Available online at www.sciencedirect.com
ScienceDirect
Optical Fiber Technology 00 (2025) 1–16
Optical Fiber Technology

Invited Paper

Addressing Data Scarcity in ML-Based Failure-Cause Identification in Optical Networks through Generative Models

Meadhbh Healy[a], Andreas Baum[a], Francesco Musumeci[b]

[a]*Department of Statistics and Data Analysis, Technical University of Denmark*
[b]*Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano*

Abstract

We consider the issue of data scarcity with class imbalance in failure-cause identification for optical fiber systems using Machine Learning (ML) techniques. We use an open dataset comprising of real Optical Time-Domain Reflectometer (OTDR) traces which have been gathered in an artificial setup spanning tens of kilometers, consistent with a long-haul network. Whilst ML methods have shown satisfactory results for automating the process of identifying failure causes in optical fiber networks, the solutions are generally strongly dependent on available labeled datasets, and require extensive data to train and validate any findings. However, in the case of failure management in optical networks, building a valuable dataset with sufficiently informative samples is in general a hard process, due to the fact that, by nature, failures occur infrequently. As such, data-labelling is time and resource intensive for domain experts. We therefore seek to mitigate these issues by exploring two generative models, namely, conditional Generative Adversarial Network (cGAN) and conditional Variational Autoencoder (cVAE), to balance the number of failures samples in a multiclass dataset. In order to balance the dataset with accurate synthetic data across the different failure causes, we adopt generative models that are conditioned on the failure classes, the SNR level of the trace and the maximum amplitude of the signal. These approaches are compared to Synthetic Minority Over-sampling TEchnique (SMOTE). We compare our approaches by training our datasets using an autoencoder classifier and testing them against three holdout datasets. Results show that, with the cGAN and cVAE, failure-cause identification can be improved by more than 5% in terms of global accuracy when compared to the imbalanced dataset, and in particular for scarcely-represented failure classes, our generative models provide an improvement in the f1 scores of over 50%.

Keywords: Failure-Cause Identification, Optical Networks, OTDR, Data Augmentation, Conditional GAN, Conditional VAE, SMOTE

1. Introduction

Optical fiber performs a crucial role in transmitting vast amounts of data across the Internet, serving as a fundamental medium for high-speed data transfer. A single fiber link can connect thousands of end users [14]. It is imperative that any disruption in these fiber links is identified and resolved as soon as possible so that network breakdowns can be minimized, and the ability to quickly understand the root phenomenon generating the failure is key in the failure management process. The continuous operation of optical fiber networks presents a significant challenge for health moni-

Email addresses: mhea@dtu.dk (Meadhbh Healy), andba@dtu.dk (Andreas Baum), francesco.musumeci@polimi.it corresponding author (Francesco Musumeci)

toring. Optical fiber systems generate vast amounts of diagnostic and performance monitoring data. When information is extracted from this data, it can enable the operators to not only observe the conditions of the system but also to track changing traffic situations [15]. At the same time, monitoring optical fiber systems can be a complex process due to both the highly complex system and the large amounts of data available [16]. The limited availability of labelled failure data in optical fiber systems has been a significant barrier to the successful automation of failure identification. Machine Learning-(ML-) based failure-cause identification can achieve reasonable results in optical networks and systems [27], [28]. However, the performance hinges on the availability of extensive, high-quality training data. Acquiring real failure samples is a resource-intensive task, often requiring specialized equipment for network monitoring, and domain experts for data annotation. Furthermore, the rarity of system failures combined with the potential privacy issues of failure data for network operators poses additional challenges. Imbalanced datasets can often perform poorly as any classification algorithm can show preference for the majority class [29]. Consequently, we propose a data augmentation strategy to mitigate the challenges associated with limited failure data availability. Data augmentation can be described as the creation of artificial instances that are indistinguishable from the real samples. By augmenting the samples in failure minority classes, better classification performance for failure-cause identification is targeted. However, traditional augmentation techniques, such as flipping, rotating, or applying geometric transformations, may prove inadequate [1] in some cases, as they often generate data samples without the necessary diversity to simulate realistic failure samples and can lead to insufficient model performance improvement or even to performance deterioration.

To surmount these limitations, generative modeling has emerged as a promising solution. Generative models such as Generative Adversarial Networks (GANs) and Variational AutoEncoders (VAEs) can generate highly complex and realistic synthetic data. They aim to learn the underlying distribution of the data in order to generate samples that retain the fundamental characteristics. They also offer scalability and flexibility to generate a large amount of synthetic data effectively. GANs have previously been used in communications systems to augment training data for a wireless jammer [2]. GANs comprise of two Neural Networks (NNs), namely the Generator G and Discriminator D, which operate on an adversarial basis where the Generator attempts to develop better quality samples in order to challenge the discriminator's ability to differentiate between real and synthetic. VAEs, on the other hand, work in a collaborative manner. They also consist of two networks; an Encoder, which is an inferential network that learns representations of the data and a Decoder, a generative network which reconstructs the data based on the Encoder's representations [3].

In this paper, we propose the utilization of two generative models, i.e., conditional GANs (cGANs) and conditional VAEs (cVAEs), to augment Optical Time Domain Reflectometry (OTDR) traces for failure-cause identification in optical networks. cGANs and cVAEs are considered a extension of the GAN and VAE framework, respectively, whereby the inputs to each NN are conditioned on auxiliary contextual information, such as class labels, making them effective for generating samples in imbalanced datasets [4]. As failures are usually rare occurrences in the data, there is generally a small amount of real high quality data available. Our intention, in this paper, is to exploit the superior learning power of conditional generative models to synthesize high quality labelled data leveraging the knowledge from a small amount of real failure data.

2. Related Work

2.1. ML-based Failure Management in Optical Networks

Artificial Intelligence (AI) and ML techniques can play a significant role in the surveillance and monitoring of the extensive amounts of data created in an optical fiber system. These algorithms can help collate and automate the vast amounts of monitoring data and process it in real-time, enabling a rapid diagnosis of failures and issues in the network [17]. ML approaches can mitigate complex processes by iteratively learning on input data and then providing feedback to the network operator. Many previous studies have focused on either using supervised or unsupervised methods to deal with problems in both the domain of optical networks and optical transmission [18]. Additionally, various metrics extracted from the optical fiber network can point to various failures in the domain. Optical spectrometer data, such as BER and OSNR data, for example, can indicate degradation of the lightpath as it travels from the source to destination node in the optical system [19].

Optical spectrum analysis effectively monitors system performance by identifying multiple types of distortions, enabling precise failure detection and prediction [32]. Mayer et. al [20] leverage ML to localize failures in an optical system. They use data from a physical

topology system of an optical network, routed lightpaths and telemetry data, such as OSNR and BER, as inputs to an NN which constructs all possible scenarios present in an optical network and uses any potential failures in the system as inputs to train the NN to localize these failures. The NN is iteratively updated after training from the database to localize probable failures. The authors further refine their approach by performing Principle Component Analysis (PCA) to reduce the amount of non-essential parameters inputted to the NN and speed up the training process of the model. Similarly, Kruse et. al [21] employ NNs with optical spectrum data when performing failure detection in an optical system.

The authors in [21] train a VAE-based GAN to identify unknown failure types. By comparing their approach to other ML methods, they not only conclude that their techniques outperform these algorithms but enable better results with significantly less training data. This is a significant advancement in optical network failure management as the labelling of real-time training data by domain experts is not always feasible, and in general it is extremely time-consuming.

Musumeci et. al provide an alternative solution to data scarcity in [22] by adopting transfer learning. More specifically, ML models are trained on a dataset where sufficient data is available for different types of failures occurring within a lightpath and collected from a so-called *source domain* (e.g., a laboratory testbed or a digital twin). The trained models are then deployed on an different domain, i.e., the *target domain*, where a limited amount of data can be used to fine tune the pretrained models in order to adapt them to the specific characteristics of the new domain. An unsupervised domain adaptation method, called correlation alignment (CORAL), is also used in [22] to align the feature distribution of the observations in the source and target domains.

2.2. Addressing Data Scarcity in Failure Management

The focus of this work is primarily to improve the performance of ML-based failure-cause identification in the specific conditions of data scarcity and class imbalance within the available labeled data. Conventional ML techniques will often fail to produce satisfactory results on imbalanced data as they are inclined towards bias in the majority class.

