Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Statistics & Machine Learning
2024 | LIU-IDA/STAT-A--24/016--SE

Counterfactual Explanations for Neural Network based Anomaly Detection

Bridging Deep Learning and Actionable Insights: Counterfactual Methodologies for Enhanced Anomaly Interpretation in Industry 4.0

Singapura Ravi Varun

Supervisor : Jose M Peña
Examiner : Johan Alenlöv

External supervisor : Abhishek Srinivasan

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Abstract

In the evolving landscape of Industry 4.0, characterized by increased automation and digitalization, ensuring the operational efficiency and reliability of systems through predictive maintenance is critical. This thesis explores anomaly detection in time-series data using neural network-based methodologies, specifically Convolutional Autoencoders and Convolutional Variational Autoencoders, to identify deviations from normal operational patterns. However, such deep neural network models are termed as black-boxes due to inherent opaqueness in their decision making process, hence addressing the need for interpretability and transparency, particularly in safety-critical applications, this study introduces a framework for generating counterfactual explanations for anomalies detected by these networks. These explanations, representing minimal changes needed to alter an anomaly prediction, bridge the gap between complex model decisions and actionable insights for operators. Through a comprehensive literature review, this research identifies gaps in existing counterfactual explainable methods such as indiscriminately modifying all the features to generate counterfactual explanations and evaluates the implemented models, focusing on multivariate time series data for efficient anomaly detection. Additionally, this work proposes methods for generating counterfactual explanations through both optimization (gradient-based) and heuristic search approaches (genetic algorithm-based), enhancing understanding of the decision-making processes of the models and consequently performing a comparative analysis of the two methods quantitatively and qualitatively emphasizing the interpretability, utility and quality of the generated counterfactual explanations. The thesis concludes with a critical discussion on the methodology, particularly highlighting the limitations of the anomaly detection models and the feature selector component within the framework.

Acknowledgments

I would like to express my deepest gratitude to those who have made this thesis possible. First and foremost, my sincere thanks go to my parents for their unwavering support and encouragement throughout my academic journey. Their faith in my capabilities has been a constant source of strength and motivation.

I am immensely grateful to my supervisor, Professor Jose M. Peña, for his guidance and expertise. His insightful feedback and constructive criticism have significantly shaped this research. I also extend my appreciation to my external supervisor, Abhishek Srinivasan, whose expertise and insights have been instrumental in refining my research methodology and enhancing the depth of my work. My thanks also go to my examiner, Johan Alenlöv, for his thoughtful evaluations and suggestions that have helped me improve the quality of this thesis.

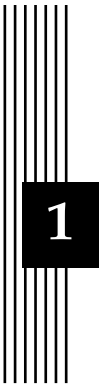
Furthermore, I would like to thank my friends Akshath and Ashwath for their unwavering support and inspiration. I am deeply indebted to them for their assistance and encouragement throughout this program. I also extend my gratitude to Kevin for hosting me in Linköping during mandatory meetings throughout the duration of this master's thesis. Finally, I would like to thank my fiancée, Shishira, for her unwavering support and understanding during challenging times, and for motivating me to be the best version of myself.

This journey would not have been possible without the support and encouragement from all of you, and for that, I am eternally grateful.

Contents

List of Figures

List of Tables



1 Introduction

1.1 Motivation

This section presents the motivation for the thesis by first providing a background to the topic and then outlining the problem statement. The aim is to highlight the importance of transparency and actionable insights in deep learning based anomaly detection models for predictive maintenance, and the necessity to address regulatory and practical challenges through counterfactual explanations.

1.1.1 Background

The era we live in has been hailed as the fourth industrial revolution, also called as Industry 4.0. This revolution is mainly characterized by an increase in automation through integration of digital technologies into manufacturing, industrial practices and automotive systems. In order to improve the operational efficiency of such systems and optimize maintenance resources, researchers have been continuously developing predictive maintenance. Predictive maintenance is defined as condition-based monitoring to optimize equipment performance and durability by continuously assessing its health in real time and is one of the three principal strategies that businesses and organizations utilize for optimized maintenance [1]. The other strategies encompass reactive maintenance, which deals with malfunctions as they manifest, and preventive maintenance, which adheres to a predetermined timetable for fault detection. However, reactive maintenance approaches lead to increased downtime in systems, preventive maintenance approaches are not very efficient and according to IBM [1], nearly half of such efforts are ineffective.

This brings us to a more modern approach, that is, machine learning based approaches which are data-driven intelligent algorithms aimed at pattern identification that provide actionable insights into system failure, thereby optimizing not only its downtime but also maintenance resources. One such ML based approach that has been extensively researched is anomaly detection, and in simple words, it refers to identifying patterns in data that deviate from normal behaviour. Anomaly detection has been an active field of research since the 1960's [2] for a plethora of applications such as fraud-detection, fault-detection and intrusion detection in safety critical systems, sensor systems and medical domains to name a few [3]. For a large part of the 20th century, researchers mainly relied on statistical techniques such as

proximity methods, but these techniques do not scale well to growing amounts of data due to the algorithmic complexity, and parametric methods suffer from the "curse of dimensionality" [4] as the data becomes more complex. Another level of complexity is added when we are forced to consider that most of the data in digitally integrated systems are complex time series in nature, that is, there are temporal dependencies in the data that would have to be modelled for effectively representing the underlying patterns in the data.

Fortunately, with the rapid advancement of deep learning, the shortcomings of the traditional statistical methods can be addressed efficiently since DL has shown promise in being capable of learning the underlying patterns of complex time series which can be a combination of multidimensional data with spatial and temporal characteristics [5].

1.1.2 Problem

In the context of this thesis, the focus is not with respect to the shortcomings of traditional statistical methods. However, it is about deep anomaly detection models that are used for diagnostics in predictive maintenance, where it is of utmost importance that there should be sufficient transparency in its decision making, since an operator or technician cannot be assumed to be well versed with the intricacies of time-series analysis or deep learning methodologies. To add to this, since most of the data emerging from sensor systems are multivariate in nature, they are challenging to interpret and stochastic deep learning only further complicates the problem [5]. Additionally, as per General Data Protection Regulation (GDPR), the "right to explanation" regulation mandates that automated decision making systems need to be transparent and explainable [6]. In the context of anomaly detection models, if a model predicts a particular instance to be anomalous then it should also provide details about how, and why it arrived at that conclusion and by doing so it can be proved whether the said model is fair or not. While explanations are important, they also need to be actionable, for example, in the context of SCANIA which is the commissioner of this thesis work, simply explaining the internal working of complex models is of little use to an operator or technician in charge of maintenance of a truck. However, if information is provided in a manner that is easily understandable regarding the reasons for a decision, they could alter behaviour for a desired outcome, and these alterations can be achieved by counterfactual explanations [6].

The term counterfactual explanations was first formally introduced by [6] and as per the author, a key motivating factor to generate and use counterfactual explanations, is a means to help someone act rather than simply understand how a model arrived at a particular decision. Counterfactuals present "what-if" scenarios that are intuitive to humans and counterfactual explanations of a prediction, in simple words are generated by minimally modifying feature values that changes the prediction to a predefined output [7]. In the context of this thesis, consider an original instance that was predicted to be non-anomalous, then it is the process of modification of feature/sensor values in order to change the model prediction to make it anomalous and vice-versa. By doing so, it would give an operator an intuitive understanding of which features/sensors contribute the most in the model's decision making process and what should change to avoid an unfavourable outcome. While feature importance can be determined by various techniques in the domain of explainable AI, the downside of these approaches is that they simply give insights into "why" and "how" a certain prediction was made and as mentioned earlier the advantage of using counterfactual explanations is that they provide "actionable" insights, meaning that they would explain "what" needs to be changed in order to achieve a desired outcome, and this is the premise and a key motivating factor for this thesis.

1.2 Aim

This thesis is aimed at generating counterfactual explanations for multivariate time series data originating from various sensors fitted on a truck in order to make deep anomaly de-

tection models more interpretable and obtain actionable insights into the decision making process of the said models. However, in order to generate these counterfactual explanations for a black-box model, first such a model has to be designed and implemented to efficiently flag anomalous behaviors in the data obtained from the sensors. In this project, Convolutional Autoencoder (C-AE) and Convolutional Variational Autoencoder (C-VAE) models are designed by taking inspirations from various works listed by Chandola [3].

First, these models are evaluated in terms of their performance in the anomaly detection task by using metrics such as F1-score, Recall, False Positive Rate to make sure models are indeed efficient in flagging anomalous behaviors, next, approaches into generating the counterfactual explanations will be explored and implemented. Finally, counterfactual explanations generated from different approaches will be quantitatively and qualitatively evaluated in a comparative fashion.

1.3 Research Questions

Now that the aims of this thesis have been mentioned in the previous section and in the process of realizing them the following research questions will be addressed and answered.

1. How does counterfactual explanations help interpret deep anomaly detection models?
2. Which method, optimization (gradient-based) or heuristic search (genetic algorithm based) generates better counterfactual explanations?
3. What quantitative and qualitative evaluation measures for the generated counterfactual explanations can be developed?

1.4 Delimitations

The delimitations for this thesis will be in the form of the type of data chosen, machine learning models used for anomaly detection and the type of explainability methods used.

1. The data is unlabelled multivariate time series data, hence only unsupervised machine learning models for anomaly detection will be designed and implemented. More specifically, only deep learning based models will be implemented for which the counterfactual explanations will be generated.
2. The application domain of anomaly detection will be limited to predictive maintenance and fault diagnosis since the data is collected from sensors from vehicles.
3. The type of counterfactual explanations generated will be also time series in nature.
4. Lastly, only counterfactual explainability will be explored and other techniques in the domain of Explainable AI will not be considered.

These delimitations are in place so as to make sure that the work done for the thesis remains focused on the area of interest and answers the research questions posed.

2 Data

2.1 Industrial Dataset

This section will provide an overview of the data collection process and introduce the datasets and the types of anomalies used for evaluating the anomaly detection and counterfactual explanation methods.

2.1.1 Data Collection

Since this thesis was conducted in collaboration with Scania CV AB, Södertälje, the data was collected by the Research and Development Team at the Connected Systems Department. The dataset was obtained through real-time operation of a vehicle, during which faults were deliberately introduced to generate anomalies. This vehicle was equipped with multiple sensors that captured a range of physical parameters. These parameters originate from sensors which are part of various systems on a truck, all of which are classified as continuous or discrete continuous features, alongside system status readings, which are classified as binary features. Readings were taken at an interval of 1 second, rendering this data as time series. Moreover, since the data is collected from different sensors, it is characterized as multivariate time series data. To simulate anomalous conditions, the vehicle underwent fault induction, with the sensor data from these irregular scenarios serving as the foundation for the anomaly dataset. The data collection infrastructure comprised both hardware and software components. This setup facilitated the live recording of sensor data as the vehicle operated under standard conditions and faced artificially induced faults. This methodology was pivotal in generating both normal and anomalous datasets.

2.1.2 Introduction

Of particular note in the evaluation process adopted in the results chapter is the distinctive approach for reporting results pertinent to the detection and explanation of anomalies. Given the occurrence of two principal anomaly types within the datasets, a strategic combination of datasets for each anomaly type has been executed. Hereafter, these combined sets will be referred to as **Anomaly Type 1** which is a combination of 3 datasets and consists of 4746 windows of sequences and **Anomaly Type 2** is a combination of 7 datasets and consists of 3138

windows of sequences, for clarity and convenience. Anomaly Type 1 encompasses anomalies primarily manifesting in *Sensor 5* which occurs due to a loss of correlation with *Sensor 6*, whereas Anomaly Type 2 is characteristic of anomalies predominantly arising in *Sensor 0* and *Sensor 3*. This bifurcation not only simplifies the narrative but also emphasizes the targeted analysis tailored to the specific anomaly patterns inherent in the respective datasets. Through this lens, the sections- in the results chapter will present the relevant results, casting light on the performance the anomaly detection models and explanatory power of counterfactual generation methods, within these defined categories. Further, as already mentioned the time series will be processed into windows of sequences and the details regarding that is presented in the methods chapter.

2.2 SKAB Dataset

The SKAB (Skoltech Anomaly Benchmark) dataset [8] comprises various simulated anomalies in a water supply system and provides a different domain to test the generalizability of the proposed methods. In presenting the results of this extended evaluation, the aim is to illustrate the replicability and robustness of the approaches across datasets with differing characteristics and anomaly profiles.

The SKAB is a comprehensive dataset designed for evaluating anomaly detection algorithms in time-series data. Developed by Skoltech, this dataset provides a robust foundation for benchmarking various anomaly detection methods, particularly those applicable in industrial and technical environments.

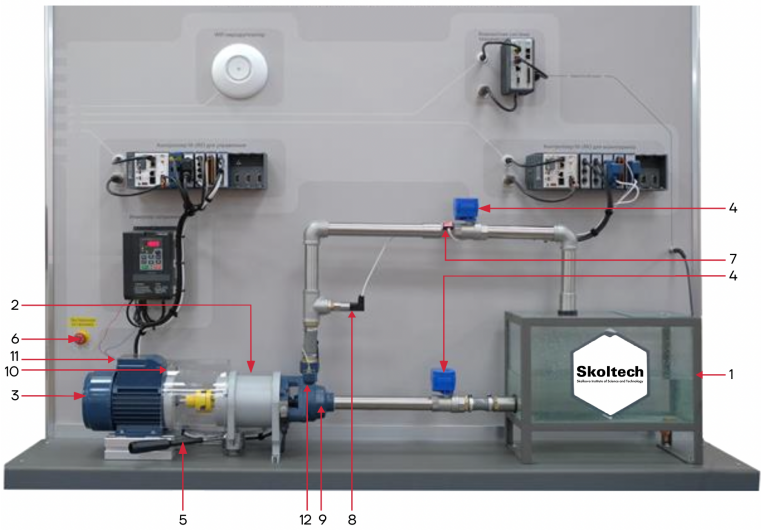


Figure 2.1: Front panel and composition of the water circulation, control and monitoring systems: 1,2 - solenoid valve (amount - 1); 3 - a tank with water (1); 4 - a water pump (1); 5 - emergency stop button (1); 6 - electric motor (1); 7 - inverter (1); 8 - compactRIO (1); 9 - a mechanical lever for shaft misalignment (1). Not shown parts - vibration sensor (2); pressure meter (1); flow meter (1); thermocouple (2). [8]

2.2.1 System Description

The SKAB dataset is derived from a controlled experimental setup involving a series of sensors attached to a testbed designed to mimic real-world industrial processes. This setup is constructed to replicate typical operational conditions as well as induced fault conditions. The primary goal is to generate time-series data that can be used to evaluate the performance

of anomaly detection algorithms under various scenarios. The testbed consists of several interconnected components that work together to create a dynamic and realistic environment for data collection.

The testbed's primary components include:

- **Solenoid Valve**: Controls the flow of fluid within the system.
- **Water Pump**: Circulates water through the system.
- **Electric Motor and Inverter**: Drive the water pump, with the inverter controlling the motor speed.
- **Sensors**: Various sensors including accelerometers, thermocouples, and pressure meters are used to monitor different physical parameters such as vibration, temperature, pressure, and flow rate.

