam Capretz. A systematic review of convolutional neural network-based structural condition assessment techniques. *Engineering Structures*, 226:111347, 2021.

[86] Nasser Kehtarnavaz. Chapter 7 - frequency domain processing. In Nasser Kehtarnavaz, editor, *Digital Signal Processing System Design (Second Edition)*, pages 175–196. Academic Press, Burlington, second edition edition, 2008.

[87] Paul S. Addison. Introduction to redundancy rules: the continuous wavelet transform comes of age. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2126):20170258, 2018.

[88] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[89] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$0.5mb model size, 2016.

[90] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[91] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.

[92] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

[93] MathWorks. Spoken digit recognition with custom log spectrogram layer and deep learning. https://ch.mathworks.com/help/signal/ug/spoken-digit-recognition-with-custom-log-spectrogram-layer-and-deep-learning.html, 2022. Accessed on 2022-02-07.

[94] MATLAB. *9.10.0.1851785 (R2021a)*. The MathWorks Inc., Natick, Massachusetts, 2021.

[95] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3369–3378, 2018.

[96] Thomas Peter Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[97] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[98] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[99] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[100] Meng Lan, Yipeng Zhang, Lefei Zhang, and Bo Du. Defect detection from uav images based on region-based cnns. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 385–390. IEEE, 2018.

[101] Lei Ma, Changfu Xu, Guoyu Zuo, Bin Bo, and Fengbo Tao. Detection method of insulator based on faster r-cnn. In *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 1410–1414. IEEE, 2017.

[102] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[103] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[104] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, February 2022.

[105] P Kulkarni, T Shaw, and D Lewis. Insulator defect image dataset - version 1.2: Documentation EPRI, Palo Alto, CA: 2020. 3002017949.

[106] Jin Li, Daifu Yan, Kuan Luan, Zeyu Li, and Hong Liang. Deep learning-based bird's nest detection on transmission lines using uav imagery. *Applied Sciences*, 10(18):6147, 2020.

[107] André Luiz Buarque Vieira-e Silva, Heitor de Castro Felix, Thiago de Menezes Chaves, Francisco Paulo Magalhães Simões, Veronica Teichrieb, Michel Mozinho dos Santos, Hemir da Cunha Santiago, Virginia Adélia Cordeiro Sgotti, and Henrique Baptista Duffles Teixeira Lott Neto. Stn plad: A dataset for multi-size power line assets detection in high-resolution uav images. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 215–222. IEEE, 2021.

[108] Ms Aayushi Bansal, Dr Rewa Sharma, and Dr Mamta Kathuria. A systematic review on data scarcity problem in deep learning: solution and applications. *ACM Computing Surveys (CSUR)*, 54(10s):1–29, 2022.

[109] Dexter Lewis and Pratik Kulkarni. Insulator defect detection, 2021.

[110] Jin Li, Daifu Yan, Kuan Luan, Zeyu Li, and Hong Liang. Supplementary Files: Deep Learning-Based Bird's Nest Detection on Transmission Lines Using UAV Imagery, September 2020.

[111] Label Studio. LabelImg.

[112] Laya Das, Mohammad Hossein Saadat, Blazhe Gjorgiev, Etienne Auger, and Giovanni Sansavini. Object detection-based inspection of power line insulators: Incipient fault detection in the low data-regime, 2022.

[113] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38, 2021.

[114] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. Anomaly detection in blockchain networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2022.

[115] Waleed Hilal, S Andrew Gadsden, and John Yawney. Financial fraud:: A review of anomaly detection techniques and recent advances. 2022.

[116] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.

[117] Christian Velasco-Gallego and Iraklis Lazakis. Radis: A real-time anomaly detection intelligent system for fault diagnosis of marine machinery. *Expert Systems with Applications*, 204:117634, 2022.

[118] Tristan Schnell, Katrin Bott, Lennart Puck, Timothée Buettner, Arne Roennau, and Rüdiger Dillmann. Robigan: A bidirectional wasserstein gan approach for online robot fault diagnosis via internal anomaly detection. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4332–4337. IEEE, 2022.

[119] Yu Chen, Zhongyong Zhao, Hanzhi Wu, Xi Chen, Qianbo Xiao, and Yueqiang Yu. Fault anomaly detection of synchronous machine winding based on isolation forest and impulse frequency response analysis. *Measurement*, 188:110531, 2022.

[120] Yukun Fang, Haigen Min, Xia Wu, Xiaoping Lei, Shixiang Chen, Rui Teixeira, and Xiangmo Zhao. Toward interpretability in fault diagnosis for autonomous vehicles: Interpretation of sensor data anomalies. *IEEE Sensors Journal*, 2023.

[121] Chenxi Li, Yongheng Yang, Kanjian Zhang, Chenglong Zhu, and Haikun Wei. A fast mppt-based anomaly detection and accurate fault diagnosis technique for pv arrays. *Energy Conversion and Management*, 234:113950, 2021.

[122] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

[123] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.

[124] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941, 2017.

[125] Philipp Liznerski, Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus Robert Muller. Explainable deep one-class classification. In *International Conference on Learning Representations*, 2021.

[126] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[127] Kentaro Wada. Labelme: Image Polygonal Annotation with Python.

[128] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[129] IEEE. Ieee guide for the interpretation of gases generated in mineral oil-immersed transformers. *IEEE Std C57.104-2019 (Revision of IEEE Std C57.104-2008)*, pages 1–98, 2019.

[130] M. Duval. A review of faults detectable by gas-in-oil analysis in transformers. *IEEE Electrical Insulation Magazine*, 18(3):8–17, 2002.

[131] Michel Duval and Laurent Lamarre. The duval pentagon-a new complementary tool for the interpretation of dissolved gas analysis in transformers. *IEEE Electrical Insulation Magazine*, 30(6):9–12, 2014.

[132] W. Wattakapaiboon and N. Pattanadech. The state of the art for dissolved gas analysis based on interpretation techniques. In *2016 International Conference on Condition Monitoring and Diagnosis (CMD)*, pages 60–63, 2016.

[133] Atefeh Dehghani Ashkezari, Tapan K. Saha, Chandima Ekanayake, and Hui Ma. Evaluating the accuracy of different dga techniques for improving the transformer oil quality interpretation. In *AUPEC 2011*, pages 1–6, 2011.

[134] N.A. Muhamad, B.T. Phung, T.R. Blackburn, and K.X Lai. Comparative study and analysis of dga methods for transformer mineral oil. In *2007 IEEE Lausanne Power Tech*, pages 45–50, 2007.

[135] Ambra Maria Van Liedekerke. Coordinated maintenance planning and congestion management for the swiss power grid: Scheduling algorithms and topological action optimization. Master thesis, ETH Zurich, 2022.