Existing work has been already completed to overcome this limitation. E.g., the work of Sun et. al [30] utilizes a stacked ensemble ML model to account for imbalanced data in the network. The authors posit that integrating multiple ML models enables an exploitation of the strengths of each model, whilst counteracting the weakness. They measure the performance of their ensemble model against the individual models it is comprised of with improved results. Cichosz et. al [29] adopt ML to assess whether the quality of transmission in a Dense Wavelength Division Multiplexing (DWDM) network is sufficient or not. The ML model is applied to a database representative of real DWDM networks as used by operators. As such, the number of failure samples is relatively small. To remedy this problem, they employ one-class ML techniques, assuming the training data is comprised of a single class, which then serves to detect instances which do not belong to this class. While the authors recognize the results as positive, they also acknowledge the small size of the failure data makes it difficult to assess the definitive success of their approach.

Other approaches may seek to effectively train anomaly detection classifiers on limited labelled samples or a support set using techniques such as siamese networks or deviation networks [8]. These methods are known as few-shot learning. The authors of [39] employ one such pre-training strategy for data-annotation in an optical transport network (OTN). They use the isolation forest (IF) technique and density-based spatial clustering of applications with noise (DBSCAN) to isolate and identify imbalanced failure samples in the dataset and label them. They demonstrate the superior results of their method when compared to unsupervised labelling and threshold based methods.

2.3. Leveraging Synthetic Data Generation in Failure Management

Creating failures synthetically has also been touted as a possible solution to balance the minority class. In [31], the authors oversample the minority class using SMOTE algorithm [7] to balance the classes. They measure each threshold using a range of ML classifiers and conclude that the best results occur with the most balanced dataset. In [26], the authors examine a range of both model-centric and data-centric ML techniques to alleviate the issue of data scarcity. The model-centric approaches involve the alteration of the loss function to help balance and identify underrepresented classes, and mini-batch training where the imbalanced training set inputted into the NN in a more balanced manner to account for the undersampled observations. For data-centric approaches, the authors applied augmentation techniques such as SMOTE, SMOTE-TOMEK and the CT-GAN to increase the representation of the undersampled failures. All methods were trained using the same NN tuned by the same hyperparameters.

Although it works well even on highly imbalanced datasets, the performance of SMOTE is dependent on the complexity of samples being generated, due to the possible lack in diversity between the synthetically-generated data.

Generative models [33], [34] have emerged as a promising alternative for generating synthetic data, particularly synthetic failure samples in various domains such as manufacturing, healthcare, and system maintenance [36]. The models can create realistic and diverse data samples that closely mimic the characteristics of real-world samples, including the complex and often rare failure conditions critical for testing and improving diagnostic algorithms [9].

Kruse et. al [40] motivate the use of a VAE and a GAN to augment real-world data samples, such as OSNR and BER, for more accurate failure identification. They calculate the optimum amount of samples to be generated in order to achieve an f1 score of 0.9 or higher in a range of classifiers. They demonstrate that the utilization of their augmentation techniques mitigates the need for real world failure samples for successful soft failure identification. Khan et. al [41] adopt a similar approach by using a VAE and then a synthetic sample selector (SSS) to select a subset of the most imbalanced classes. The SSS selects samples by minimizing their Euclidean distance from the mean of that class in the imbalanced experimental dataset. They discover that they achieve better results with a balanced dataset as a result of augmentation and a less complex NN, than can be achieved with an imbalanced dataset and a more complex NN. In [34], the authors use both a VAE and a GAN, where the VAE is used to extract the most important features of the data which the GAN then uses to generate samples that balance the classes. They find that their method outperforms both SMOTE and an individual GAN to generate failure samples. In this research, although the authors use SMOTE as a method of comparison to their VAE, they criticize the use of SMOTE for oversampling, claiming that the technique indiscriminately creates samples and thus many samples are created around the decision boundary, skewing the original distribution of the data. They claim that using a generative model such as a VAE and creating samples from the latent space allows a greater control over the distribution of the data.

The notion of controlling the distribution or the output can be extended further. Conditioning in generative models, such as Conditional VAEs (cVAEs) and Conditional GANs (cGANs), enables the generation of data samples based on specified conditions or attributes [35] [13]. This approach enhances model flexibility and applicability, allowing for targeted outputs that meet predefined characteristics . Generally, conditioning involves providing the model with additional contextual information, often categorical like class type, alongside random noise [37], [11]. This is particularly useful for balancing minority classes [10]. For instance, Ezeme et al. , [12], used conditioning to generate failure samples for a minority class in a cGAN, aiding a cluster-based anomaly detector.

2.4. Failure Management using OTDR Data

OTDR data is used ubiquitously in fiber optical systems for failure location and quality evaluation of signals. Optical fibers can be susceptible to numerous types of failures and degradation, due to events occurring at the physical layer, e.g., fiber cut, equipment malfunctioning, ageing, etc., or to external factors, such as malicious fiber eavesdropping [6]. These can result in a compromised performance in the system. The manual identification of failure instances requires a significant amount of expertise as well as being time consuming. OTDR operates on the principle of Raleigh backscattering and is used extensively for the purpose of fiber failure identification and localization. The equipment transmits a number of fiber pulses into the fiber link being monitored. The reflected signals are then registered as a function of time, which can then be computed as a position on the optical fiber [5]. Traditional event detection in OTDR traces include Short-Time Fourier Transform (STFT) methods and wavelet transform methods [23], however these techniques decrease in efficacy when OTDR have a lower SNR, and are more prone to being noisy. Yang et. al [24] propose an event detection methods for raw OTDR traces that excludes much of the preprocessing that occurs in transform methods and instead differentiates and denoises the trace before inputting it into two machine learning classifiers: Random Forest and Gradient Boosting. OTDR enables specific failure identification as different failures manifest themselves differently in the optical system. Nyarko-Boateng et. al [25] utilize an LSTM based NN to detect and localize fiber cuts, a prolific cause of optical fiber failure, in the optical system. The work of Abdelli et. al [5] is also pivotal regarding the automation of OTDR event detection using ML techniques. The authors use a bidirectional GRU autoencoder, in order to classify OTDR traces each representing a particular failure in the optical domain. This work is later extended to use the same autoencoder for multi-task performance, the tasks being failure identification, failure characterisation and failure localisation [6]. The

authors use additional parameters which are concatenated to the OTDR sequences in order to optimize the performance of the classifier. These parameters are the SNR and the Maximum Amplitude of the signal.

2.5. Paper Contribution

In this work, we aim to address the issue of data scarcity and class imbalance in optical network failure-cause identification using OTDR data. We explore generative models cGAN and cVAE to generate diverse samples of optical fiber failures and compare them to SMOTE-based data generation to quantify the impact of high-quality data generation on the performance of failure-cause identification. As OTDR traces for the same failure type can vary depending on the SNR level [23], this work will seek to use learnable conditional embeddings which not only generate OTDR traces for a particular failure type but also account for real world variability such as the noise level of the signal.

3. Dataset Description

In this paper, we consider an open dataset [38], comprising of segmented OTDR traces which encompass various types of failures in an optical fiber system.

Table 1. Categorization of Failures in Fiber Optic Systems.

	Failure Type	**Description**	**# Samples**
1	Fiber Tapping	Security breaches due to unauthorized tapping	16000
2	Bad Splice	Incorrect cable joining causing signal loss	16000
3	Bending Event	Excessive bending of fibers leading to signal attenuation	16000
4	Dirty Connector	Signal loss due to contaminated connectors	16000
5	Fiber Cut	Complete severance of the fiber	13849
6	PC Connector	Connector finish imperfections causing reflections	16000
7	Reflector	Events causing light to reflect back towards the source	15935

The dataset consists of 8 classes, including one class with "normal" (i.e., non-faulty) behaviour and 7 classes associated with typical failures occurring in an optical fibre network. All classes, including the "normal" class, are used to train the classifier and the generative models. There are approximately 16,000 observations for each of the classes. The description of each failure type is reported in Table 1.

Each observation is structured as follows:

- **Trace Sequences:** Every OTDR trace is segmented into sequences that are scaled between 0 and 1. These sequences have been extracted from the original OTDR trace of a variable length and divided into fixed lengths of 30. An example of each segmented trace is available at Figure 1.

- **Class:** The failure type, which is denoted by an integer value from 0 to 7.

- **Signal-to-Noise Ratio (SNR):** An SNR value, calculated for each sequence, quantifies the signal quality. The SNR values in this dataset range between 0 and 30 dB and are continuous.

- **Maximum Amplitude:** The variable "Maximum Amplitude" is computed from the dataset values. It is calculated using the maximum of the values in P1 to P30, (the maximum peak signal), which is divided by the event location. It is included in order to provide the model with a metric that captures the intensity or strength of the event and can be useful to make more accurate distinctions between "Dirty Connector" and "PC Connector" for example, which might otherwise have been challenging based on the shape and pattern of the events. The "Maximum Amplitude" value ranges from 0 to 50 in this dataset and the values are continuous. They are factorized in order to be absorbed into the embedding layers.