This setup is instrumental in generating time-series data under both normal and faulty conditions, making it ideal for testing anomaly detection algorithms.

2.2.2 Nature of Anomalies

The anomalies in the SKAB dataset are categorized into two main types:

1. **Outliers**: These are single-point anomalies where a data point deviates significantly from the rest of the data.
2. **Changepoints**: These represent collective anomalies where a sequence of data points shows a significant deviation, indicating a shift in the system's behavior.

Each dataset within SKAB includes multiple features such as vibration acceleration (from accelerometers), current, pressure, temperature (from both engine body and fluid), voltage, and flow rate. The data points are labeled to indicate whether they are normal or anomalous, and whether they constitute a changepoint.

The SKAB dataset is essential for developing and benchmarking algorithms aimed at predictive maintenance and fault detection, providing both a rich and challenging environment for evaluating the robustness and accuracy of anomaly detection methods.

2.2.3 Data Structure

The dataset consists of 35 individual comma separated value (CSV) files, each representing a unique experiment. The columns in these files include:

- **datetime**: Timestamp of the recorded data.
- **Accelerometer1RMS** and **Accelerometer2RMS**: Vibration acceleration measurements.
- **Current**: Amperage on the electric motor.
- **Pressure**: Pressure in the loop after the water pump.
- **Temperature**: Temperature of the engine body.
- **Thermocouple**: Temperature of the fluid in the circulation loop.
- **Voltage**: Voltage on the electric motor.
- **RateRMS**: Circulation flow rate of the fluid inside the loop.
- **anomaly**: Binary indicator of whether the point is anomalous.

- **changepoint**: Binary indicator of whether the point is a changepoint for collective anomalies.

The detailed labeling and comprehensive nature of this dataset make it highly valuable for research and development in the field of anomaly detection, particularly for applications requiring high reliability and precision in predictive maintenance scenarios.

3 Theory

3.1 Theoretical Background

This section serves as a foundational framework for the reader, detailing essential concepts employed in the study of counterfactual explanations for neural network based anomaly detection. It encompasses a review of key concepts in anomaly detection, neural networks, explainable AI, particularly counterfactual explanations. By exploring the theoretical foundations, the aim is to provide a comprehensive understanding of the concepts necessary for design and development of deep anomaly detection models and consequently counterfactual explanations.

3.1.1 Anomaly Detection

The domain of anomaly detection has been extensively researched for the last century, as early as the 19th century by Edgeworth in 1887 [9], and as a result there have been many attempts to define what an *anomaly/outlier* is.

- Anomalies are patterns in data that do not conform to a well defined notion of normal behaviour [3].
- An outlier/anomaly is a data point that is significantly different from the remaining data [10].
- An outlier/anomaly is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism [11].
- An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data [4].

The process of identifying the aforementioned patterns or observations using statistical/data-driven approaches can be defined as anomaly detection [3]. In Figure 3.1, we can see the anomalies visualized, the figure is a simple time series data with points O_1, t_2 and the region marked in magenta all correspond to different types of anomalies explained below.

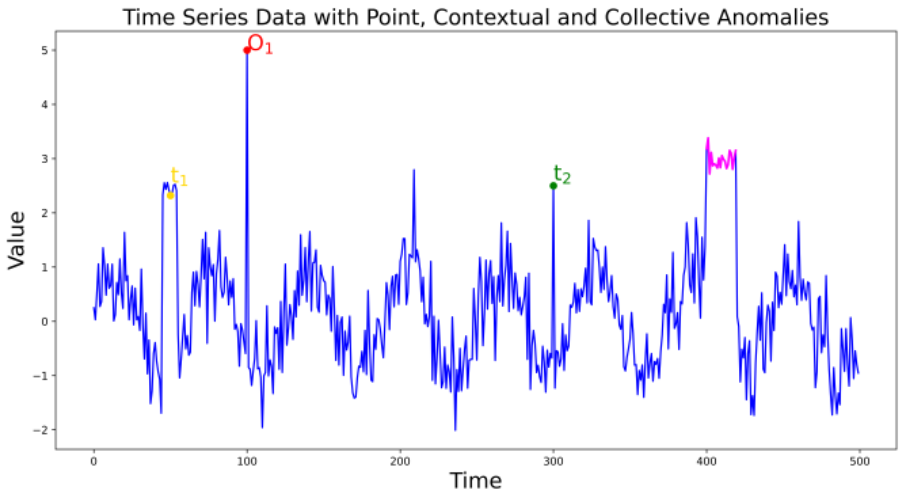


Figure 3.1: A simple 2D visualization of Point, Contextual and Collective anomalies.

Now that a formal definition of anomalies has been presented, it is also important to discuss what types of anomalies can occur because the nature of anomalies determine the applicability of anomaly detection techniques [3]. In real world applications, data points are usually related to one another either *temporally*, *spatially* or through network relationships [10]. In such cases, anomalies have been classified into three categories [3],[10],[5]:

- *Point Anomalies:* In the context of time series nature of the input data, assuming that it is a short sequence, if an unexpected event occurs at one point in time, then it is known as point anomaly [5], and has been visualized in Figure 3.1, where O_1 is a point anomaly.
- *Contextual Anomalies:* When an observation deviates from a given context, it is defined as contextual anomalies, for example, consider some time series data that has been pre-processed with a sliding window of size w, then $X_{t-w:t+w}$ is the context of a point x_t and is considered anomalous if the deviation crosses some threshold. From Figure 3.1, it can be seen that t_2 is anomalous given some context however, t_1 with a similar value is not anomalous due to its context.
- *Collective Anomalies:* When a group of data points, taken together, deviates significantly with respect to the entire data, it is considered a collective anomaly. In Figure 3.1, the region marked in magenta corresponds to the collective anomaly; however, the time points individually are not considered anomalous.

3.1.1.1 Deep Learning for Anomaly Detection

Anomaly detection utilizing deep learning is a dynamic and expansive area of research, with a focus on various paradigms. Hence, presenting a theoretical background for each paradigm would exceed the scope of this thesis. As a result, keeping in mind the objectives of the research questions already presented, this subsection will focus primarily on the fundamental concepts pertinent to AEs and VAEs, which are essential for the comprehension of the methodologies employed.

Before doing so, it is crucial to address two important categories of techniques/approaches with respect to the existing literature for time series anomaly detection, *reconstruction-based* and *forecasting-based* [5].

- *Reconstruction based* approaches in DL focus on reducing the dimensionality of input data by compressing it to a latent space and consequently reconstructing it back to its

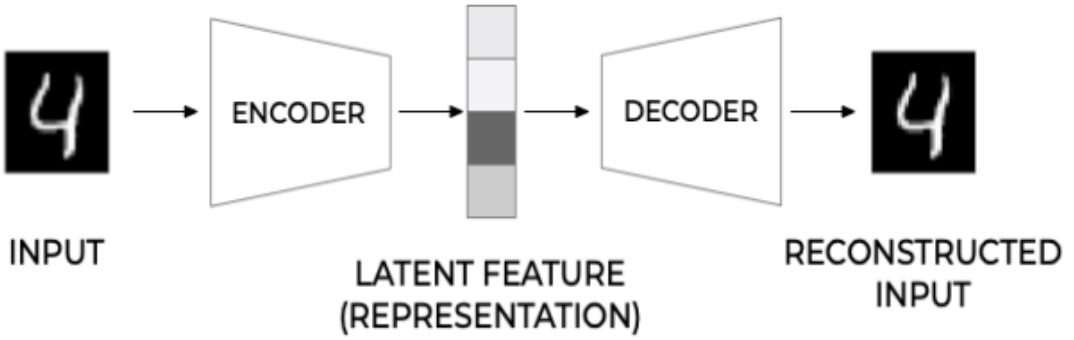


Figure 3.2: A general structure of an AE used for reconstructing a handwritten digit. (Image courtesy of [14])

original state, with some degree of accuracy and consequently a threshold can be determined. Any observation whose reconstruction loss exceeds this threshold is then determined as an anomaly.

- *Forecasting based* approaches on the other hand learn the inherent data generation mechanism and is trained to predict the next time stamp. A significant level of deviation in this forecasting leads to the time-stamp being classified as an anomaly.

Autoencoders

In the domain of anomaly detection, the data is generally unlabelled and hence unsupervised machine learning algorithms are used in such a setting. AEs play an important role and provide one of the fundamental paradigms [12] in this aspect, first introduced in the 1980s and again re-introduced with the advent of deep learning [13]. The basic idea behind the AE architecture is to reduce some high dimensional data into a low dimensional representation and consequently reconstructing the original data. This has been described as a non-linear generalization of PCA [13] that employs a neural network based "encoder-decoder" architecture to achieve the aforementioned task and in Figure 3.2 we can see the reconstruction of a handwritten digit using a simple AE.

To formalize the above definition of an AE, consider a time series dataset, $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$, where each $x^{(i)} \in \mathbb{R}^d$ is a time-series sequence. For each sequence $x^{(i)}$, a sliding window of size w is applied to generate overlapping sub-sequences, $x^{(i)}_{t:t+w}$, where t denotes the current time step and $(t + w)$ indicates end of the window. The AE consists of two components: an encoder and a decoder.

- *Encoder:* The encoder maps a sub-sequence $x^{(i)}_{t:t+w}$ to a hidden representation $z^{(i)}_{t:t+w}$, where $z^{(i)}_{t:t+w} \in \mathbb{R}^p$ and $p < d$, through a deterministic mapping f, parameterized by $\theta = \{W, b\}$,
$$z^{(i)}_{t:t+w} = f_\theta(x^{(i)}_{t:t+w}) = \sigma(W x^{(i)}_{t:t+w} + b), \tag{3.1}$$
where σ is an element-wise activation function, W is a weight matrix, and b is a bias vector.

- *Decoder:* The decoder maps the hidden representation $z^{(i)}_{t:t+w}$ back to a "reconstructed" input $\hat{x}^{(i)}_{t:t+w}$, where $\hat{x}^{(i)}_{t:t+w} \in \mathbb{R}^d$, through a deterministic mapping g, parameterized by $\theta' = \{W', b'\}$,
$$\hat{x}^{(i)}_{t:t+w} = g_{\theta'}(z^{(i)}_{t:t+w}) = \sigma'(W' z^{(i)}_{t:t+w} + b'), \tag{3.2}$$
here σ' may be the same as σ, or a different activation function.

The parameters $\Theta = \{\theta, \theta'\}$ are learned by minimizing a loss function L that measures the difference between the input $x^{(i)}_{t:t+w}$ and the reconstructed input $\hat{x}^{(i)}_{t:t+w}$. The loss function that is minimized during the training phase of the AE is the *Huber Loss*,

$$L_\delta(x^{(i)}_{t:t+w}, \hat{x}^{(i)}_{t:t+w}) = \begin{cases} \frac{1}{2}(x^{(i)}_{t:t+w} - \hat{x}^{(i)}_{t:t+w})^2 & \text{for } |x^{(i)}_{t:t+w} - \hat{x}^{(i)}_{t:t+w}| \leqslant \delta, \\ \delta|x^{(i)}_{t:t+w} - \hat{x}^{(i)}_{t:t+w}| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases} \tag{3.3}$$

The training process involves using an optimization technique such as stochastic gradient descent to find the parameters Θ that minimize the loss function L. Upon training, the encoder provides a compressed representation of the input data in a lower-dimensional space, which can be useful for various applications such as data compression, denoising, or feature extraction. The decoder can reconstruct the data from this compressed form, allowing us to evaluate the quality of the learned representations.

Variational Autoencoders

VAEs are generative models that simulate the data generation process and provide a principled framework for learning deep latent-variable models and corresponding inference models [15]. They are probabilistic graphical models combining deep learning and variational inference [16], by imposing a probabilistic structure in order to regularize the latent space representation. The major difference between AEs and VAEs is in the way they model the latent space, the former learns a deterministic mapping, while the VAEs learn a probabilistic mapping of the latent space. VAEs consist of two coupled, but independently parameterized models, the *encoder* and *decoder*. The *encoder* learns to approximate its posterior over the latent representation of the input data thereby compressing it into a lower-dimensional space. Meanwhile, the *decoder* learns to reconstruct the original data by mapping the latent space back to the original data space [15], the same is visualized as a schematic diagram of a variational autoencoder in Figure 3.3.

Similarly as above, given a time-series dataset $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$, where each $x^{(i)}$ represents a sequence of time-series data. For each sequence $x^{(i)}$, we apply a sliding window of size w to generate overlapping subsequences, $x^{(i)}_{t:t+w}$, where t is the current time step and $t : t + w$ indicates the start of the window. The VAE introduces a probabilistic latent variable $z^{(i)}_{t:t+w}$ and consists of two primary components: an encoder (inference model) and a decoder (generative model).

- *Encoder:* The *encoder* in a VAE approximates the posterior distribution $p(z^{(i)}_{t:t+w}|x^{(i)}_{t:t+w})$ with a variational distribution $q_\phi(z^{(i)}_{t:t+w}|x^{(i)}_{t:t+w})$ which is typically assumed to be a multivariate Gaussian with a diagonal covariance structure,

  $$q_\phi(z^{(i)}_{t:t+w}|x^{(i)}_{t:t+w}) = \mathcal{N}(z^{(i)}_{t:t+w}; \mu_\phi(x^{(i)}_{t:t+w}), \text{diag}(\sigma^2_\phi(x^{(i)}_{t:t+w}))), \tag{3.4}$$

  where ϕ denotes the parameters of the *encoder*, $\mu_\phi(x^{(i)}_{t:t+w})$ is the mean vector and $\sigma^2_\phi(x^{(i)}_{t:t+w})$ is the variance vector of the approximate posterior.

- *Decoder:* The *decoder* defines a likelihood $p_\theta(x^{(i)}_{t:t+w}|z^{(i)}_{t:t+w})$ which is the probability of the data given the latent representation. This is typically modelled as,

  $$p_\theta(x^{(i)}_{t:t+w} \mid z^{(i)}_{t:t+w}) = \mathcal{N}(x^{(i)}_{t:t+w}; f_\theta(z^{(i)}_{t:t+w}), \Sigma), \tag{3.5}$$

  where $f_\theta(z^{(i)}_{t:t+w})$ is a nonlinear transformation from the latent space to the data space, and Σ is often assumed to be a diagonal covariance matrix.

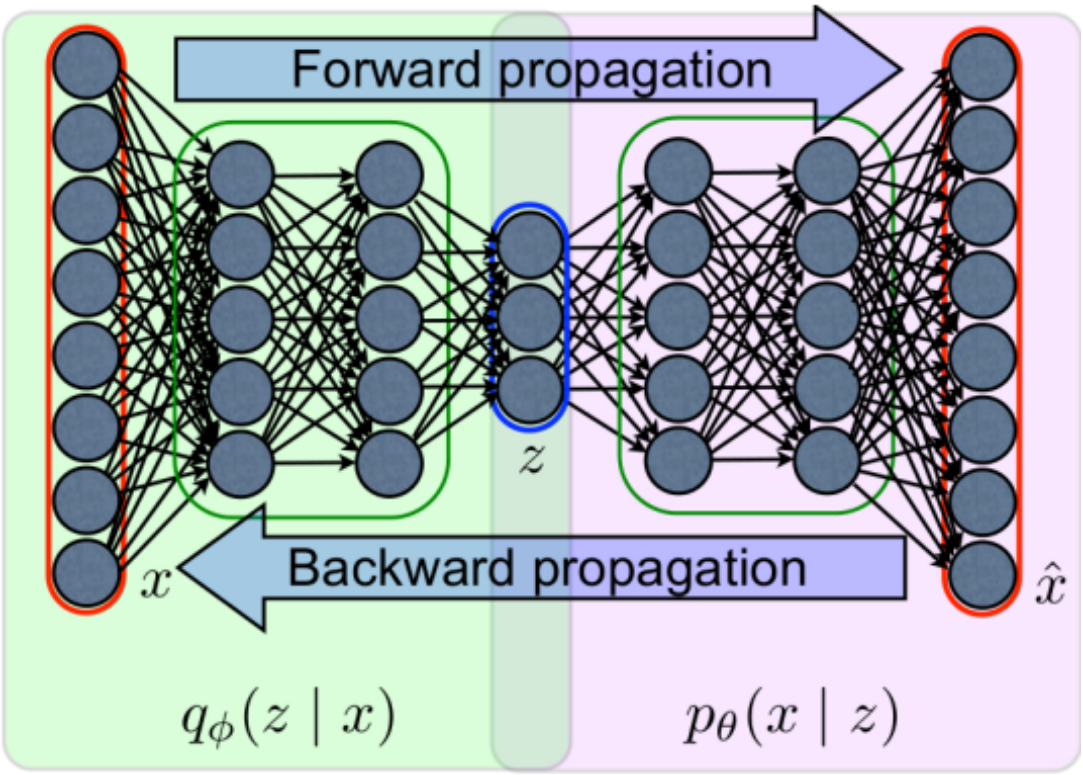


Figure 3.3: A schematic representation of a VAE with the inference model $q_\phi(z \mid x)$ for encoding input x into latent representation z, and the generative model $p_\theta(x \mid z)$ for reconstructing x from z, with forward and backward propagation phases indicated. (Image courtesy of [16])

The learning objective of a VAE is to maximize the evidence lower bound (ELBO) which is given by,

$$L(\phi, \theta; x^{(i)}_{t:t+w}) = \mathbb{E}_{q_\phi(z^{(i)}_{t:t+w}|x^{(i)}_{t:t+w})}[\log p_\theta(x^{(i)}_{t:t+w}|z^{(i)}_{t:t+w})] - \mathrm{KL}[q_\phi(z^{(i)}_{t:t+w}|x^{(i)}_{t:t+w}) \| p(z^{(i)}_{t:t+w})], \tag{3.6}$$

$$\mathrm{KL} = \mathbb{E}_{q_\phi(z_{t:t+w}|x^{(i)}_{t:t+w})}\left[-0.5 \cdot \left(1 + \log(\sigma^2_\phi(x^{(i)}_{t:t+w})) - \mu^2_\phi(x^{(i)}_{t:t+w}) - \exp(\log(\sigma^2_\phi(x^{(i)}_{t:t+w})))\right)\right], \tag{3.7}$$

where KL denotes the Kullback-Leibler divergence between the approximate and true posteriors. The first term in Equation 3.6 represents the reconstruction likelihood, while the second term acts as a regularizer, encouraging the variational distribution to be close to the prior $p(z^{(i)}_{t:t+w})$. The gradients of the ELBO with respect to the parameters can be computed using the *reparameterization trick* for the Gaussian distributions.

The *reparameterization trick* (Figure 3.4) is a technique used to enable gradient descent optimization in VAEs by reparameterizing the random variable z in a way that allows the gradient to pass through the stochastic node. Specifically, instead of sampling z directly from $q_\phi(z|x)$, z can be expressed as a deterministic variable,

$$z = \mu_\phi(x^{(i)}_{t:t+w}) + \sigma_\phi(x^{(i)}_{t:t+w}) \odot \epsilon, \tag{3.8}$$

where ϵ is an auxiliary variable sampled from a standard normal distribution $\epsilon \sim \mathcal{N}(0, I)$ and $\odot$ denotes the element-wise product. This reparametrization allows for the gradient of the ELBO with respect to the parameters to be calculated directly, facilitating the use of a gradient based optimization technique of the parameters ϕ and θ.

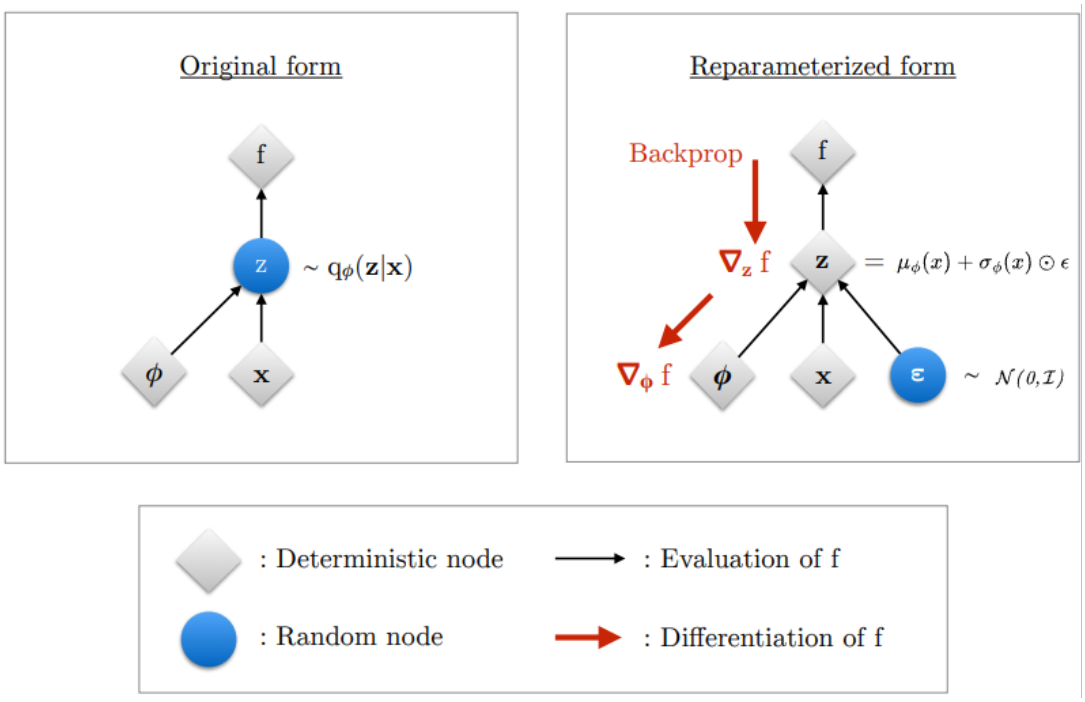


Figure 3.4: An illustration of the reparametrization trick and how it enables a direct back-propagation through the random variable **z**. (Image courtesy of [15])

3.1.2 Local Outlier Factor Based Feature Selector

This section discusses use of the Local Outlier Factor (LOF) based feature selector, a key step in creating sparse and clear counterfactual explanations. After identifying anomalies with the trained models, the process now focuses on pinpointing specific anomalous features. This is different from examining the entire data instance and helps to make the counterfactual explanations simpler and more direct. This approach leads to explanations that are easier to understand compared to the methods talked about in the *Related Works* section, that is most of the generation methods modify all the features to generate counterfactual explanations which lead to overly complex explanations in contrast to the method employed in this work which only modifies anomalous features thereby promoting sparse explanations.

3.1.2.1 Formal Definition of Local Outlier Factor (LOF)

The LOF is an unsupervised anomaly detection algorithm that quantifies the degree of outlierness of a data point. It measures how isolated a data point is with respect to the surrounding neighborhood, considering the local density of the data.

Steps to Calculate LOF:

1. **k-distance and k-distance Neighborhood**: For a data point p, the k-distance (k-distance(p)) is defined as the distance to the k-th nearest neighbor. The set of k-distance neighbors, $N_k(p)$, includes all points within this distance.
2. **Reachability Distance**: The reachability distance of a point p from another point o is defined as:

$$\text{reachability-distance}_k(p,o) = \max\{\text{k-distance}(o), \text{distance}(p,o)\}$$

This ensures that the reachability distance is at least the k-distance of o.

3. **Local Reachability Density (LRD)**: The local reachability density of a point p is the inverse of the average reachability distance of p from its k-nearest neighbors:

$$\text{LRD}_k(p) = \left(\frac{\sum_{o \in N_k(p)} \text{reachability-distance}_k(p, o)}{|N_k(p)|} \right)^{-1}$$

4. **Local Outlier Factor (LOF)**: The LOF of a point p is the average ratio of the local reachability density of p and those of p's k-nearest neighbors:

$$\text{LOF}_k(p) = \frac{\sum_{o \in N_k(p)} \frac{\text{LRD}_k(o)}{\text{LRD}_k(p)}}{|N_k(p)|}$$

A data point p with an LOF value significantly greater than 1 is considered an outlier. This value indicates how much p deviates from its neighbors in terms of local density, with higher values corresponding to stronger indications of anomaly.

3.1.3 Counterfactual Explanations

Interpretability or Explainability in general terms, is to understand a particular process or an event, and in the context of machine learning models, it refers to a set of techniques developed to extract relevant information and provide insights about domain relationships contained in the data [17]. A machine learning model's level of interpretability is directly proportional to a chosen audiences' comprehension with respect to decision making process [7]. The Figure 3.5, shows a general overview of different learning techniques such as neural networks, ensemble methods, and statistical models that are widely used in the machine learning community as a function of their predictive power versus the inherent explainability they offer in the process of decision making. Before, presenting the theory for counterfactual explanations, it is important to present a general overview of the taxonomies and paradigms in the domain of Explainable AI.

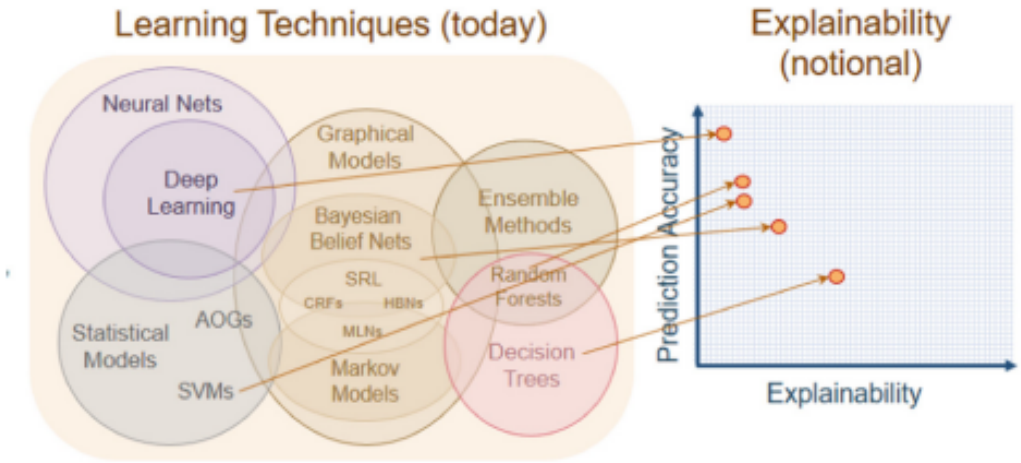


Figure 3.5: ML models as a function of accuracy vs. explainability. (Image courtesy of [18])

The authors [19],[20] classify types of interpretability into two categories, namely:

- *Global Interpretability:* Where the model's decision making across all instances can be described, and are expressed as an expected values based on the distribution of the data [7].

- *Local Interpretability:* These, in contrast focuses only on interpreting decision making for a particular instance.

In this thesis, the focus will be mainly on *local interpretability* since that is the most commonly used method for explaining deep neural networks [20].

Furthermore, interpretability methods can be classified as being:

- *Model specific*, meaning that these methods are limited to specific model classes, algorithms [20] or for some model's internals [21].
- *Model agnostic*, meaning that these methods are those that can be applied to any ML algorithm as they separate prediction from explanations [20] and they are generally applied after training is complete and are *post-hoc* [21].

The authors [20],[21] have further classified model agnostic methods based on the techniques as:

- *Visualization* wherein various techniques like Partial Dependence Plots (PDP), which can be intuitively thought of as expected target response as a function of the input features of interest, whereas Individual Conditional Expectation (ICE) visualizes the dependence of prediction on a feature for each sample separately. Surrogate Models such as LIME [22] which gives the significance of each input characteristic for the prediction and produces explanations in the form of feature weights. These are some of the techniques that are used to explore the pattern hidden inside a neural unit or any black-box model.
- *Knowledge Extraction* is the process of extracting explanations in an interpretable form from a DNN during training [20] by using techniques such as Rule Extraction and Model Distillation, which is a technique that transfers explanations' information from a teacher model to a student model [23].
- *Influence Methods* is the process of estimating feature importance by modifying internal components and observing the change in model performance by Sensitivity Analysis, Layer Wise Relevance Propagation [24] a novel method to visualize contribution of single pixels to predictions, SHAP [25] which uses a game theoretic approach to examine feature importance
- *Example-based explanation* which select particular instances of the dataset to provide interpretability [20] by using techniques such as *Counterfactual Explanations*.

More specifically speaking, it is not sufficient to just extract relevant information, but rather information upon which a certain action can be taken, and this is why counterfactual explanations are important. The reason for this is, counterfactual explanations at its core is able to describe causal situations that are rather intuitive to humans [7]. One of the most widely agreed upon definition for counterfactual explanations is, **counterfactual explanations of a prediction describes the smallest change to feature values that changes the prediction to a predefined output** [6],[7].

The Figure 3.6 depicts a machine learning model's decision boundary, where an individual with an income of $45,000, debt of $11,000, age of 29, and savings of $6,000 is placed in the reject region, while three others with higher incomes and/or older age are in the accept region, illustrating how different features can affect the outcome of a model's decision.

In order to further simplify what counterfactual explanations are and how they would aid in the context of this thesis, consider a scenario where a system of sensors fitted on a truck start emitting anomalous signals and consequently a complex black-box anomaly detector predicts an anomaly. However, this is not very insightful to a technician because of the opaqueness in the decision making process. Furthermore, it also gives the operator no input

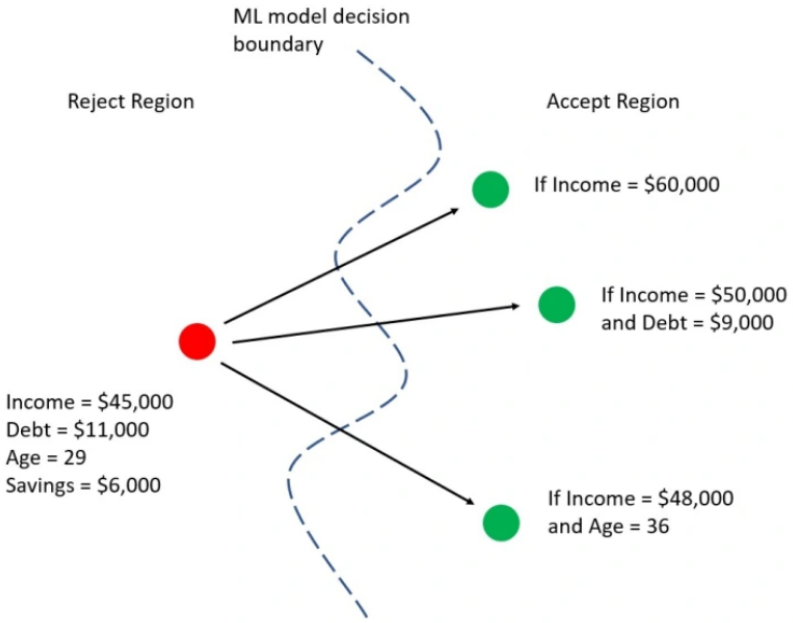


Figure 3.6: An illustration of counterfactual explanations in the case of loan rejection, and the actionable insights it provides in order to change the rejection into an accepted application. (Image courtesy of [26])

on what needs to be done in order to avoid such an anomaly. In contrast, a counterfactual explanation in such a scenario would modify the original anomalous instance's feature values to change the prediction to being "normal". By doing so, the technician now intuitively knows not only which feature/sensor lead to an anomaly, but also what to be done in the future in order to avoid such a situation.