4. ML-based Failure-Cause Identification and Data Augmentation Approaches

In this paper, we address failure-cause identification in optical networks using OTDR data and focus on the data scarcity problem under the specific condition of data imbalance. In the following subsections, we detail our approach and discuss the ML-based classifier used to implement failure-cause identification and the different approaches for data augmentation.

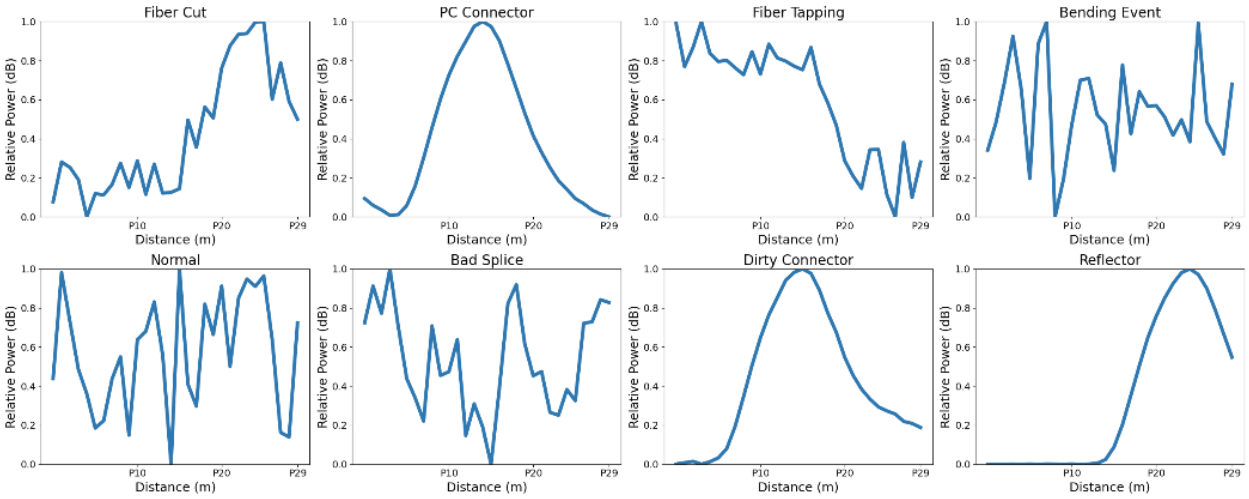

Figure 1. Segmented OTDR traces for seven failure classes and normal behaviour.

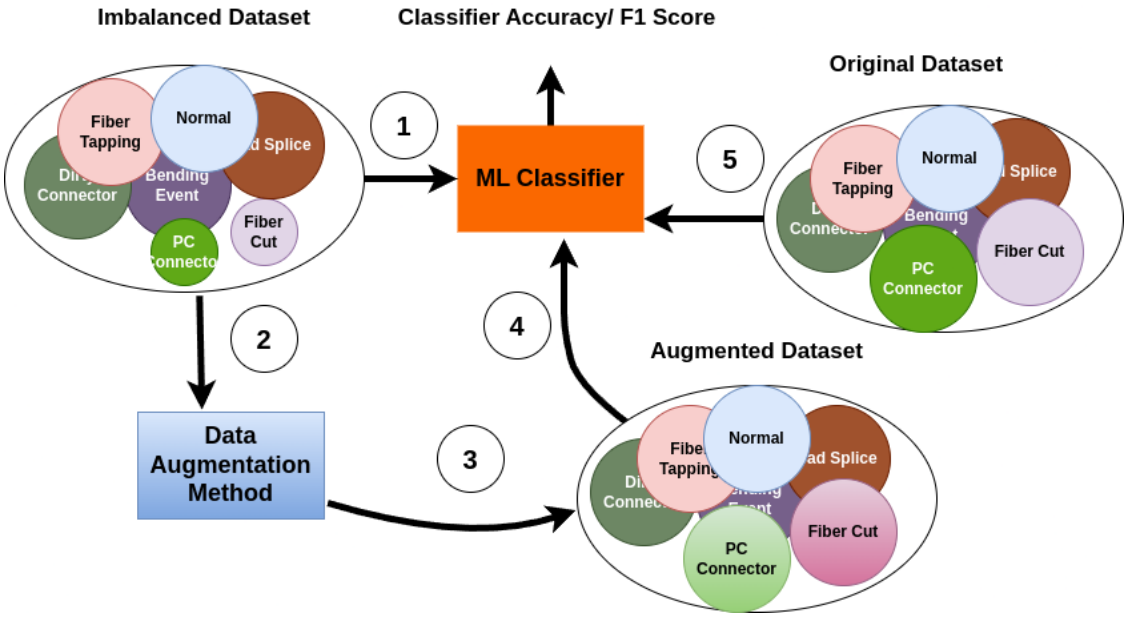

Figure 2. Work-flow diagram of training and testing process.

4.1. Approach Overview

The goal of data augmentation is to improve the classification and generalization outcome of a ML model with respect to the case when it is trained using imbalanced data. The overall framework of our process can be observed in Figure 2 and is summarized below.

1. To motivate our approach, we input the imbalanced dataset to our ML classifier. The imbalanced dataset contains six classes with 16000 samples each, whilst two classes ”Fiber Cut” and ”PC Connector” and have approximately 1600 samples each, or 10% of the other classes.
2. We then perform the three data augmentation methods on the imbalanced datasets. The generative models, cGAN and cVAE, take the entire dataset as an input, whereas SMOTE takes the two minority classes only.
3. Each augmented dataset is inputted into the ML classifier and tested against three holdout datasets, the results of which are averaged and a margin of error is calculated. These three holdout

dataset contain eight classes, each with 3200 samples, that were created from the original dataset prior to training.

4. Finally, the original dataset, with all eight balanced classes, is inputted into the ML classifier for comparison.

4.2. ML Classifier

We design a classifier to distinguish between the failure classes in our dataset with a high degree of accuracy. The architecture of the classifier is an autoencoder comprised of GRU layers, followed by one fully connected layer. This architecture, as well as the number of neurons in each layer, has been influenced by the work of Abdelli et. al in [6]. GRU layers are chosen for their ability to process short sequences and create a low-dimensional representation of the input. The GRU layer of the encoder and decoder consist of 30 and 15 neurons respectively. The fully connected layer has 16 neurons and outputs an integer between 0 and 7, depending on whatever class it classifies the failure as. For each class in each dataset, the input to the ML classifier is a 32-length sequence; the length of the OTDR trace, the 'SNR' value, and the 'Maximum Amplitude' value of the trace. The architecture of the ML classifier can also be seen in Figure 3.

4.3. Conditional Embeddings

Due to the inherent variability of real-world OTDR traces, the generative models condition the inputs on the 'Class', 'SNR' and 'Maximum Amplitude' variable. The lower the 'SNR' value of the trace, the noisier it is. The 'Maximum Amplitude' of the trace helps distinguish between similar events, such as PC Connector and Dirty Connector. Embeddings enable a representation of data in a low dimensional space, which is typically dense and continuous. By mapping the conditional information to embeddings, a more nuanced representation is provided and the network can learn more complex patterns more effectively. As well as this, embeddings allow for greater flexibility as they can be fine tuned to suit the requirements of the task. In the following, we present the methodologies that we adopt to perform data augmentation to address data scarcity in the specific case of high class imbalance, namely, conditional GAN and conditional VAE, which are compared against a more traditional methodology relying on SMOTE.

4.4. Conditional GAN

The GAN is a generative ML model, which generates samples that are indistinguishable from real data. The Generator, G, typically takes random noise as an input to generate instances and the Discriminator, D, attempts to differentiate between the real and generated samples. The function of the Discriminator is to maximise the probability of assigning the correct labels to real and generated instances and whilst the Generator is trained to minimize the loss term $log(1\text{-}D(G(z))$, where z is the random noise input to the Generator.

The standard GAN model is unsupervised learning and thus cannot create samples of a specific labelled class. GAN does not have control over the samples generated and can only map from the random noise input z to a target output x. However, in numerous failure identification tasks, the datasets featuring failures tend to be imbalanced. Consequently, a model that can generate labeled data presents a more viable option in these scenarios. In this paper, we propose the utilization of a cGAN to effectively overcome the limitation and control the type of data the model is producing. The cGAN will be conditioned on three different pieces of contextual information; the failure type, the SNR value and the Maximum Amplitude value.

In an expansion of the original GAN framework, both the Generator and Discriminator are conditioned on the conditional information, y, which acts as an extension of the latent space z in order to synthesize and distinguish generated data [7]. The architecture of the cGAN can be observed in Figure 4, which uses the contextual information Y_syn to train the networks.