To mathematically formulate the above definition of what counterfactual explanations are, consider $x \in \mathbb{R}^n$ which denotes the original feature vector, and $x' \in \mathbb{R}^n$ represents the counterfactual feature vector, where n is the number of features. Let $f : \mathbb{R}^n \to \mathbb{Y}$ be a prediction function that maps an input feature vector to a prediction space $\mathbb{Y}$. The counterfactual explanations aims to find x' such that,

$$f(x') = y', \quad y' \neq f(x), \tag{3.9}$$

subject to the constraint,

$$\|x' - x\|_p = \min_{x'} \|x' - x\|_p, \quad \text{where } f(x') = y', \tag{3.10}$$

for some norm $\|\cdot\|_p$ and predefined output y'. The goal is to make the smallest change to x and achieve the new predefined output y', diverging from the original prediction $f(x)$. There are various methods, that have been presented in the related works section, such as the optimization based approaches and heuristic search strategies, to apply the smallest possible change to an instance of interest and consequently generate a *counterfactual explanation*. The subsequent sections will focus in depth about how these methods are used to generate counterfactual explanations.

3.1.3.1 Gradient Based Approach

This approach falls under an umbrella of *backpropagation* based interpretability methods, and it fundamentally involves computation of gradients of a differentiable black box model [27]. Furthermore, it has been shown that the model gradient vector with respect to the input can be important when trying to interpret the effect of input features in complex models [28].

Consider a differentiable black box model, in this case, the AE and VAE, $f : X \to \hat{X}$ where X is the input space and $\hat{X}$ is the reconstructed output space, let us consider a time series dataset, $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ which has been transformed into overlapping sequences windows, similarly as in the case of AE and VAE. Then, one can compute the partial derivative,

$$\nabla f(x_{t:t+w}) = \left(\frac{\partial f}{\partial x^{(1)}_{t:t+w}}, \frac{\partial f}{\partial x^{(2)}_{t:t+w}}, \ldots, \frac{\partial f}{\partial x^{(N)}_{t:t+w}} \right), \tag{3.11}$$

where the resulting vector can be studied to explain the influence of the corresponding variable on the entire model.

The process for generating counterfactual explanations via the *Gradient based* method can be formalized mathematically, which was first formalized by Wachter [6],

$$\arg\min_{x'_{t:t+w}} L(f_\theta(x'^{(i)}_{t:t+w}), x^{(i)}_{t:t+w}), \tag{3.12}$$

where $x'^{(i)}_{t:t+w}$ is the *counterfactual* to the original instance $x^{(i)}_{t:t+w}$, f_θ is some model with parameters θ that are fixed and L is some objective function that is optimized.

In this case, the optimization involves iteratively adjusting an input $x_{t:t+w}$ to minimize the reconstruction loss L between the model's output $f_\theta(x'^{(i)}_{t:t+w})$ and $x_{t:t+w}$. A counterfactual $x'^{(i)}_{t:t+w}$ is obtained when the loss $L(f_\theta(x'^{(i)}_{t:t+w}), x^{(i)}_{t:t+w})$ is minimized for a set number of iterations. The loss L is a combination of Mean Squared Error (MSE) and Mean Absolute Error (MAE), and is defined as,

$$L(f_\theta(x'^{(i)}_{t:t+w}), x^{(i)}_{t:t+w}) = \text{MSE}(f_\theta(x'^{(i)}_{t:t+w}), x^{(i)}_{t:t+w}) + \text{MAE}(f_\theta(x'^{(i)}_{t:t+w}), x^{(i)}_{t:t+w}), \tag{3.13}$$

where both MSE and MAE are computed element-wise between the output and the target.

During optimization with the Adam algorithm, the input $x^{(i)}_{t:t+w}$ is updated according to the gradient of L with respect to $x^{(i)}_{t:t+w}$, denoted $\nabla_{x_{t:t+w}} L$. The update rule, accounting for any excluded signals, is expressed as,

$$x'^{(t+1)}_{t:t+w} = x^{(t)}_{t:t+w} - \eta \cdot (\nabla_{x_{t:t+w}} L(f_\theta(x'^{(i)}_{t:t+w}), x^{(i)}_{t:t+w})), \tag{3.14}$$

where η is the learning rate, t is the iteration count. Finally, to make sure that the generated counterfactuals are sparse, only those features are adjusted which are determined to be anomalous and the rest are excluded from the optimization process.

During the generation of counterfactual explanations, an optimization process adjusts features of $x^{(i)}_{t:t+w}$ only when a particular feature is deemed anomalous by the LOF-based feature selector, as indicated by the indicator function $\mathbb{I}(j)$ defined in Equation 4.9. This condition enforces sparsity by ensuring that modifications are made solely to features that the LOF has identified as contributing to the anomaly, thus maintaining simplicity and interpretability in the resulting counterfactual explanations.

3.1.3.2 Genetic Algorithm

In this work, a genetic algorithm is employed from the *pymoo* framework [29], an optimization heuristic inspired by the process of natural selection, to generate counterfactual explanations

for deep learning models. The aim is to identify perturbations to the input data that lead to significant output changes while remaining within realistic bounds, providing interpretable counterfactuals.

The genetic algorithm (Figure 3.7) is adapted to evolve a population of individuals, each representing a possible counterfactual, through a series of genetic operations. It belongs to a larger class of *evolutionary algorithms* that are used to solve constrained and unconstrained optimization problems inspired by natural selection, the process that drives biological evolution. The algorithm continually adjusts a population of individual solutions and in every iteration, it chooses individuals from the existing population as parents to generate offspring for the subsequent generation. Through consecutive generations, the population progresses towards an optimal solution [30].

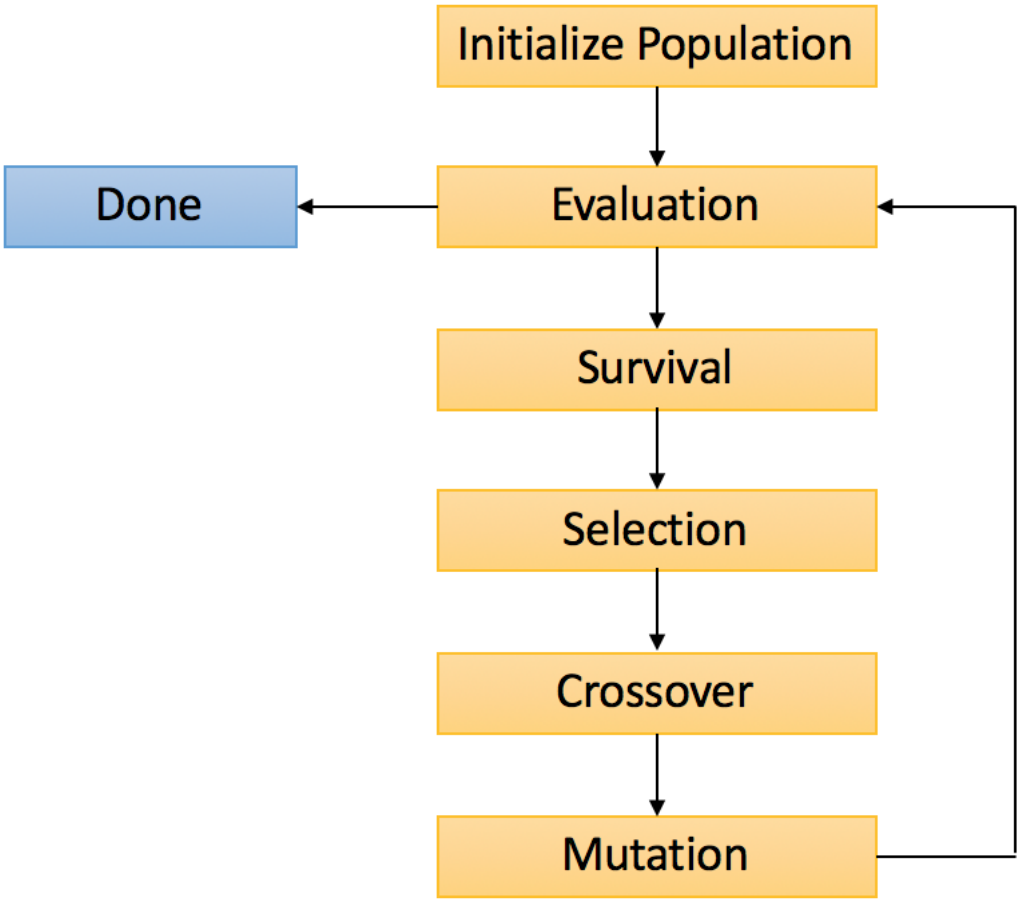


Figure 3.7: Flowchart depicting the cycle of a genetic algorithm: initialization, evaluation, selection, crossover, mutation, and termination.

The process is as follows:

1. **Initialization**: The population is initialized with perturbations around the original input instances to explore the input space near actual data points. The perturbation applied is *gaussian noise* with zero mean and unit variance.

2. **Fitness Evaluation**: The fitness of each individual in the population is determined by a multi-objective function,

$$f(x) = MSE(\hat{x}'^{(i)}_{t:t+w}, x'^{(i)}_{t:t+w}) + MAE(\hat{x}'^{(i)}_{t:t+w}, x'^{(i)}_{t:t+w}) + D(x^{(i)}_{t:t+w}, x'^{(i)}_{t:t+w}), \tag{3.15}$$

where $\hat{x}'^{(i)}_{t:t+w}$ is the reconstruction of the counterfactual instance $x'^{(i)}_{t:t+w}$ and the distance metric D is a weighted sum of three norms: l_0, l_1, and l_∞. These norms measure the count of changed features, the average change in feature values, and the maximum change across all features, respectively. The weights α, β, and γ are applied to the respective norms as follows,

$$D(x^{(i)}_{t:t+w}, x'^{(i)}_{t:t+w}) = \alpha \cdot \frac{l_0}{F} + \beta \cdot \frac{l_1}{F} + \gamma \cdot l_\infty, \tag{3.16}$$

with F being the number of features excluding the batch dimension. Here, l_0 norm is computed as the count of features for which the perturbation exceeds a small threshold, the l_1 norm as the average absolute difference between the perturbed and original batch, and l_∞ norm as the maximum absolute difference. The goal is to minimize this fitness function, yielding counterfactuals that are close to the original input yet sufficiently altered to cross the decision boundary of the model.

3. **Selection**:Individuals are chosen for reproduction based on their fitness, favoring those with lower values that indicate more accurate and sparse *counterfactuals*.

4. **Crossover and Mutation**: To enhance genetic diversity within the population, our approach employs two key operations: crossover and mutation.

   Crossover: This genetic operator is implemented using the *differential evolution* algorithm's crossover strategy to generate new offspring solutions. In this process, offspring are created by adding the weighted difference between two randomly selected population vectors (donors) to a third vector (target). The weighting factor, commonly referred to as the differential weight, is a crucial parameter in controlling the amplification of the differential change. This method effectively combines attributes from multiple parents to explore new regions of the solution space.

   Mutation: This genetic operator is carried out through *Polynomial Mutation* (PM) introduces random variations to the offspring's traits. In this method, a polynomial probability distribution is used to perturb a solution in the parent's vicinity thereby maintaining diversity in the population and in combination with the crossover operator it helps making the overall search efficient [31].

5. **Survivor Selection**: The algorithm updates the population by selecting the fittest individuals from the current generation, often using elitist strategies to ensure high-quality solutions persist.

6. **Termination**: The evolution continues for a set number of generations or until a satisfactory counterfactual is found, as indicated by the fitness falling below a certain threshold.

3.1.3.3 Evaluation Metrics

Now that the two different methods for generating counterfactual explanations have been presented, this subsection will deal with the quantitative and qualitative evaluation assessment to evaluate their effectiveness and relevance. The quantitative evaluation metrics not only asses the quality of the explanations but also ensure that properties related to them are meticulously evaluated. In parallel, the qualitative evaluation aims to contextualize and validate the counterfactual explanations within real-world scenarios, enhancing our understanding of their practical implications. While quantitative evaluation metrics ensure the

effectiveness and relevance of counterfactual explanations by providing objective measures of sparsity, validity, and distance, qualitative evaluation, on the other hand, enriches this understanding by assessing the intuitiveness, plausibility, and real-world applicability of these explanations.

Quantitative Evaluation

- **Validity**: This metric quantifies the proportion of counterfactual explanations that successfully alter the model's prediction to a different outcome than the original prediction. A higher validity score indicates that a larger fraction of counterfactuals are effective in changing the model's decision, thus demonstrating their potential in providing alternative outcomes.

$$\text{validity} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(L_w < \tau), \tag{3.17}$$

  where L_w is the reconstruction loss for the windows (Equation 4.2), τ is the threshold of detecting anomalies (Equation 4.3), and $\mathbb{I}$ is an indicator function, reflecting the accuracy of the counterfactual in mirroring an alternative yet plausible reality.

- **Distance**: This metric assesses the average distance between the original instances and their respective counterfactuals across all features and sequences. It measures how significantly the counterfactuals deviate from the original instances, aiming for minimal yet effective modifications.

$$d(x^{(i)}_{t:t+w}, x'^{(i)}_{t:t+w}) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{j=1}^{T}\sqrt{\sum_{k=1}^{F}(x^{(i)}_{t:t+w} - x'^{(i)}_{t:t+w})^2}\right), \tag{3.18}$$

  where $x^{(i)}_{t:t+w}$ and $x'^{(i)}_{t:t+w}$ are the original and counterfactual instances.

- **Sparsity**: This metric calculates the overall average sparsity across all sequences, providing insight into the number of feature modifications required to generate each counterfactual. Ideally, a sparse counterfactual, which changes fewer features, is preferred for its simplicity and ease of interpretation.

$$\text{sparsity}(x^{(i)}_{t:t+w}, x'^{(i)}_{t:t+w}) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T\cdot F}\sum_{j=1}^{T}\sum_{k=1}^{F}\mathbb{1}(|x^{(i)}_{t:t+w} - x'^{(i)}_{t:t+w}| > \theta)\right), \tag{3.19}$$

  where N is the total number of sequences, T is the sequence length, F is the number of features per time step and θ is a predefined threshold for significant change.

where,

- N: the total number of sequences.
- T: the sequence length, i.e., the number of time steps per sequence.
- F: the number of features per time step.
- θ: the predefined threshold for significant change.
- $\mathbb{1}(\cdot)$: the indicator function, returning 1 when its argument condition $|x^{(i)}_{t:t+w} - x'^{(i)}_{t:t+w}| > \theta$ is true, indicating that the absolute difference between the anomaly and counterfactual at a given point exceeds the threshold, and 0 otherwise.

Qualitative Evaluation

UMAP

The UMAP (Uniform Manifold Approximation and Projection) algorithm [32] serves as a powerful tool for the qualitative evaluation of counterfactual explanations methods, specifically the gradient and genetic algorithm approaches. By leveraging its manifold learning capabilities, UMAP assists in visually distinguishing the distribution patterns of original, anomalous, and counterfactual data instances within a two-dimensional embedding space. This visualization process is crucial for assessing the effectiveness of the counterfactual generation methods.

Through UMAP, the aim is to observe the proximity of counterfactual instances to the normal data distribution, thereby evaluating the counterfactuals' alignment with expected, non-anomalous states. A successful counterfactual explanations method should ideally result in counterfactuals that cluster closely with normal instances, distinct from the anomalous ones. Such a pattern would suggest that the generated counterfactual explanations are not only plausible but also relevant, offering actionable insights into the conditions leading to anomalies.

This qualitative analysis complements quantitative metrics, providing a deeper insight into the practical utility of counterfactual explanations. By illustrating how different counterfactual generation methods influence the positioning of counterfactuals relative to normal and anomalous instances, the UMAP visualization underscores the interpretability and actionable nature of the generated explanations. Such evaluations are invaluable for refining counterfactual generation techniques, ensuring they produce meaningful and understandable explanations for complex anomaly detection scenarios.

Naive Anomaly Detector based on Correlation Loss

Leveraging domain knowledge, this approach predicates on the understanding that a certain sensor pairs, say *Sensor5* (Anomaly Type 1) and *Sensor6*, exhibit a strong correlation under normal operational conditions, which diminishes in the presence of anomalies. This phenomenon underpins the development of a qualitative metric to evaluate the effectiveness of counterfactual explanations generated for anomalous conditions.

To illustrate the concept further, let's consider a practical example involving two sensors, namely *Sensor5* and *Sensor6*, which for example may represent pressure and temperature readings, respectively. Under normal operating conditions, it's expected that the readings from these sensors would exhibit a high degree of correlation. This is because, in many mechanical systems, temperature and pressure are interdependent: as the temperature increases, so does the pressure, and vice versa, adhering to the ideal gas law under certain conditions.

However, in the event of an anomaly affecting *Sensor5*'s (pressure sensor) readings, this correlation is disrupted. The anomalous conditions could result from various factors, such as a leak or a blockage in the system, leading to abnormal pressure levels that do not align with the corresponding temperature readings from *Sensor6*.

In such scenarios, the counterfactual explanations generated for *Sensor5* would aim to simulate a modification in the pressure readings that could revert the system back to its normal operational state. Specifically, these counterfactuals would adjust *Sensor5*'s readings to values that restore the high correlation with *Sensor6*'s (temperature sensor) normal readings. This adjustment is guided by the domain knowledge that under normal conditions, pressure and temperature should correlate closely.

The method employs *Pearson correlation coefficient* to quantify the correlation between *Sensor5* and *Sensor6* across different states: normal, anomalous, and counterfactual. The assump-

tion is that under normal conditions, *Sensor5* and *Sensor6* should be highly correlated, reflecting their operational interdependence. In contrast, during anomalous conditions in *Sensor5*, this correlation is expected to approach zero, indicating a disruption in their typical interaction pattern. Consequently, when generating counterfactuals for anomalies in *Sensor5*, the correlation between the counterfactuals of *Sensor5* and the normal instances of *Sensor6* should closely resemble the high correlation observed under normal conditions.

Drift Detection

It assesses how the distributions of data points transition from normal operational states to anomalous conditions. A comparative analysis of Gaussian Kernel Density Estimation (KDE) fitted distributions between a normal dataset and an anomalous dataset reveals the extent of distributional drift—a marker of anomalies.

The impact of counterfactual explanations is then critically evaluated by observing their influence on this distributional drift. Successful counterfactual explanations should demonstrate an alignment of its data distribution with that of the normal state, indicative of an effective simulation of typical sensor behavior.

Statistical methods like the Jensen-Shannon and Wasserstein distance metrics provide a measure of how much the distributions have diverged, offering a way to quantify drift. When evaluating the effectiveness of counterfactuals, we look for reductions in these distances, indicating that the counterfactuals are effectively simulating normal conditions. The Kolmogorov-Smirnov test complements these by assessing the similarity between the empirical distribution functions of the pre- and post-counterfactual datasets, with smaller test statistics suggesting a closer match to the normal data distribution.

Visualizations further clarify this adjustment, *pairplots* compare sensor distributions pre- and post-counterfactual intervention, with the latter ideally mirroring the normative state, thereby substantiating the counterfactuals' efficacy in addressing drift.

For example, consider an industrial process where *Sensor 0* measures the flow rate and *Sensor 3* (Anomaly Type 2) measures the quality of the output product. Under normal conditions, there is a certain distribution pattern between the flow rate and product quality, suggesting a balance that ensures high-quality production. An anomaly, such as a sudden drop in flow rate due to equipment failure, would lead to a detectable shift in the product quality, thus altering the distribution pattern observed between *Sensor 0* and *Sensor 3*. Counterfactuals generated in this scenario would aim to identify modifications to the flow rate (simulated by adjustments in *Sensor 0*'s readings) that could potentially bring the product quality back to its desired level. By applying Drift Detection methods, we can qualitatively assess how closely the counterfactual-modified distributions align with the original, normal operational state distributions. The effectiveness of these counterfactuals is then substantiated through observed reductions in distributional drift, measured by the aforementioned statistical distances, and visually confirmed through comparative pairplots, highlighting the counterfactuals' success in mitigating the anomaly-induced drift and restoring operational normalcy.

This analysis supports the broader evaluation framework, offering a refined perspective on the counterfactual explanations' capacity to address anomalies. Such qualitative assessments are invaluable, particularly in settings where domain knowledge may be limited, positioning drift detection as a versatile evaluative tool within the anomaly detection and interpretability landscape.

3.2 Related Works

The purpose of this section is to provide a comprehensive and an in-depth literature review by examining existing research with respect to deep learning for anomaly detection, with a subsequent focus on various approaches to generation of counterfactual explanations

for black-box models, particularly applied to multivariate time series data and consequently their evaluation. By doing so, it will provide a solid foundation of evidence, inform methodological choices for this thesis, and situate the research within the broader academic and practical contexts.

3.2.1 Anomaly Detection

Anomaly detection is the process of identifying instances in datasets that deviate significantly from other observations [11]. This section outlines the evolution of anomaly detection techniques from statistical methods to advanced deep learning approaches.

3.2.1.1 Traditional Anomaly Detection Methods

- **Statistical Methods:** Employ proximity techniques using various modifications of the k-nearest neighbor algorithm with a suitable distance metric such as Euclidean distance or Mahalanobis distance, or parametric methods such as Minimum Volume Ellipsoid Estimation (MVE), which fits a smallest permissible ellipsoid volume around the majority of the data distribution model. However, such techniques cannot scale to increasing amounts of data [4] and also suffer from the "curse of dimensionality".
- **Classification by Methodology:** Research in terms of traditional methods can be classified into five distinct methodologies, namely, clustering, density-estimation, distance-based, reconstruction, and forecasting based methods [33],[5]. For instance, clustering methods include k-means and DBSCAN; density estimation methods include Gaussian Mixture Models and Kernel Density Estimation; distance-based methods include k-Nearest Neighbors and Local Outlier Factor; reconstruction methods include Principal Component Analysis (PCA); and forecasting-based methods include ARIMA.

3.2.1.2 Advancement to Deep Learning Approaches

With the rapid advancement of deep learning algorithms and methodologies, it has become apparent that deep learning methods outperform traditional approaches [34], particularly in unsupervised learning paradigms in machine learning, where the data is completely unlabelled. Furthermore, the use of 1D convolutions to detect anomalies for univariate/multivariate time series data in order to capture the temporal dependencies has been extensively surveyed [35]. Following are some of the DL paradigms applied to the anomaly detection task:

- **Autoencoders (AEs):** One of the fundamental paradigms in unsupervised deep anomaly detection models. AEs, especially C-AE, have been shown to learn subtle patterns such as non-linear correlations between features, avoiding complex computations [12],[36],[13] and falls under the umbrella of reconstruction based anomaly detection.
- **Variational Autoencoders (VAEs):** The C-VAE, a probabilistic model providing a probability measure rather than a reconstruction error as an anomaly score, has shown better performance [16] and are classified as reconstruction based anomaly detection.
- **Forecasting Based Models:** Primarily consist of Recurrent Neural Network (RNN) based architectures such as LSTMs, which predict subsequences compared to actual values to determine the degree of anomaly [5]. While, this approach is promising, it is inefficient for processing long sequences due to its dependency on window size and forecasting based methods have been shown that they are not robust to rapidly and continuously changing time series [5],[37]

- **Hybrid Approaches:** A combination of reconstruction and forecasting methods [5], such as a C-VAE combined with an LSTM module, offers advantages of both approaches [38]. In such an approach, the VAE model summarizes local information in a low dimensional embedding while the LSTM model acts on these embeddings to learn sequential patterns over longer time frames [38].
- **Attention Mechanisms and GANs:** Recent research has explored attention-based mechanisms such as MSCRED [39] and the use of Generative Adversarial Networks (GANs) [40] in anomaly detection, enhancing performance and interpretability.
- **Transformers:** Overcoming the limitations of RNNs in sequential processing, transformer-based architectures [41] have been implemented for anomaly detection [42],[43], showing promising results in processing clinical time series data [44].

3.2.2 Explainable AI: Counterfactual Explanations

There is a major shortcoming with respect to using the black-box models mentioned above, in the sense that they are weak in explaining their inference process and in its current form it is often viewed as less scientific [45]. Furthermore, there is the risk of inheriting human biases and prejudices from the large amounts of data the models are trained on [19]. The concept of Explainable AI is not new and has been the subject of research since the 1970's [46] and was motivated by saying that a program, modelling an expert in a given domain would be more accepted by experts of that particular domain if it could explain its actions [46]. In terms of relevance, the European Union granted their citizens a "right to explanations" under the GDPR if they are affected by algorithmic decision making [45],[47] and additionally even the Chinese government has given a high priority to develop highly explainable AI [48]. In order for these technologies to be incorporated in safety critical industries such as self-driving cars, personalized medicines and given the ubiquitous nature of time series data and its applications in such domains, it becomes imperative that ML techniques such as DL, applied to it are transparent and interpretable [49]. There are various definitions of the term *interpretability*, but in the context of ML, it is the process of being able to provide meaning in understandable terms about how a model reached a particular decision [19].

The concept, counterfactual explanations, was first introduced by Wachter [6] and there is a general consensus [7],[50] that it is described as the smallest possible change to the features in order to achieve an alternate outcome/prediction. Counterfactuals are intuitive to humans due to the nature of *what if* analysis it provides [7] and additionally due to its causal nature and through the concept of unit level counterfactuals [51] which enables explanation of changes in outcome under different treatment conditions, a very high level of interpretability can be achieved [50]. The generation of *Counterfactual Explanations* has been categorized by Guidotti [50] based on strategy to retrieve them as:

- **Optimization Based:** Counterfactual explanations generated by solving optimization problems typically employ a loss function that modifies an input instance in order to generate a counterfactual [50].
  - The most famous is the one proposed by Wachter [6] where the author proposes that the loss function can be minimized through any suitable optimization algorithm such as Nelder-Mead method, and if the model is differentiable then gradient based methods such as gradient descent or SGD or Adam [15] can be employed.
  - In CEM [52], the author defines a counterfactual as an original instance to which a perturbation is applied such that the model predicts an alternate outcome and these counterfactuals are retrieved by minimizing the loss function as a result of optimizing the FISTA [53], an optimization algorithm designed to solve convex

optimization problems efficiently by combining gradient descent with a proximal operator to achieve a fast convergence rate.

  - The authors of DICE [26] solve a gradient-based optimization to generate counterfactuals and additionally, they also propose novel methods to generate diverse counterfactuals by adding a regularization term to the loss function in order to penalize similar counterfactuals.
  - POLYJUICE [54] is a framework designed to produce counterfactuals as a conditional text generation task using language models for text data.
  - These authors [55], [56] propose a gradient-based approach to generate plausible and reliable counterfactuals for anomaly detection.

- **Heuristic Search Strategy:** Heuristic Search Strategies update the counterfactual solution iteratively to minimize one or more objective functions, based on local, heuristic decisions to achieve a valid counterfactual similar to the original instance [50].

  - MOC translates the counterfactual search into a multi-objective optimization problem using a modification of the genetic algorithm to return a diverse set of counterfactual explanations [57].
  - CERTIFAI generates counterfactual explanations using a custom genetic algorithm that respects the validity and proximity properties [58].
  - GSG method generates a counterfactual by expanding a sphere of synthetic instances around an instance of interest in all directions using a uniform distribution and halving its radius until the closest counterfactual instance is found [59].

- **Instance Based:** Instance based strategies directly searches the reference population instance for counterfactual generation [50] such as FACE [60] which argues that state of the art counterfactual generation do not necessarily produce counterfactuals that are representative of underlying data distribution and proposes a novel method of counterfactual generation by discovering feasible paths i.e., shortest path distances to identify plausible and actionable counterfactuals.

The major shortcoming of the methods listed above is that none of them have an inherent mechanism in place that selects the features to be modified to generate counterfactuals, rather, they end up modifying all the features while minimizing distance or they dynamically choose features during the optimization process. The former leads to low sparsity, in other words complex counterfactuals that are counterproductive and the latter may produce sub-optimal counterfactual explanations.

Once the counterfactual explanations have been generated, there is also a need to evaluate them using quantitative and qualitative metrics. These metrics can also be thought of as some properties any good counterfactual ought to have and quantitative evaluation metrics have been discussed by [19],[61] in their survey as the following:

- *Validity* is a measure of whether the generated counterfactual explanations changed the prediction outcome or not.

- *Sparsity* is a measure of feature modifications required to generate a counterfactual explanation and one that is sparse, that is, lower number of feature modifications is preferred due its lower complexity.

- *Diversity* states that given a single instance of interest a set of valid counterfactuals rather than a single counterfactual, must be generated while being similar and plausible, a few existing works [26],[56],[62],[55] have modified the optimization problem in order to generate diverse counterfactual explanations and consequently compute a diversity loss.