A detailed description of how the cGAN is used to augment OTDR traces is as follows. The conditional information, 'Class', 'SNR' and 'Max Amplitude' is embedded and concatenated to the random noise input z before being passed to the first layer of the Generator. The conditional information guides the generator to generate specific instances. The discriminator also takes the conditional information embeddings and, concatenated with either a real or synthetic OTDR trace, acts as a classifier and outputs a value of 0 or 1, depending on whether it determines the trace as being real or fake. The generator takes the evaluation from the discriminator to generate new OTDR traces that are closer to the real traces. The generator continues to refine its process until the discriminator can no longer differentiate between real and synthetic traces.

After considering a number of different architectures, we identified the most effective architecture as being inspired by our baseline classifier, and the work

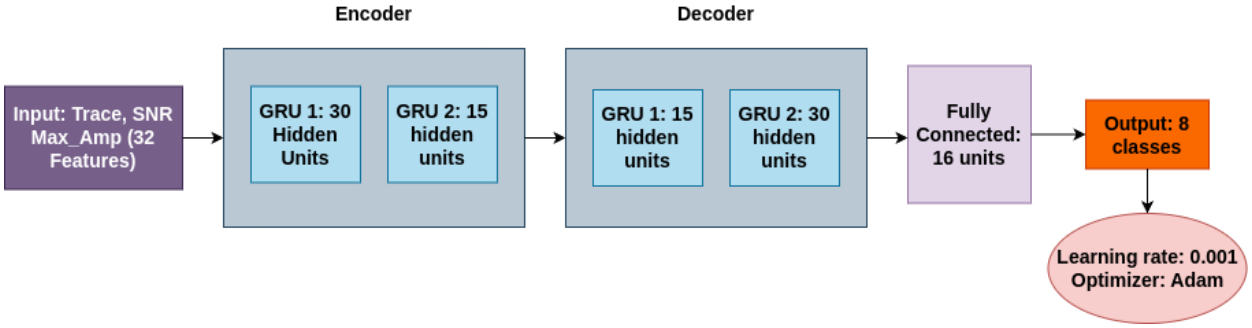

Figure 3. The ML Classifier Architecture.

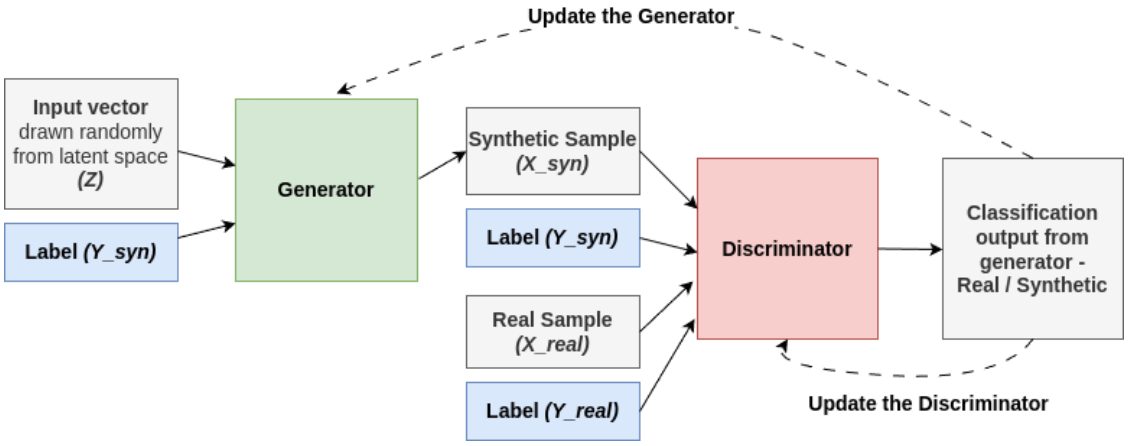

Figure 4. The cGAN Architecture.

of Abdelli et. al [5], [6]. It therefore comprises of GRU layers followed by fully connnected layers. The generator network initializes the embedding layer for the class labels, SNR values and Maximum Amplitude values. These embeddings are concatenated to a latent noise vector z of dimension 100 the first of two unidirectional layers which takes the input and increases to 128. The second layer takes the input of 128 and increase it to 256. This is followed by two fully connected layers which process the input of 256, and reduces it to 64. The last layer outputs a sequence of 32 which is the length of a single trace, the SNR value and the Maximum Amplitude value for that trace. The discriminator network contains a similar structure of the three embedding layers, two GRU layers and two fully connected layers. The first layer takes a 32 length sequence which represents a single trace, (either real or synthetic), and concatenates it with the embedding vectors. The GRU layers in the discriminator are bidirectional so that the sequence can be processed from both directions. The first GRU layer has an input and output of 32, and due to bidirectionality, the second layer has an input of a 64 and an output of 64. The first fully connected layer has an input of 128 and an output of 32 and the second has an output of 32 to a single unit 1, a classification output determining whether the output is real or synthetic. The learning rate of both NNs is 0.0001. The discriminator also has a weight decay of 0.00001, and Adam optimizer is used for both networks.

Figure 5 and 6 illustrate the generator and discriminator model respectively. The loss function of the generator is displayed in Equation 1.

$$\mathcal{L}_G = \cdot\mathbb{E}_{z,y}[\log(D(G(z,y)))] \tag{1}$$

The loss function for the discriminator is displayed in Equation 2.

$$\mathcal{L}_D = -\mathbb{E}_{x,y}[\log(D(x,y))] - \mathbb{E}_{z,y}[\log(1 - D(G(z,y),y))] \tag{2}$$

where D is the discriminator G is the generator z is the random noise input to the generator and y is the

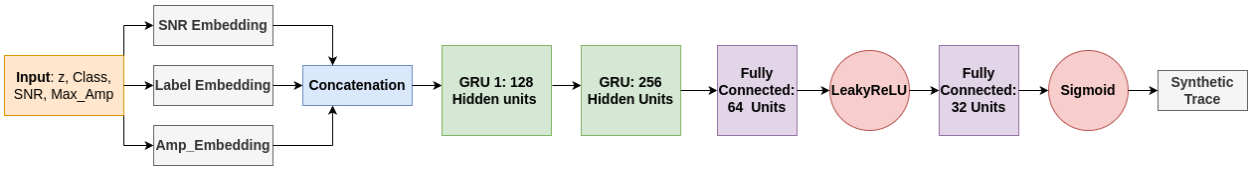

Figure 5. The Generator Model.

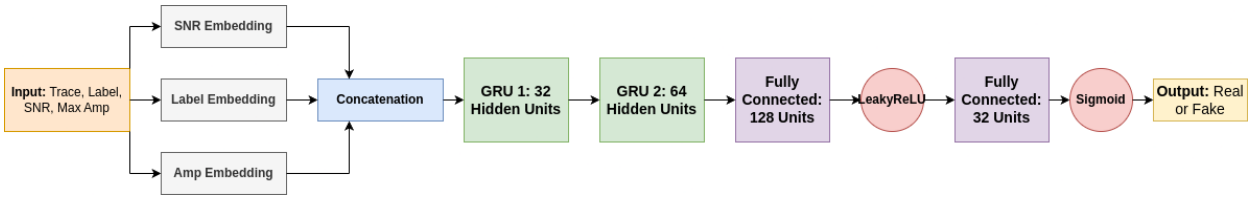

Figure 6. The Discriminator Model.

conditioning information. The cGAN operates as an adversarial loss function where the goal of the generator is to fool the discriminator into classifying the generated samples as real samples. The discriminator acts as a critic and provides information to the generator regarding how the generated samples differ from the real ones.

4.5. Conditional VAE

Similar to a GAN, a VAE consists of two networks. However, unlike a GAN, the two networks in a VAE do not compete against each other. Instead, the encoder compresses the input data into a low dimensional representation to capture the fundamental components of the data. This is then reconstructed from the latent space by the decoder. In a traditional autoencoder the latent space is discrete, however a VAE endeavours to take these discrete values and construct a known distribution of the latent space. Therefore, both encoder and decoder networks are probabilistic. The encoder is a probabilistic function $q_\Phi(x|z)$ that approximates the true (but intractable) posterior distribution of the latent variable z given an input x. It is parametrized by Φ. This is then reconstructed by the decoder, a probabilistic function $p_\Theta(z|x)$, that generates data in the latent space z. It is parameterized by Θ [8]. VAEs are trained by minimizing the Evidence Lower Bound (ELBO) loss, as observed in Equation 3:

$$ELBO = \underbrace{-logp_\Theta(z|x)}_{\text{MLE Reconstruction}} + \underbrace{D_{KL}(q_\Phi(x|z)||p(z))}_{\text{KL Regularisation}} \quad (3)$$

where the D_{KL} is the Kullback Leibler Divergence used to quantify the differences between the two probability function q and p. Th MLE Reconstruction term ensures that the generated samples are faithful to the original distribution of the data.

It is also possible for the VAE be extended in a manner comparable to the GAN, and for it to be conditioned on contextual information. As the sole difference between the VAE and cVAE is the addition of contextual information, the computational and optimizational methods in a cVAE are the same as those in a VAE. Therefore, the loss function of the cVAE can be described as in Equation 4:

$$ELBO = \underbrace{-logp_\Theta(z|x,y)}_{\text{MLE Reconstruction}} + \underbrace{D_{KL}(q_\Phi(x|z,y)||p(z))}_{\text{KL Regularisation}} \quad (4)$$

.

As with the VAE, samples are generated from a prior distribution and then inserted into the probabilistic decoder conditioned on y. The architecture of the cVAE can be observed in Figure 7.

The architecture of the cVAE used to synthesize the OTDR traces is as follows. The encoder network consists of 2 bidirectional GRU layers, the first of which takes the length the concatenation of trace length plus the embedding dimensions as an input and outputs 128 features in each direction, making it 256 features in total due to bidirectionality. The second layer takes 256 features from the first layer and outputs 128 features in each direction, making it a total of 256 features. This is followed by a fully connected layer sequence comprising of 2 linear layers and 2 leakyReLU activation functions. The first linear layer inputs 256 neurons and outputs 128 and the second inputs 128 and outputs 240 neurons. The encoder maps the input data to the latent space, which is represented by the mean

and the logarithm of the variance, the outputs of the encoder. As the latent space is modelled probabilistically, the reparametrization trick is used for backpropagation of the gradients. The reparametrization trick restructures the sampling process by sampling from a standard Gaussian distribution.

Once the encoder has learned a latent representation of the input, it is up to the decoder to reconstruct the original data. The structure of the decoder model closely mirrors the structure of the encoder with 2 GRU layers followed by a sequence of fully connected layers and non-linear activation functions. The GRU layers in the decoder are unidirectional, as the traces can only be generated one element at a time, the first takes in the latent dimension of size 120 and outputs 256 neurons. The second layer takes 256 neurons and outputs 128 neurons. The first linear layer reduces it back to 64 and the second outputs it to 30 for the correct output size. The structure of the cVAE encoder layers can be observed in Figure 8 and the decoder layers can be observed in Figure 9.

4.6. SMOTE

SMOTE is used ubiquitously to approach a class imbalance in a dataset, and create synthetic samples for a minority class [9]. Essentially, SMOTE generates new samples by performing a random linear interpolation between a sample in the minority class and their neighbouring samples. SMOTE selects an instance a at random and calculates their k nearest neighbours. One of these neighbours b is then randomly selected and a synthetic sample is created as a convex combination between the chosen instances a and b. The type of SMOTE used in this research is SMOTE-ENC, due to the combination of numerical, (the 30 length sequence of the trace), and categorical (the SNR and Maximum Amplitude) features that are used as an input into the ML classifier. The number of k-nearest neighbors to define the neighborhood of samples to used to generate the synthetic samples is five.

5. Research Motivation

5.1. Class Selection

All the datasets are used to train the same baseline classifier with all of the same tuned hyperparameters. The augmented datasets are also tested against the same holdout three datasets. This approach allows us to comprehensively assess the impact of data augmentation on the classifier's ability to identify different failure types under varying data augmentation schemes. The datasets are compared under the global accuracy, precision, recall and f1 score results. The f1 scores are per-class f1 scores, which provides a measure of the models performance on each class individually. The ML classifier is trained on the original dataset, with eight failure classes comprised of 16000 samples each. The Original dataset obtains a global accuracy of 92.6%. The results of precision, recall and f1 scores for each of the eight classes is recorded in Table 2.

Class Label	Precision	Recall	F1
Normal	1.0	1.0	1.0
Fiber Tapping	0.99	0.98	0.99
Bad Splice	0.98	0.96	0.97
Bending Event	0.92	0.92	0.92
Dirty Connector	0.99	0.95	0.97
Fiber Cut	0.74	0.88	0.80
PC Connector	0.94	0.83	0.81
Reflector	0.95	0.99	0.97

Table 2. Performance Metrics by Class Label for Original Dataset.

Motivated by the fact that the poorest performance results are obtained by the classes "Fiber Cut" and "PC Connector", we specifically reduce the number of samples in these classes.

5.2. Sensitivity Analysis

The sensitivity analysis presented in Figure 10 highlights the critical role of data augmentation in enhancing model performance across varying levels of available training data, using the "PC Connector" class as a representative example. The "Amount of Training Data" axis represents the proportion of data available in the class, with smaller values corresponding to increasingly challenging scenarios due to reduced data availability.

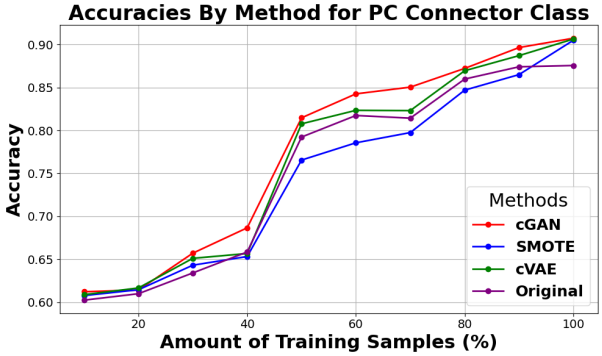

Figure 10. Sensitivity Analysis of Class Accuracy for Different Amounts of Training Samples for PC Connector Class.

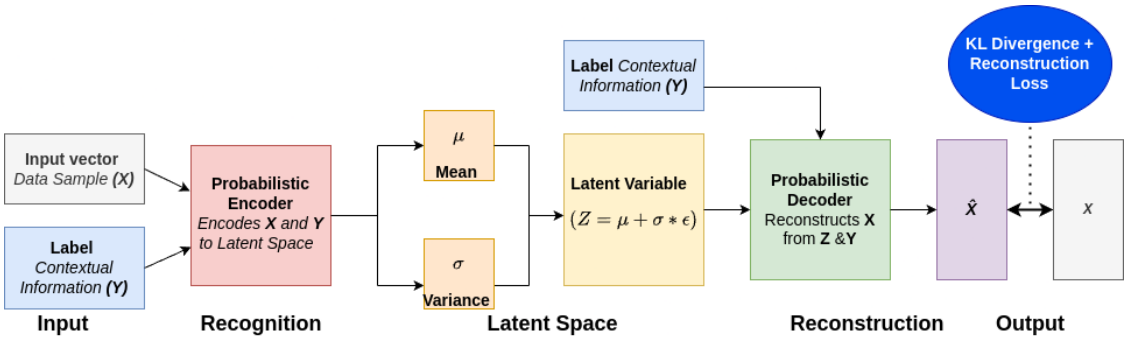

Figure 7. The cVAE Architecture.

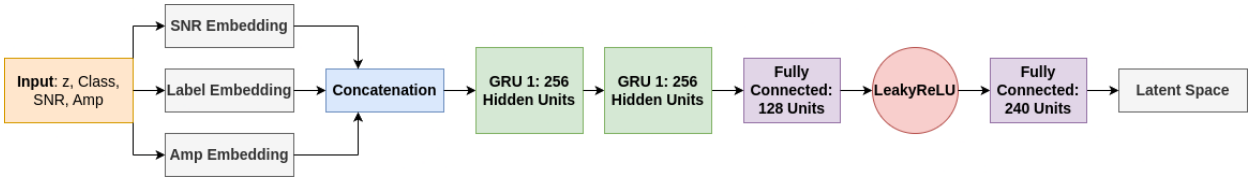

Figure 8. The Encoder Model.

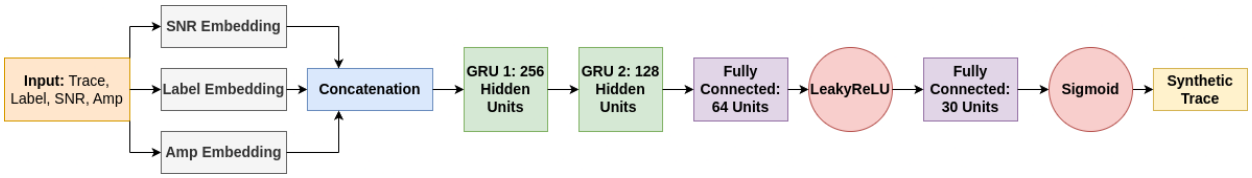

Figure 9. The Decoder Model.

For this analysis, data availability was set to 10% of the original training samples (approximately 1,600 samples), representing the most extreme scenario of data scarcity. This decision was motivated by the need to evaluate model performance under critical and challenging conditions for the classifier. While data availability levels of 20% or 30% (corresponding to less severe data scarcity) provide useful insights, they represent less challenging scenarios. By focusing on the most demanding condition, this analysis rigorously examines the ability of data augmentation techniques to address extreme data imbalance.