Evaluating the quality of counterfactual explanations is challenging and often specific to the application domain. One important measure is Plausibility/Feasibility, which can be understood in statistical terms as the similarity of the distributions from which the data was generated. Several methods have been proposed to compute this measure. For instance, the concept of ϵ-chain distance, as described by Laugel et al. [59], is one approach to assess the plausibility of a counterfactual. Another perspective, proposed by Artelt [63], suggests that a plausible counterfactual should lie in a dense region relative to instances in the reference population. Additionally, researchers [64],[55],[56] have employed visualization techniques like t-SNE to visually inspect whether the generated counterfactual explanations cluster with normal data.

4 Method

In this chapter, we detail the methodological framework adopted for developing and evaluating an anomaly detection model, augmented with counterfactual explanations for enhanced interpretability. Initially, this chapter introduces the framework, then data collection and preprocessing strategies, crucial for preparing the data-set for modeling process. This sets the stage for discussing the design and architecture of the anomaly detection model.

Further, the author explores methods for creation of counterfactual explanations aimed at demystifying the model's decisions. This dual focus not only aims at detecting anomalies but also at providing insights into the underlying reasons for such detections, thereby marrying the objectives of accuracy and interpretability in model evaluation. The narrative is designed to offer a comprehensive yet concise overview of the entire process from data handling to the generation and evaluation of counterfactual explanations, ensuring the replicability and robustness of the research conducted.

4.1 Framework: Anomaly Detection and Counterfactual Explanation

The framework (Figure 4.1) depicted in the image provides a structured approach to anomaly detection and the generation of counterfactual explanations. It begins with the Data Collection phase, where features of interest are selected followed by data imputation and normalization processes to prepare the data for analysis. Two models of anomaly detection are employed: the C-AE and the C-VAE, which are types of neural networks specifically designed for tasks like this.

Anomaly detection is performed in a black-box manner, where the system identifies whether the data is anomalous or not. Once an anomaly is detected, reconstruction errors from the models are fed into a Local Outlier Factor (LOF) based feature selector which pinpoints the specific features to modify for generating counterfactual explanations.

Counterfactual explanations methods are then applied using two distinct approaches: gradient and genetic algorithm. These methods aim to understand and explain the decisions made by the anomaly detection models by altering the identified features to change the outcome.

Finally, the framework includes a comprehensive quantitative and qualitative evaluation, assessing the effectiveness of the counterfactual explanations in terms of both measurable performance and interpretability. This thorough evaluation ensures that the generated expla-

nations are not only statistically valid but also meaningful and useful for understanding the model's decisions.

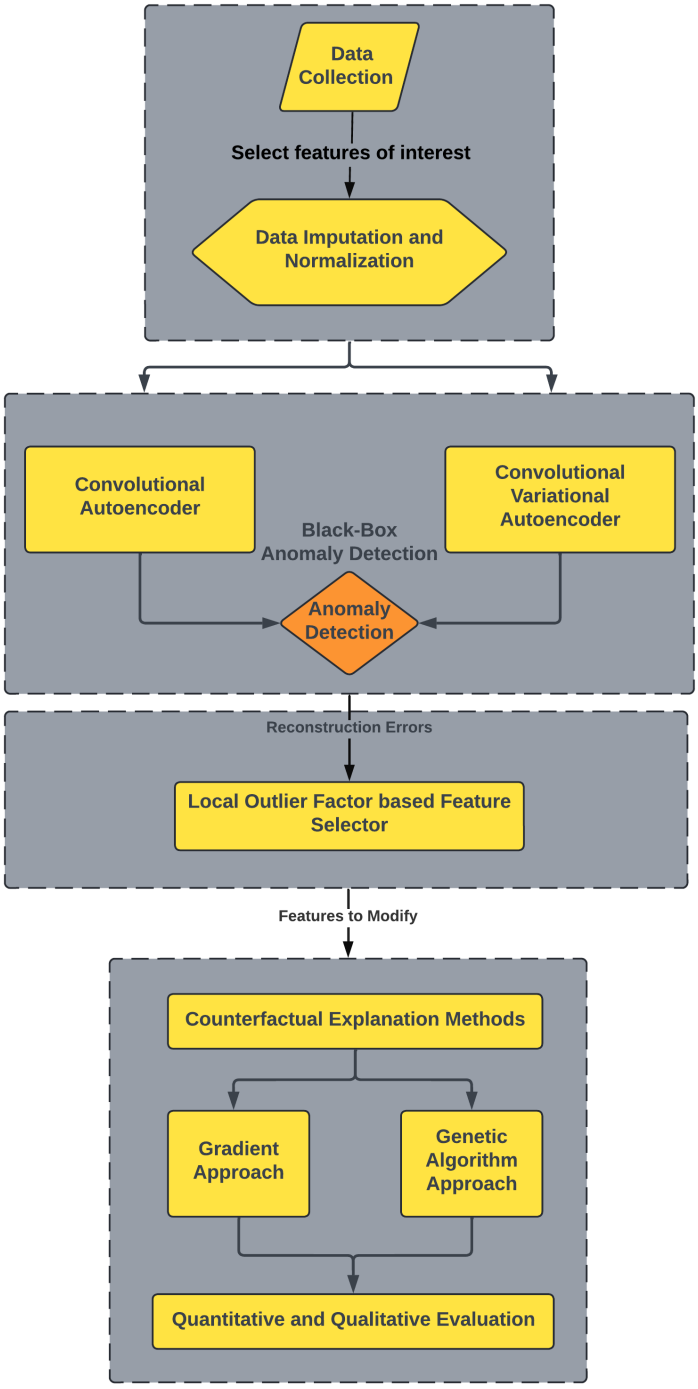


Figure 4.1: A systematic flowchart illustrating the process from data collection to quantitative and qualitative evaluation of counterfactual explanations in anomaly detection.

4.2 Data

This section is designed to offer insights into the methodologies applied for data collection and pre-processing prior to delving into the specifics of model architecture and development.

4.2.1 Data Preprocessing

The data pre-processing has been structured into several key steps, each designed to ready the data for effective model training and evaluation.

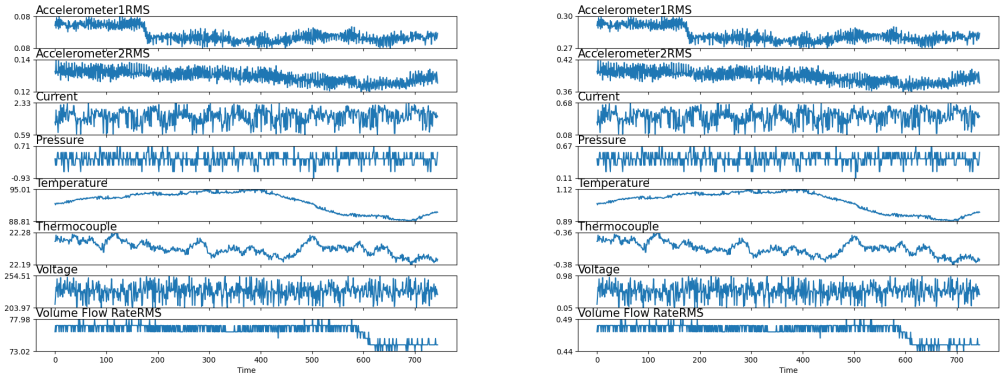


Figure 4.2: The image displays ten raw signals on the left alongside their scaled and normalized versions on the right, highlighting the effects of data standardization for the SKAB dataset.

4.2.1.1 Data Import and Initial Processing

The necessary data are imported and stored as *dataframes* using the *Pandas* library in Python. Additionally, only a select number of signals are chosen that were deemed necessary for anomaly detection, however, the selection of features was an iterative process based on model performance.

4.2.1.2 Data Normalization

- *Training Data:* The data after being imported had to be normalized to account for the varying scales of values among the features. First, the training data was normalized using the *MinMaxScaler* from Python's *sklearn* library, an important thing to note here is that only the continuous valued features were normalized. Furthermore, in order to impute any missing data, a fill-forward operation using a *'ffill'* method is used to ensure continuity in the time series. See Figure 4.2 for the effect of imputation and scaling.
- *Validation and Test Data Preprocessing:* Both validation and test datasets undergo a similar data imputation process and the same scaler object that was used to transform the training data is applied to maintain consistency in scaling across the datasets.

4.2.1.3 Sequence Data Preparation

As mentioned earlier, the data is a *multivariate time series* in nature and to maintain the temporal dependecies in the data, we employ a sliding window approach to further transform the data before model training.

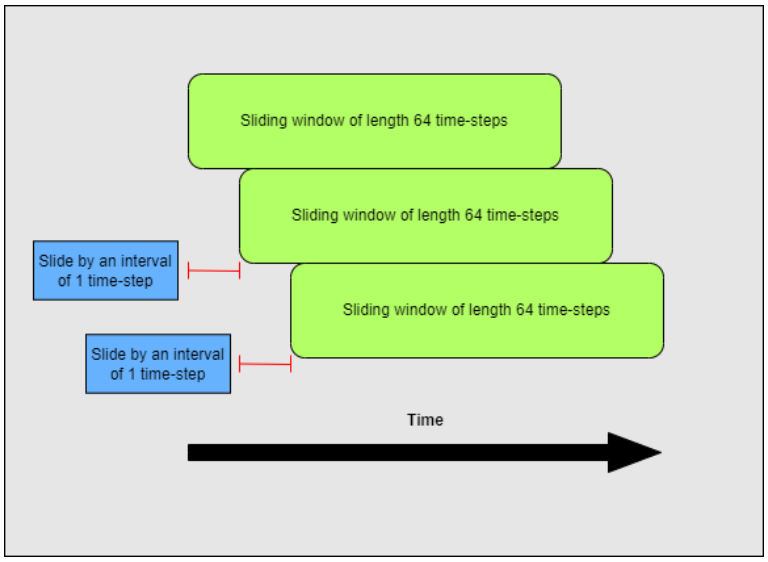


Figure 4.3: Visualization of a sliding window technique with a stride of one time-step, demonstrating how each consecutive window in a time-series overlaps the previous one by all but one point.

- **Generation of Overlapping Sequences:** To transform the preprocessed time-series data into sequences that overlap, a sliding window technique with a stride of 1 and a window length of 64 is employed to generate the sequences. The overlapping nature of these sequences, achieved through the sliding window approach (Figure 4.3), significantly enhances the model's ability to discern and learn from temporal patterns by providing it with densely sampled segments of the time series. This sliding window technique helps increase data density, as it generates more data points from the same original series, allowing the model to learn more effectively from a richer and more detailed dataset.

4.2.1.4 Data Loader Preparation

The model development was done in the *PyTorch* framework for machine learning in Python, as a result of which, the data needs to be in the form of tensors and to further optimize the training process, `Dataloader` objects are created.

- *Conversion to PyTorch Tensors and DataLoader Creation:* The preprocessed and sequenced data are converted into *PyTorch* tensors, facilitating their use in neural network models. The continuous and binary sequences are concatenated, and the resulting tensor is used to create a `TensorDataset` object. A `DataLoader` object is then instantiated for each dataset (training, validation, and anomaly test data), specifying batch size of 32 for training and test data, and a batch size of 4 for validation data. The data is not shuffled in order to respect the temporal dependencies that exist in the data. This step optimizes data handling during model training and evaluation, allowing for efficient batching and optional shuffling to improve model generalization.

4.3 Anomaly Detection

Now that we have explained how the data was prepared and transformed into overlapping sequences, we will move on to discuss the model architectures and method used for detecting anomalies. The following sections describe the models and techniques applied to spot

unusual patterns in the time-series data. We will look at how these models work, both in theory and in practice, and how effective they are at identifying anomalies.

4.3.1 Autoencoder

This subsection will present the implementation details of the C-AE model architecture for the sake of replicability in terms of the number of layers, channels, kernel sizes, padding sizes and the number of epochs to train the model.

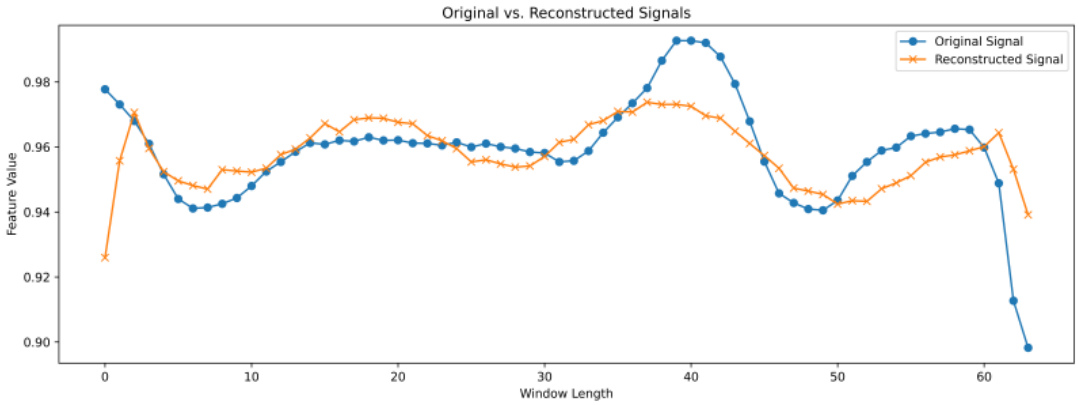


Figure 4.4: The image compares an original signal with its AE-based reconstructed version, demonstrating the model's capability in signal reconstruction.

4.3.1.1 Model Architecture

- **Encoder Layers:** The encoder utilizes sequential convolutional layers to progressively transform and compress the input time-series data into a latent space representation:
  1. A 1D convolutional layer with the number of input channels equal to the number of features which in this case is 10, 32 output channels, a kernel size of 5, and padding of 2, followed by a ReLU activation function. This layer aims to capture the initial level of temporal features from the input data.
  2. A subsequent 1D convolutional layer increases the depth to 64 output channels with the same kernel size and padding, again followed by a ReLU activation function, further abstracting the data into a higher-level representation.
- **Fully Connected Layers (Encoding):** After convolutional encoding, the model's architecture transitions through a series of linear transformations:
  1. The first linear layer reduces the feature dimension from 64 to 32.
  2. The dimensionality is further reduced from 32 to 16, and then to 8 units, establishing a compact bottleneck that captures the essence of the input data's temporal dynamics.
- **Fully Connected Layers (Decoding):** To reconstruct the input data from the bottleneck representation, the architecture symmetrically expands the latent space representation:
  1. Starting with a linear expansion from 8 to 16 units.
  2. Followed by expansions from 16 to 32, and then from 32 to 64 units, preparing the latent representation for decoding back into the time-series space.

- **Decoder Layers:** The decoder mirrors the encoder structure, utilizing transposed convolutional layers to upscale the latent representation back to the original input dimension:

    1. A 1D transposed convolutional layer with 64 input channels, 32 output channels, a kernel size of 5, and padding of 2, followed by a ReLU activation function, begins the process of reconstructing the time-series data from its latent space representation.
    2. The final layer, another 1D transposed convolutional layer, returns to the original number of features as output channels, with the same kernel size and padding, concluding with a Tanh activation function to produce the reconstructed time-series data.

The ReLU activation functions within the encoder and decoder facilitate non-linear transformation during the encoding process, while the ReLU activation in the final decoder layer serves to normalize the output, aligning it within the range of the normalized input data. The model is trained over 100 epochs, using the Adam optimizer with a learning rate of 0.001 and he AMSGrad variant enabled to ensure convergence stability, the Huber Loss being back-propagated and the loss values being recorded at each epoch for both training and validation datasets, to monitor the training process.