Although the sensitivity analysis is performed on the "PC Connector" class, similar trends were observed across other classes. This motivated the decision to use one class as a representative example to ensure clarity and conciseness. Following the data reduction, the classifier is retrained on the reduced dataset, allowing us to analyze the effects of classification under imbalanced conditions and evaluate how well augmentation strategies mitigate these challenges. This approach ensures that the impact of augmentation methods is assessed under the most critical conditions, providing a comprehensive evaluation of their effectiveness.

6. Numerical Results

6.1. Global Performance

In Table 3 records the results of the global accuracy for each data augmentation approach are compared to the two benchmark scenarios: the Imbalanced dataset and the Original dataset. The augmentation strategies have been tested against the three holdout datasets and averaged. The margin of error was calculated using a t-confidence interval threshold of 95%. The global accuracy, in multiclass classification, is a measure of how many instances across all classes, are correctly identified from the dataset.

Table 3. Global Accuracy of five Training Datasets.

Training Set	Accuracy (%)
Original	92.4 ± 0
Imbalanced	97.6 ± 0
SMOTE	94.59 ± 0
cGAN	95.96 ± 0.2
cVAE	96.62 ± 0.4

The Original dataset achieves the lowest global accuracy with a score of 92.4%. The Imbalanced dataset achieves an accuracy score of 97.6%. This occurs as the Imbalanced dataset has a lower amount of samples overall, (as the two minority classes in the training data contain only 1600 samples each whereas the Original dataset have 16000 samples in each class). Therefore, the majority classes dominate classification even on the holdout test set, the Imbalanced dataset has a lower number of misclassifications. The relatively high accuracy score for the Imbalanced dataset can also be attributed to the fact that there are eight failure classes in the entire dataset, six of which are approximately 16000 samples each, meaning that the accuracy metric will be skewed towards the performance of the more frequent classes. This is further enforced by the analysis of the other per-class performance metrics, which are recorded in Table 4.

Class Label	Precision	Recall	F1
Normal	1.0	1.0	1.0
Fiber Tapping	0.98	0.99	0.99
Bad Splice	0.99	0.97	0.98
Bending Event	0.96	0.99	0.98
Dirty Connector	0.97	1.0	0.98
Fiber Cut	0.69	0.22	0.34
PC Connector	0.78	0.60	0.68
Reflector	0.98	0.99	0.99

Table 4. Performance Metrics by Class Label.

It can be observed that the per-class metrics for the six real classes remain largely unaltered from the performance metrics obtained in the Original dataset. For some metrics they have risen slightly, such as the improved precision and recall scores for the "Bending Event" class from the scores in the Original dataset in Table 2. However, it can be seen that the performance metrics for the minority classes are significantly reduced. The recall score for the "Fiber Cut" class falls from 0.88 to 0.22, a 75% reduction, suggesting that the classifier is having difficulty capturing the diversity in that class. The recall score for "PC Connector" decreases from 0.83 to 0.60, which is a considerable difference from the recall scores achieved in the six majority classes. The f1 scores for the "PC Connector" class is 0.68 which is 30% less than any of the majority classes. The f1 score for the "Fiber Cut" class is 0.34, motivation a need for data augmentation to improve this class performance in the ML classifier. The cGAN, cVAE and SMOTE methods to balance the dataset achieve a higher accuracy than the case where the Original non-augmented dataset is used. It can be noted that as 16000 samples are generated by each augmentation method, meaning that the number of samples for the Fiber Cut class is increased from 13849 in the Original dataset 1. cVAE has the highest global accuracy of the generative models and SMOTE with a value of 96.62%, indicating that this model has the fewest misclassifications. The cGAN has less than a 1% difference in global accuracy when compared to the cVAE, also indicating a strong performance in global accuracy. The SMOTE augmentation method has a global accuracy of 94.59%, which is over 2% greater than the Original dataset.

6.2. Per-class Performance

6.2.1. Fiber Cut

Table 5 records the metrics precision, recall and f1 Score of the Original and Imbalanced dataset, as well as the cGAN, cVAE and SMOTE datasets.

Table 5. Per-class performance metrics for Fiber Cut.

Class	Precision	Recall	F1 Score
Original	0.73 ± 0	0.88 ± 0	0.80 ± 0
Imbalanced	0.69 ± 0.01	0.22 ± 0.01	0.34 ± 0.01
SMOTE	0.87 ± 0.01	0.88 ± 0	0.88 ± 0
cGAN	0.86 ± 0.01	0.95 ± 0.01	0.91 ± 0.01
cVAE	0.85 ± 0	0.97 ± 0.01	0.90 ± 0.01

The Original dataset has a reasonably good balance between precision and recall, leading to an f1 Score of 0.80. The absence of variance (± 0) suggests consistent performance across iterations. The Imbalanced dataset performs poorly, particularly in recall (0.22 ± 0.01), indicating that the model struggles to identify positive instances. This contributes to a very low f1 Score (0.34 ± 0.01). The slight variance indicates some inconsistency across iterations. SMOTE significantly improves both precision and recall, when compared to both the Original and Imbalanced dataset. This results in a high f1 Score (0.88). The slight variance in precision suggests minor inconsistency, but recall and f1 Score are very consistent. The cGAN shows strong performance across all metrics, with a high recall (0.95 ± 0.01) and

f1 Score (0.91 ± 0.01). The cGAN performs very well, with the highest recall (0.97 ± 0.01) among the methods, leading to a high f1 Score (0.90 ± 0.01). This indicates that the cVAE generated the most diverse traces for the Fiber Cut class. Whilst the cVAE and cGAN generate traces with a similar precision to SMOTE, they both have a higher recall illustrating their ability to generate diverse samples. All the augmentation methods (SMOTE, cGAN, cVAE) improve performance metrics significantly over the Original dataset. The Original dataset already has reasonable performance but can be improved further with augmentation.

To evaluate the performance of the models under class imbalance, precision-recall (PR) curves were generated, with detailed analysis provided for critical classes such as Fiber Cut in Figure 11 and PC Connector in Figure 12. Precision-recall curves are particularly suitable for Imbalanced datasets, as they highlight the trade-offs between precision (positive predictive value) and recall (sensitivity) without being biased by the predominance of majority classes.

It can be observed that the cGAN has the best balance between precision and recall, maintaining a high precision across a wide range of recall values. At a precision value of 0.7, the recall value of the cGAN samples are 0.82, the highest value of all the methods. The cVAE dataset has an earlier dropoff than the cGAN but has a recall value of 0.78 when the precision is 0.7. SMOTE also maintains high precision however it decreases at an earlier point than cGAN and cVAE. At a precision value of 0.7, it has a recall value of 0.75. The Imbalanced dataset illustrates a significant drop in precision as recall increases, indicating poor performance in terms of maintaining high precision. At a precision value of 0.7, it has a recall value of approximately 0.3. Whilst the Original dataset performs better than the Imbalanced dataset, the precision still decreases more rapidly than cGAN, cVAE and SMOTE. The Original dataset has a recall value of 0.68, when precision is 0.7.

Both cGAN and cVAE demonstrate a strong capacity to maintain high Precision while increasing Recall, indicating robust augmentation performance by the generative models. While SMOTE has a superior to the Original and Imbalanced dataset, it decreases precision sooner, showing a less powerful performance when compared to cGAN and cVAE, especially in areas of high recall, showing that the simpler augmentation method struggles to create more diverse samples.

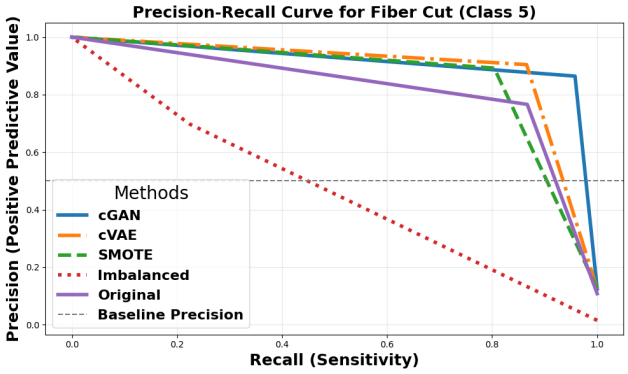

Figure 11. Precision-Recall Curve for Fiber Cut.

6.2.2. PC Connector

Table 6 records the per-class performance metrics for the PC Connector class. The Original dataset shows high precision and good recall, leading to an f1 Score of 0.81. The variance is ± 0 which suggests consistent performance across iterations. The Imbalanced dataset performs poorly, particularly in recall (0.60), indicating that the model struggles to identify positive instances. This results in a low f1 Score of 0.68. Again, SMOTE shows improved performance over the Original dataset, with balanced precision and recall, resulting in a high f1 Score (0.88). cVAE demonstrates strong performance, with a high precision (0.96 ± 0.01) and balanced recall, leading to a high f1 Score (0.90 ± 0.01). cGAN shows the highest precision (0.98 ± 0.0) among all methods, with balanced recall, again resulting in a high f1 Score (0.90 ± 0.01). SMOTE has a slightly higher recall value than cGAN or cVAE, indicating that it is slightly better at generating diverse samples for this class.

Table 6. Per-class performance metrics for PC Connector.

Class	Precision	Recall	F1 Score
Original	0.94 ± 0	0.83 ± 0	0.81 ± 0
Imbalanced	0.78 ± 0	0.6 ± 0	0.68 ± 0
SMOTE	0.92 ± 0	0.85 ± 0	0.88 ± 0
cVAE	0.96 ± 0.01	0.84 ± 0.01	0.9 ± 0.01
cGAN	0.98 ± 0	0.82 ± 0.01	0.9 ± 0.01

The precision-recall curve for the PC Connector class can be seen in Figure 12. cGAN achieves the best overall balance between precision and recall for this class. It consistently maintains high precision over a wide range of recall values, with only a slight decline at very high recall levels. At a precision value of 0.7, the recall of the cGAN dataset is approximately 0.8, the highest of all the methods. The cVAE has a sim-

ilar performance to the cGAN, however the precision value decreases more gradually at an earlier point. At a precision value of 0.7, the cVAE has a recall value of 0.75, slightly lower than a cGAN. SMOTE has a high precision but decreases earlier than the cVAE and more gradually than the cGAN. At a precision of 0.7, SMOTE reaches a recall of 0.7, which is lower than both cGAN and cVAE. This demonstrates that both the cGAN and cVAE are capable of generating the most realistic synthetic samples, enabling the classifier to maintain a low-false positive rate even as it identifies more true positives. Whilst SMOTE generates useful synthetic samples, its simplistic oversampling method limits its utility for high recall scenarios. The Original dataset achieves a recall score of 0.6 when precision is 0.7, however this is inferior to all three augmentation methods. The Imbalanced dataset achieves only of a recall score of 0.45 at precision 0.7, highlighting the necessity of augmentation for this class.

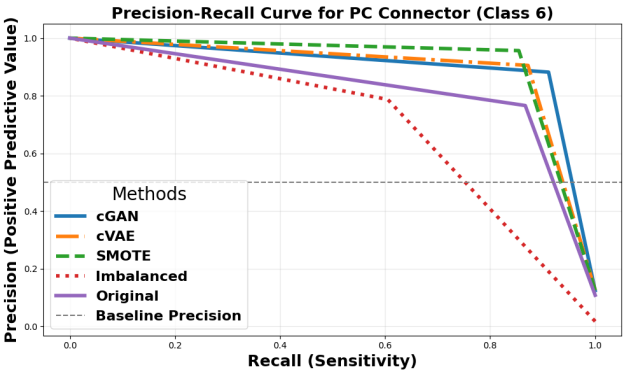

Figure 12. Precision-Recall Curve for PC Connector.

6.3. Computational Complexity Analysis

The cGAN, cVAE and SMOTE have a significantly different time required in order to generate accurate samples. The generative models, cGAN and cVAE involve a training process to optimize the generation process, whereas SMOTE does not. It can be seen in Table 7 that the cGAN takes the most time to train. This difference arises because the cGAN trains for 1000 epochs, whereas the cVAE trains for 500. The cGAN employs an adversarial loss function, where the generator and discriminator compete against each other, requiring more time for the training process to stabilize. In contrast, the cVAE utilizes a more straightforward loss function, where training concludes once the loss stops decreasing. The equations for the cGAN and cVAE loss functions are presented in Equations 1, 2, and 4, respectively. Additionally, the cGAN is a more complex model, as reflected by its higher number of parameters shown in Table 7, which further contributes to its longer training time compared to the cVAE.

The time taken to generate samples is also recorded in Table 7. It can be observed that SMOTE has the highest generation time to generate 16000 samples per class. The cGAN has the lowest inference time with 0.04 seconds.

Approach	Training Time (s)	Generation Time (s)	No. Hyper-Parameters
cGAN	20019.19	0.04	14
cVAE	4676.73	0.14	10
SMOTE	0	0.82	3

Table 7. Generation Time of cGAN, cVAE and SMOTE.

The augmentation methods also have an optimum number of hyperparameters which are used to refine the generation process. It can be observed in Table 7 that the SMOTE method uses the least amount of hyperparameters, using a sampling strategy, a k-nearest neighbours value and a random seed. The cGAN requires the most hyperparameters to optimize. This is due to the fact that, in the cGAN, two neural networks are being trained to compete against each other and each require a separate set of hyperparameters. Both generative models also are conditioned on three embeddings which are also inputted as hyperparameters to be optimized.

7. Conclusion

We study the impact of data scarcity and possible countermeasures when addressing failure-cause identification in optical networks using OTDR data. This research was motivated by the high operational cost of collecting and labelling data in a fiber optical system. Three different data augmentation techniques are compared to generate realistic and diverse OTDR traces, namely, two generative models (cGAN and cVAE) and the SMOTE method. We apply ML-based failure-cause identification to a multiclass dataset comprising seven failure-type classes and one ”normal” (i.e., non-faulty) class. We evaluate the performance of the three aforementioned data generation approaches compared to the case of Imbalanced dataset as a benchmark scenario.

We found that the classification performance of the minority classes, Fiber Cut and PC Connector, in the Imbalanced dataset was extremely poor. By applying our methods, we found that the cVAE had an increase of 75% in recall over the Imbalanced dataset. SMOTE had an increase in precision of 27% for the same class. The

cGAN had the highest f1 score of the datasets, with an improvement 57%, compared to the Imbalanced dataset. For the PC Connector class, SMOTE improved the recall compared to the Imbalanced dataset by 25%, cGAN had the highest precision with a 20% improvement on the imbalance dataset, and both cGAN and cVAE had a 22% improvement in f1 score compared to the imbalance dataset. We believe that our contributions merit further research. In this work, we endeavour to balance the various failure classes in our dataset meaning that we generate 16000 samples for each of the minority classes. We begin by creating a scenario of extreme data scarcity with an imbalanced dataset of 10%, which we motivate by performing sensitivity analysis on a single class. However, future studies could explore the minimum amount of samples required to increase accuracy and improve classification. It is clear that the use of data augmentation to improve classification in optical fiber failure data can play a pivotal role in both improving model performance and reducing the need for failure-data collection and annotation.

Acknowledgements

This work was partly supported by the PRIN project ZeTON, funded by Italian Ministry of University and Research. This research was also partially funded by the Otto Mønsted Foundation, Denmark.

References

[1] Lei, C., Hu, B., Wang, D., Zhang, S., & Chen, Z. (2019). A preliminary study on data augmentation of deep learning for image classification. Proceedings of the 11th Asia-Pacific Symposium on Internetware.

[2] Erpek, T., Sagduyu, Y. E., & Shi, Y. (2019). Deep learning for launching and mitigating wireless jamming attacks. IEEE Transactions on Cognitive Communications and Networking, 5(1), 2–14.

[3] Liu, C., Antypenko, R., Sushko, I., & Zakharchenko, O. (2022). Intrusion Detection System after Data Augmentation Schemes Based on the VAE and CVAE. IEEE Transactions on Reliability, 71(2), 1000–1010. https://doi.org/10.1109/TR.2022.3164877

[4] Patel, M., Wang, X., & Mao, S. (2020). Data augmentation with conditional GAN for automatic modulation classification. Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning.

[5] Abdelli, K., Cho, J. Y., Azendorf, F., Griesser, H., Tropschug, C., & Pachnicke, S. (2022). Machine-learning-based anomaly detection in optical fiber monitoring. Journal of Optical Communications and Networking, 14(5), 365. https://doi.org/10.1364/jocn.451289

[6] Abdelli, K., Azendorf, F., Grießer, H., Tropschug, C., & Pachnicke, S. (2021). Gated Recurrent Unit-based Autoencoder for Optical Link Fault Diagnosis in Passive Optical Networks. European Conference on Optical Communication (ECOC), 2021, 1–4. https://doi.org/10.1109/ECOC52684.2021.9605969

[7] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

[8] Belton, N., Hagos, M.T., Lawlor, A., & Curran, K.M. (2023). FewSOME: Few Shot Anomaly Detection. ArXiv, abs/2301.06957.