4.3.2 Variational Autoencoder

This subsection will present the implementation details of the C-VAE model architecture for the sake of replicability in terms of the number of layers, channels, kernel sizes, padding sizes and the number of epochs to train the model.

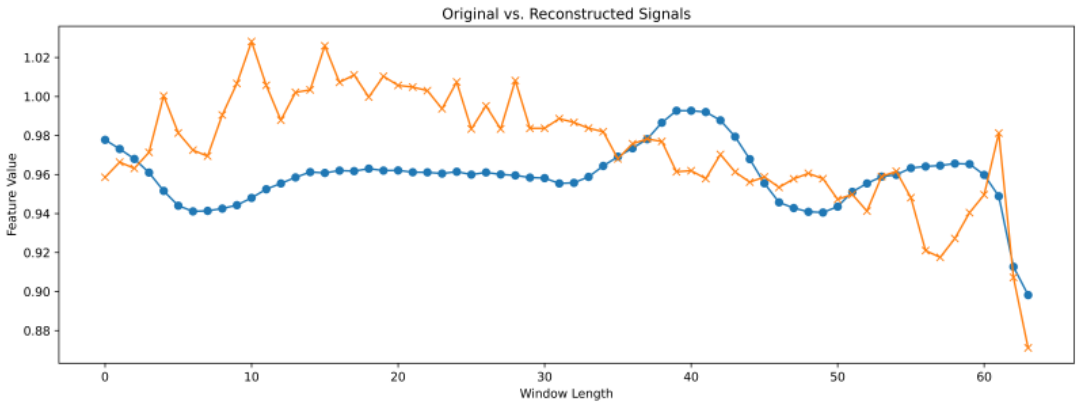


Figure 4.5: The image compares an original signal with its VAE-based reconstructed version, demonstrating the model's capability in signal reconstruction.

4.3.2.1 Model Architecture

- **Encoder Layers:** The encoder in the C-VAE model employs a sequence of 1D convolutional layers designed to efficiently encode the input time-series data into a condensed latent space representation:

    1. The first 1D convolutional layer, with an input channel size corresponding to the number of features which in this case is 10, outputs 32 channels. This layer uses a kernel size of 5 and padding of 2, followed by a ReLU activation function, aiming to extract and encode the primary temporal patterns from the data.

2. Following this, another 1D convolutional layer further condenses the data into 64 output channels, employing the same kernel size and padding, with a subsequent ReLU activation function to refine the data abstraction into a more compressed representation.

- **Fully Connected Layers (Encoding):** Sequentially, the model progresses through a set of linear transformations to further compress the data and to prepare for latent variable derivation:

    1. An initial linear transformation expands the representation from 64 to 64 units, aimed at a measured reduction in dimensionality.
    2. Subsequent layers narrow down the representation from 64 to 32 units, and eventually to 16 units. Two distinct pathways then emerge to calculate the mean (μ) and log variance ($\log(\sigma^2)$) of the latent space, indicating a bifurcation designed for the reparameterization step, with each pathway outputting 16 dimensions.

- **Reparameterization Trick:** A fundamental step in the VAE architecture, this technique allows for the sampling of a latent variable z by utilizing the derived μ and $\log(\sigma^2)$, thus enabling stochastic gradient descent through the latent space and facilitating the generation of diverse, yet plausible, data reconstructions.

- **Fully Connected Layers (Decoding):** Reconstruction from the latent space begins with fully connected layers that symmetrically expand the condensed representation:

    1. Initially, the latent space is expanded from 16 to 32 units, and further to 64 units, effectively mirroring the encoder's compressive path in reverse, setting the stage for the decoding process.

- **Decoder Layers:** The reconstruction phase concludes with a series of 1D transpose convolutional layers that gradually reconstruct the data back to its original feature space:

    1. The initial 1D transposed convolutional layer transforms the 64-channel representation back down to 32 channels, utilizing a kernel size of 5 and padding of 2, accompanied by a ReLU activation function to initiate the reconstruction of the time-series data.
    2. The final layer, another 1D transposed convolutional layer, adjusts the channel size back to the original number of features, with the same kernel size and padding, employing a ReLU activation function to finalize the reconstruction of the time-series data.

The model is trained over 100 epochs, using the Adam optimizer, a custom loss function (3.3) being back-propagated and the loss values being recorded at each epoch for both training and validation datasets, to monitor the training process.

4.3.3 Reconstruction Error Based Anomaly Detection

The process of detecting anomalies after training both the AE and VAE in the datasets are operationalized through the following key steps:

- *Reconstruction Error Calculation:* For each data point within the training, validation, and test sets, the reconstruction error is computed. This is achieved by calculating the sum of the Squared Error and Absolute Error for the continuous features between the reconstructed signals and the original sequences (Figure 4.4 and Figure 4.5). The reason for using the Squared Error is because it penalizes larger deviations and can detect significant anomalies, while the Absolute Error ensures a level of robustness to legitimate

variations in the data that are not anomalies. Mathematically, the reconstruction error for each element $x^{(i)}_{t:t+w}$ in the window can be expressed as:

$$E^{(i)}_{t:t+w} = \text{Squared Error}(x^{(i)}_{t:t+w}, \hat{x}^{(i)}_{t:t+w}) + \text{Absolute Error}(x^{(i)}_{t:t+w}, \hat{x}^{(i)}_{t:t+w}) \tag{4.1}$$

This operation is performed element-wise across the window, maintaining the original 3-dimensional shape of the data tensor.

- *Loss Aggregation:* The reconstruction errors for each window are aggregated across features to derive a single loss value per window. This value is further averaged out across all time steps and features to obtain mean sample losses for the training, validation, and test datasets. Mathematically, the aggregated loss for each window w can be expressed as:

$$L_w = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{F}\sum_{i=1}^{F} E^{(i)}_{t:t+w}\right) \tag{4.2}$$

where $E^{(i)}_{t:t+w}$ is the reconstruction error calculated in the previous step, T is the sequence length (time dimension), and F is the number of features and $L_w \in \mathbb{R}^N$ where N is the number of windows. After this step, the result is a 1D vector with a reconstruction loss per window.

- *Threshold Determination:* An anomaly detection threshold is set at the 95th percentile of the aggregated validation window losses. This statistical approach helps in identifying the upper threshold of reconstruction error that would flag a window as anomalous. Mathematically, the threshold τ can be defined as:

$$\tau = \text{Percentile}(L_w, 95) \tag{4.3}$$

where, L_w is the validation window losses aggregated as per the method shown in the previous step.

- *Labeling Anomalies:* Each window's reconstruction loss is then compared against the threshold computed in the previous step. Windows with losses exceeding this threshold are labeled as anomalous. Mathematically, for each window w, this can be expressed as:

$$A_w = \begin{cases} 1 & \text{if } L_w > \tau \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

- *Result Compilation:* A dataframe is assembled containing details such as the reconstruction loss per window, the window number, and the anomaly prediction label, which facilitates an organized analysis of the detected anomalies.

This methodology allows for a reproducible and quantitative approach to anomaly detection, hinging on the AE and VAE reconstruction error as the discriminative metric.

4.3.4 Evaluation Metrics

To assess the anomaly detection models' effectiveness through metrics like *recall*, *F1-score*, and *False Positive Rate*, first, we must generate *ground truth* labels. These labels are derived from timestamps in *Unix* epoch time, a system counting seconds since $00:00:00$ UTC on January 1, 1970. Using the `pandas` library, timestamps are converted to readable datetime

strings. By comparing these against known anomaly periods, we assign binary labels (1 for presence, 0 for absence) to each timestamp, indicating anomalies. Now, we can measure and quantify the following metrics as we would do for any classification model:

- **Precision:** Precision quantifies the accuracy of the anomaly detection model in identifying true anomalies among all detected anomalies. It is defined as the ratio of true positive predictions to the total number of positive predictions (both true positives and false positives). The formula for precision is given by:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.5}$$

  where TP represents true positives and FP denotes false positives.

- **Recall:** Recall assesses the model's capability to identify all actual anomalies within the dataset. It is calculated as the ratio of true positive predictions to the actual number of anomalies (true positives and false negatives). The formula for recall is:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.6}$$

  where FN stands for false negatives.

- **F1 Score:** The F1 Score provides a harmonic mean of precision and recall, serving as a composite metric to evaluate the model's performance considering both the precision and the recall. The F1 Score is calculated using the following formula:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.7}$$

- **False Positive Rate (FPR):** FPR measures the proportion of normal instances mistakenly identified as anomalies by the model. It is defined as the ratio of false positive predictions to the total number of actual normal instances (false positives and true negatives). The formula for FPR is:

$$\text{FPR} = \frac{FP}{FP + TN} \tag{4.8}$$

  where TN is true negatives.

4.4 Local Outlier Factor Based Feature Selector

Figure 4.6 shows the *LOF* based feature selection method. It begins with a heatmap that shows the reconstruction losses from a window-based analysis, indicating how well the anomaly detection model is performing across different sensor readings or features over a series (window of length 64) of data points. The color intensity in the heatmap reflects the level of reconstruction errors, with darker colors indicating smaller errors and lighter colors showing larger errors.

By employing LOF in the context of feature selection, we can effectively identify which features exhibit anomalous behavior, facilitating the creation of clear and sparse counterfactual explanations. The LOF algorithm, applied to the window-based reconstruction errors, determines which features are outliers.

Mathematically, the LOF model is fitted to the window based reconstruction losses (Equation 4.1) by averaging out the time dimension, then the feature based reconstruction losses L_f, where $L_f \in \mathbb{R}^{N \times d}$ represents the reconstruction errors averaged over the time dimension for N time windows and d features. The LOF algorithm outputs a prediction vector $\mathbf{y} \in \{-1, 1\}^d$, where $\mathbf{y}_j = -1$ indicates that the j-th feature is an anomaly.

An indicator function $\mathbb{I}$ can be defined to identify anomalous features,

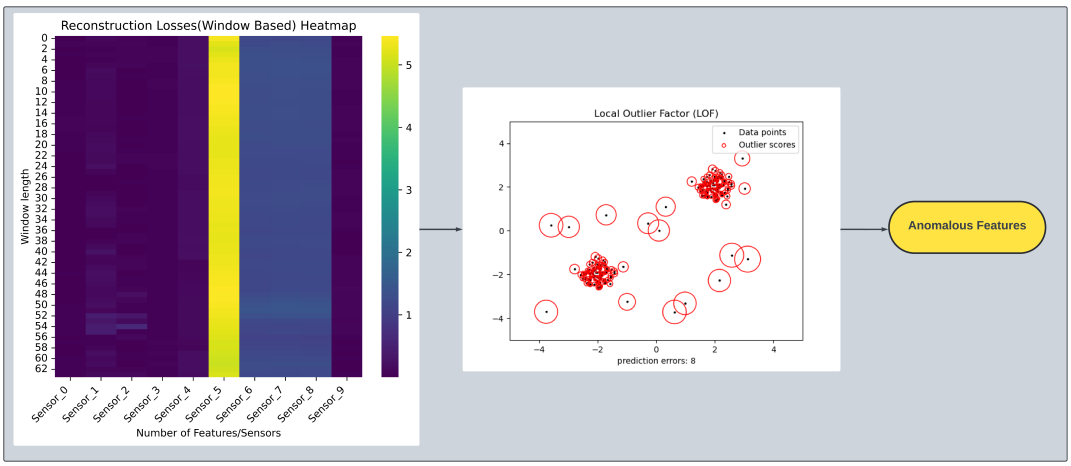


Figure 4.6: Visualization of window-based reconstruction errors feeding into an LOF for selecting features to generate sparse counterfactual explanations.

$$\mathbb{I}(j) = \begin{cases} 1 & \text{if } \mathbf{y}_j = -1, \\ 0 & \text{otherwise,} \end{cases} \tag{4.9}$$

where $j \in \{1, 2, \ldots, d\}$ represents the feature index.

After the LOF identifies the outlier features, these are the ones targeted for changes when creating counterfactual explanations. This focused method means that any changes to the data are kept to a minimum and are very specific, leading to counterfactuals that are simple to interpret and offer clear and actionable insights into the decision-making process of the anomaly detection model. The goal is to provide counterfactual explanations that are not just precise, but also practical and easy for experts to grasp.

The implementation details for the LOF-based feature selection are as follows. First, the feature based reconstruction losses, which is obtained by averaging out the time dimension from losses in Equation 4.1, are transposed and scaled using the `MinMaxScaler` from the `sklearn.preprocessing` library. The `MinMaxScaler` scales each feature individually to a given range, often between zero and one, which is useful for normalization.

Next, the `LocalOutlierFactor` class from the `sklearn.neighbors` module is used to detect anomalies. This class implements the LOF algorithm, which is an unsupervised learning algorithm used for anomaly detection. The following parameters are set:

- `n_neighbors=6`: This parameter specifies the number of neighbors to use for the LOF calculation. Choosing an appropriate number of neighbors is crucial as it affects the sensitivity of the algorithm to detect outliers.
- `contamination='auto'`: This parameter specifies the expected proportion of outliers in the data set. Setting it to 'auto' allows the algorithm to determine the contamination automatically.
- `algorithm='ball_tree'`: This parameter specifies the algorithm to use for nearest neighbor search. The `'ball_tree'` algorithm is efficient for high-dimensional data.
- `metric='euclidean'`: This parameter specifies the distance metric to use for the tree. The Euclidean distance is the default and most common metric.

The LOF model is initialized and fitted to the scaled feature loss data, and predictions are made using the `fit_predict` method, which fits the model and predicts anomalies in one

step. In the resulting prediction vector, normal data points are labeled as 1, and outliers are labeled as -1 as per the indicator function (Equation 4.9). The anomalous features are identified by finding the indices where the prediction is -1 using the `np.where` function from the `numpy` library. This approach allows for the identification of specific anomalous features, which are then targeted for modification during the generation of counterfactual explanations. The LOF-based feature selection ensures that changes are made only to features identified as anomalous, promoting sparsity and interpretability in the resulting explanations.

4.5 Counterfactual Explanation

The necessity for interpretability in black-box models is critical, given their opaque decision-making processes as per the motivations presented before. To address this challenge, this section introduces and details the implementation of various approaches for generating counterfactual explanations. These methodologies aim to enhance model transparency by providing insights into how input alterations can lead to different prediction outcomes. Furthermore, we will explore the evaluation metrics designed to assess the effectiveness and relevance of the generated counterfactual explanations. In Figure 4.7 the transition from an initial anomalous to a refined counterfactual sample as the loss decreases over iterations for different batches is visualized.