[9] Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences, 479, 448–459. https://doi.org/10.1016/j.ins.2017.12.030

[10] Elaraby, N., Barakat, S., & Rezk, A. (2022). A conditional GAN-based approach for enhancing transfer learning performance in few-shot HCR tasks. Scientific Reports, 12(1), 13115. https://doi.org/10.1038/s41598-022-20654-1

[11] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 5967–5976). Honolulu, HI, USA. https://doi.org/10.1109/CVPR.2017.632

[12] Ezeme, O. M., Mahmoud, Q. H., & Azim, A. (2020). Design and Development of AD-CGAN: Conditional Generative Adversarial Networks for Anomaly Detection. IEEE Access, 8, 177667–177681. https://doi.org/10.1109/ACCESS.2020.3025530

[13] De Donno, C., Hediyeh-Zadeh, S., Moinfar, A. A., Wagenstetter, M., Zappia, L., Lotfollahi, M., & Theis, F. J. (2023). Population-level integration of single-cell datasets enables multi-scale analysis across samples. Nature Methods, 20(11), 1683–1692. https://doi.org/10.1038/s41592-023-02035-2

[14] Wellbrock, G. A., Xia, T. J., Huang, M.-F., Han, S., Chen, Y., Wang, T., & Aono, Y. (2023). Explore benefits of distributed fiber optic sensing for optical network service providers. Journal of Lightwave Technology, 41(12), 3758–3766. https://doi.org/10.1109/jlt.2023.3263795

[15] Nevin, J. W., Nallaperuma, S., Shevchenko, N. A., Li, X., Faruk, Md. S., & Savory, S. J. (2021, December 29). Machine learning for Optical Fiber Communication Systems: An introduction and overview. AIP Publishing. https://doi.org/10.1063/5.0070838

[16] Zhu, C., Alsalman, O., & Naku, W. (2023). Machine learning for a vernier-effect-based optical fiber sensor. Optics Letters, 48(9), 2488. https://doi.org/10.1364/ol.489471

[17] Reyes-Vera, E., Valencia-Arias, A., García-Pineda, V., Aurora-Vigo, E. F., Alvarez Vásquez, H., & Sánchez, G. (2024). Machine learning applications in optical fiber sensing: A research agenda. Sensors, 24(7), 2200. https://doi.org/10.3390/s24072200

[18] Gu, R., Yang, Z., & Ji, Y. (2020). Machine learning for Intelligent Optical Networks: A comprehensive survey. Journal of Network and Computer Applications, 157, 102576. https://doi.org/10.1016/j.jnca.2020.102576

[19] Wang, D., Zhang, C., Chen, W., Yang, H., Zhang, M., & Lau, A. P. (2022). A review of machine learning-based failure management in Optical Networks. Science China Information Sciences, 65(11). https://doi.org/10.1007/s11432-022-3557-9

[20] Mayer, K. S., Soares, J. A., Pinto, R. P., Rothenberg, C. E., Arantes, D. S., & Mello, D. A. (2021). Machine-learning-based soft-failure localization with partial software-defined networking telemetry. Journal of Optical Communications and Networking, 13(10). https://doi.org/10.1364/jocn.424654

[21] Kruse, L. E., Kühl, S., Dochhan, A., & Pachnicke, S. (2024). Experimental investigation of machine-learning-based soft-failure management using the optical spectrum. Jour-

nal of Optical Communications and Networking, 16(2), 94. https://doi.org/10.1364/jocn.500930

[22] Musumeci, F., Venkata, V. G., Hirota, Y., Awaji, Y., Xu, S., Shiraiwa, M., Mukherjee, B., & Tornatore, M. (2021). Domain adaptation and transfer learning for failure detection and failure-cause identification in optical networks across different light-paths [invited]. Journal of Optical Communications and Networking, 14(2). https://doi.org/10.1364/jocn.438269

[23] Lunglmayr, M., & Amaral, G. C. (2019). Linearized Bregman Iterations for Automatic Optical Fiber Fault Analysis. IEEE Transactions on Instrumentation and Measurement, 68(10), 3699–3711. https://doi.org/10.1109/TIM.2018.2882258

[24] Yang, Z., Hong, D., Feng, X., & Xie, J. (2021). A novel event detection method for OTDR trace with high sensitivity based on machine learning. 2021 2nd Information Communication Technologies Conference (ICTC). https://doi.org/10.1109/ictc51749.2021.9441614

[25] Nyarko-Boateng, O., Adekoya, A. F., & Weyori, B. A. (2021). Tracing the exact location of failures in underground optical networks using LSTM deep learning model. Indian Journal of Science and Technology, 14(4), 297–309. https://doi.org/10.17485/ijst/v14i4.2008

[26] Khan, L. Z., Pedro, J., Costa, N., Sgambelluri, A., Napoli, A., & Sambo, N. (2024). Model and data-centric machine learning algorithms to address data scarcity for failure identification. Journal of Optical Communications and Networking, 16(3), 369. https://doi.org/10.1364/jocn.511863

[27] Silva, M. F., Pacini, A., Sgambelluri, A., & Valcarenghi, L. (2022). Learning long- and short-term temporal patterns for ML-driven fault management in optical communication networks. IEEE Transactions on Network and Service Management, 19(3), 2195–2206. https://doi.org/10.1109/tnsm.2022.3146869

[28] Chen, X., Liu, C.-Y., Proietti, R., Li, Z., & Yoo, S. J. B. (2022). Automating optical network fault management with machine learning. IEEE Communications Magazine, 60(12), 88–94. https://doi.org/10.1109/MCOM.003.2200110

[29] Cichosz, P., Kozdrowski, S., & Sujecki, S. (2021). Application of ML algorithms for prediction of the QOT in optical networks with imbalanced and incomplete data. 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). https://doi.org/10.23919/softcom52868.2021.9559095

[30] Sun, Z., Zhang, C., Zhang, M., Yang, F., & Wang, D. (2023). A stacking ensemble ML-based failure prediction model for optical networks with imbalanced data. 2023 Asia Communications and Photonics Conference/2023 International Photonics and Optoelectronics Meetings (ACP/POEM). https://doi.org/10.1109/acp/poem59049.2023.10369189

[31] Krasic, I., & Celar, S. (2022). Telecom fraud detection with machine learning on imbalanced dataset. 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). https://doi.org/10.23919/softcom55329.2022.9911518

[32] Rafique, D., & Velasco, L. (2018). Machine learning for network automation: Overview, architecture, and applications [Invited Tutorial]. Journal of Optical Communications and Networking, 10, D126–D143.

[33] Lin, Y.-D., Liu, Z.-Q., Hwang, R.-H., Nguyen, V.-L., Lin, P.-C., & Lai, Y.-C. (2022). Machine learning with variational autoencoder for imbalanced datasets in intrusion detection. IEEE Access, 10, 15247–15260. https://doi.org/10.1109/access.2022.3149295

[34] Wang, Y., & Zhang, Z. (2024). Industrial internet intrusion detection method based on VAE-WGAN-GP data enhancement. Academic Journal of Computing & Information Science, 7(4). https://doi.org/10.25236/ajcis.2024.070406

[35] Peng, Y., Wang, Y., & Shao, Y. (2022). A novel bearing imbalance fault-diagnosis method based on a Wasserstein conditional generative adversarial network. Measurement, 192, 110924. https://doi.org/10.1016/j.measurement.2022.110924

[36] Islam, Z., Abdel-Aty, M., Cai, Q., & Yuan, J. (2021). Crash data augmentation using variational autoencoder. Accident Analysis & Prevention, 151, 105950. https://doi.org/10.1016/j.aap.2020.105950

[37] Awan, S. E., Bennamoun, M., Sohel, F., Sanfilippo, F., & Dwivedi, G. (2021). Imputation of missing data with class imbalance using conditional generative adversarial networks. Neurocomputing, 453, 164–171. https://doi.org/10.1016/j.neucom.2021.04.010

[38] Abdelli, K., Azendorf, F., Tropschug, C., Griesser, H., Pachnicke, S., & Choo, J. (2022, October 17). Dataset for optical fiber faults. IEEE Dataport.

[39] Yang, W., Zhang, C., Wang, D., Zhu, H., Xu, X., Shi, D., & Zhang, M. (2024). Data labeling using unsupervised cascaded pre-training with fused multi-port data for optical failure management. Optical Fiber Communication Conference (OFC) 2024. https://doi.org/10.1364/ofc.2024.w2b.8

[40] Kruse, L. E., Kühl, S., Dochhan, A., & Pachnicke, S. (2024). Monitoring data augmentation of spectral information using VAE and GAN for soft-failure identification. 2024 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 1–3.

[41] Khan, L. Z., Freire, P. J., Pedro, J., Costa, N., Napoli, A., & Sambo, N. (2023). Data augmentation to reduce computational complexity of neural-network-based soft-failure cause identifier. Optical Fiber Communication Conference (OFC) 2023, Technical Digest Series (Optica Publishing Group), Paper M3G.3.