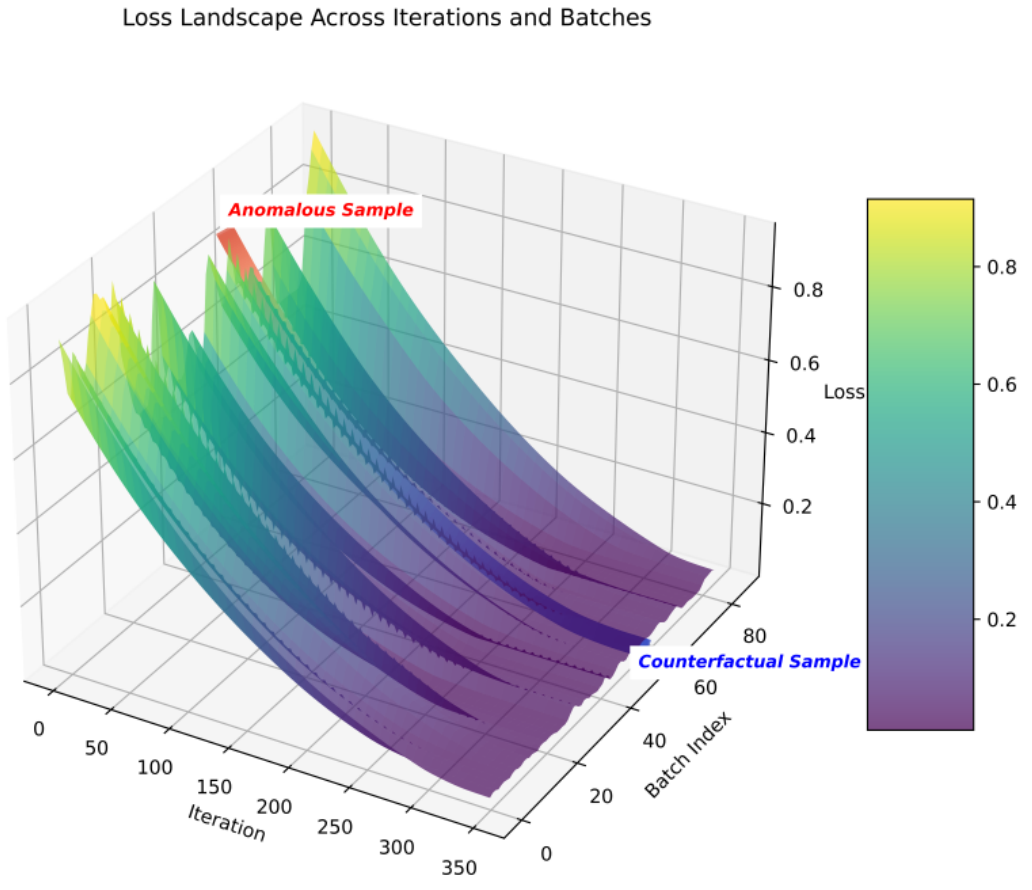


Figure 4.7: The image depicts a 3D visualization of a model's loss reduction over training iterations, highlighting the optimization path from an initial 'Anomalous Sample' to a refined 'Counterfactual Sample'.

4.5.1 Gradient Based Approach

This section discusses the implementation details for generating counterfactual explanations using a gradient-based approach. This method leverages backpropagation to compute the gradients of a differentiable black-box model, such as Autoencoders (AEs) and Variational Autoencoders (VAEs), to interpret the effect of input features.

The process begins by preparing the data and setting up the necessary tools for optimization. The model is set to evaluation mode to ensure that it does not update its parameters during the optimization process using the `eval()` function.

To generate counterfactuals, the input data is made to require gradients, allowing for gradient computation with respect to the input features. The Adam optimizer, from the `torch.optim` module, is utilized for this purpose. The Adam optimizer is chosen for its efficiency and adaptive learning rate capabilities, which help in converging faster during optimization.

The core of the gradient-based approach involves iteratively adjusting the input data to minimize the reconstruction loss. The reconstruction loss is a combination of Mean Squared Error (MSE) and Mean Absolute Error (MAE), computed element-wise between the model's output and the input.

The optimization loop runs for a predefined number of iterations. During each iteration, the following steps are performed:

- Zero the gradients of the optimizer to clear the previous gradients.
- Compute the model's output for the current input.
- Calculate the combined reconstruction loss.
- Perform backpropagation to compute the gradients of the loss with respect to the input.
- Update the input data using the computed gradients.

The gradients are computed using the `backward()` functionality in the *pytorch* library. If there are specific features to exclude from the optimization (e.g., non-anomalous features identified by the LOF-based feature selector), these features are excluded from gradient updates by setting their gradients to zero. The input data is then updated according to the gradient of the loss, using the Adam optimizer's `step()` function.

The optimization process continues until the loss falls below a predefined threshold or the maximum number of iterations is reached. This iterative process results in counterfactual explanations that are obtained by minimizing the reconstruction loss while making sparse modifications only to the anomalous features.

The final counterfactuals and the corresponding losses for each iteration are stored and returned as outputs, providing insights into the decision-making process of the anomaly detection model and ensuring the resulting counterfactuals are practical and easy to interpret.

4.5.2 Genetic Algorithm Based Approach

This section discusses the implementation details for generating counterfactual explanations using a genetic algorithm based approach. The genetic algorithm is employed from the *pymoo* framework [29], an optimization heuristic inspired by the process of natural selection. The aim is to identify perturbations to the input data that lead to significant output changes while remaining within realistic bounds, providing interpretable counterfactuals.

The implementation involves the following key steps:

- **Initialization**: The initial population is created by applying Gaussian noise perturbations around the original input instances to explore the input space near actual data

points. This is done using a custom sampling class, `CustomSampling`, which generates the initial population with perturbations. The perturbation scale is set to 0.1.

- **Fitness Evaluation**: The fitness of each individual in the population is determined by a multi-objective function that combines Mean Squared Error (MSE), Mean Absolute Error (MAE), and a custom distance metric D that accounts for the number and magnitude of changes to the input features. The custom distance metric D is a weighted sum of three norms: l_0, l_1, and l_∞, with weights $\alpha = 0.2$, $\beta = 0.7$, and $\gamma = 0.1$.
- **Selection**: Individuals are chosen for reproduction based on their fitness, favoring those with lower values that indicate more accurate and sparse counterfactuals.
- **Crossover and Mutation**: To enhance genetic diversity, the algorithm employs:
  - *Crossover*: Using the differential evolution algorithm's strategy, specified by `'real_de'`, with a crossover probability of 0.9, to generate new offspring solutions.
  - *Mutation*: Using Polynomial Mutation (PM), specified by `'real_pm'`, with a mutation probability of 0.9, to introduce random variations and maintain diversity within the population.
- **Survivor Selection**: The algorithm updates the population by selecting the fittest individuals from the current generation, often using elitist strategies to ensure high-quality solutions persist.
- **Termination**: The evolution continues for 25 generations or until a satisfactory counterfactual is found, as indicated by the fitness falling below a certain threshold.
- **Optimization Problem**: A custom problem class, `CounterfactualBatchProblem`, is defined to handle the evaluation and optimization specific to the batch of input data. This class evaluates the fitness function and updates the population accordingly. The total number of variables (`n_var`) is calculated based on the number of features to be optimized, the batch size, and the sequence length.

The `CustomSampling` class generates the initial population by applying Gaussian noise to the original data, ensuring exploration near actual data points. The fitness function combines MSE, MAE, and a custom distance metric, ensuring that the counterfactuals are close to the original input yet sufficiently altered.

The optimization algorithm is configured using the `GA` class from *pymoo*, with settings for a population size of 50, and using `'real_de'` for crossover with a probability of 0.9, and `'real_pm'` for mutation with a probability of 0.9. The `CounterfactualBatchProblem` class defines the problem to be solved, including the evaluation of the fitness function and the constraints on the input variables. This genetic algorithm-based approach ensures that counterfactual explanations are generated by exploring the input space through evolutionary strategies.

4.5.3 Evaluation Metrics

This section will deal with the implementation details of the quantitative and the qualitative evaluation metrics of the counterfactual explanations generated by the previously mentioned methods.

4.5.3.1 Quantitative Evaluation Metrics

This section discusses the implementation details for the quantitative evaluation of counterfactual explanations. The metrics considered are Validity, Distance, and Sparsity.

- **Validity**: This metric quantifies the proportion of counterfactual explanations that successfully alter the model's prediction to a different outcome than the original prediction. To implement this:
  1. Reconstruct the counterfactuals using the trained model.
  2. Calculate the reconstruction losses for the counterfactuals (Equation 4.1).
  3. Average the reconstruction losses across the time and feature dimensions to obtain a window-based loss.
  4. Compute the validity score (Equation 3.17) as the proportion of counterfactuals with a window-based loss below a predefined anomaly detection threshold.
- **Distance**: This metric assesses the average distance between the original instances and their respective counterfactuals across all features and sequences. To implement this:
  1. Calculate the Euclidean distance between the original and counterfactual instances for each time step and feature.
  2. Compute the average distance per sequence by averaging these distances across the time steps.
  3. Determine the overall average distance (Equation 3.18) by taking the mean of these average distances across all sequences.
- **Sparsity**: This metric calculates the overall average sparsity across all sequences, providing insight into the number of feature modifications required to generate each counterfactual. Ideally, a sparse counterfactual, which changes fewer features, is preferred for its simplicity and ease of interpretation. To implement this:
  1. Define a threshold for significant change.
  2. Identify the features where the absolute difference between the original and counterfactual instances exceeds this threshold.
  3. Calculate the sparsity per sequence as the ratio of significantly changed features to the total number of features.
  4. Compute the overall average sparsity (Equation 3.19) by averaging the sparsity values across all sequences.

The reconstruction losses for the counterfactuals are first computed using the trained model. These losses are combined using Equation 4.1, and then averaged across the time and feature dimensions to obtain a window-based loss.

For **Validity**, the proportion of counterfactuals with a window-based loss below a predefined threshold (τ from Equation 4.3) is calculated. This indicates how effectively the counterfactuals alter the model's prediction.

For **Distance**, the Euclidean distance between the original anomalous data and the counterfactual data is computed. The average distances per sequence are then calculated, followed by determining the overall average distance across all sequences.

For **Sparsity**, a predefined threshold ($\theta = 0.01$) for significant change is used to identify which features have changed significantly. The sparsity per sequence is calculated as the ratio of significantly changed features to the total number of features. The overall average sparsity is then computed by averaging these values across all sequences.

These metrics provide a comprehensive quantitative evaluation of the counterfactual explanations, ensuring they are valid, close to the original instances, and sparse enough to be interpretable.

4.5.3.2 Qualitative Evaluation Metrics

UMAP

Through UMAP, the aim is to observe the proximity of counterfactual instances to the normal data distribution, thereby evaluating the counterfactuals' alignment with expected, non-anomalous states. A successful counterfactual explanation method should ideally result in counterfactuals that cluster closely with normal instances, distinct from the anomalous ones. Such a pattern would suggest that the generated counterfactual explanation are not only plausible but also relevant, offering actionable insights into the conditions leading to anomalies.

The implementation involves the following steps:

- **Data Preparation**: Flatten the original train, test (anomalous), and counterfactual sequences.
- **Combining Datasets**: Combine the flattened normal and anomalous datasets into a single array for UMAP processing.
- **Label Creation**: Create labels for the combined dataset to distinguish between normal (label 0) and anomalous (label 1) data instances.
- **UMAP Application**: Apply the UMAP algorithm with specific parameters to reduce the combined dataset to a two-dimensional space.
  - `n_neighbors = 10`: This parameter controls the local neighborhood size used in manifold approximation.
  - `min_dist = 0.3`: This parameter controls the minimum distance between points in the low-dimensional space, affecting the clustering.
  - `n_components = 2`: This sets the number of dimensions for the UMAP embedding space, which is 2 for visualization purposes.
  - `metric = 'euclidean'`: This parameter specifies the distance metric to be used in the UMAP algorithm.
- **Visualization**: Transform the counterfactual data using the fitted UMAP reducer and plot the UMAP embeddings of the normal, anomalous, and counterfactual data instances.
  - Normal instances are plotted in limegreen.
  - Anomalous instances are plotted in crimson.
  - Counterfactual instances are plotted in gold.

  The plot is saved with high resolution for further analysis and reporting.

This qualitative analysis complements quantitative metrics, providing a deeper insight into the practical utility of counterfactual explanation. Such evaluations are invaluable for refining counterfactual generation techniques, ensuring they produce meaningful and understandable explanations for complex anomaly detection scenarios.

Naive Anomaly Detector Based On Correlation Loss

Leveraging domain knowledge, this approach depends on the understanding that certain sensor pairs, such as *Sensor5* and *Sensor6* in the case of Industrial Data, exhibit a strong correlation under normal operational conditions, which diminishes in the presence of anomalies.

This phenomenon supports the development of a qualitative metric to evaluate the effectiveness of counterfactual explanations generated for anomalous conditions.

The implementation involves the following steps:

- **Correlation Calculation**: Compute the Pearson correlation coefficients between *Sensor5* and *Sensor6* for normal, anomalous, and counterfactual data using the `pearsonr` function from the `scipy.stats` module.
- **Window-wise Computation**: Calculate the correlation coefficients for each window of data.
- **Handling Exceptions**: Ensure robustness by handling potential issues such as invalid inputs or constant signals, which can result in undefined correlation coefficients.
- **Visual Comparison**: Use histograms to visually compare the distribution of correlation coefficients across normal, anomalous, and counterfactual data. This visualization highlights the recovery of normal correlation patterns in the counterfactual instances.

The correlation coefficients are computed for each window and histograms are used to visualize the distribution of these coefficients, demonstrating how well the counterfactuals restore the normal correlation patterns.

This qualitative analysis provides a deeper insight into the practical utility of counterfactual explanations, ensuring they align with domain knowledge and exhibit expected correlations, thus offering actionable insights into the conditions leading to anomalies.

Drift Detection

The impact of counterfactual explanations is critically evaluated by observing their influence on this distributional drift. Successful counterfactual explanations should demonstrate an alignment of their data distribution with that of the normal state, indicative of an effective simulation of typical sensor behavior. The implementation involves the following steps:

- **Data Preparation**: Prepare the prior and post data for analysis.
  - Prior represents the normal data.
  - Post represents either the anomalous or counterfactual data.
- **Drift Detection Initialization**: Initialize the Data Drift Detector with the prior and post data. The `DataDriftDetector` class is used to perform this analysis.
- **Drift Calculation**: Calculate the drift using statistical distance metrics such as the Jensen-Shannon and Wasserstein distances, and the Kolmogorov-Smirnov test to assess distributional changes using the `calculate_drift()` method.
- **Visualization**: Generate pairplots to visualize the distributional alignment between normal, anomalous, and counterfactual data. Focus on specific sensors to illustrate the drift and its correction using the `plot_numeric_to_numeric()` method.
  - *T*he sensors selected by the LOF-based feature selector are the primary focus of such an evaluation since they are anomalous in their behaviour.
- **Comparison and Analysis**: Compare the pairplots to evaluate the effectiveness of the counterfactuals in reducing distributional drift and simulating normal conditions.

The analysis involves comparing the KDE-fitted distributions of normal, anomalous, and counterfactual data to evaluate the extent of distributional drift and the effectiveness of counterfactuals in mitigating this drift. By examining visualizations, we can assess how well the counterfactuals simulate normal conditions and address the anomalies.

This qualitative analysis supports the broader evaluation framework, offering a refined perspective on the counterfactual explanations' capacity to address anomalies. Such qualitative assessments are invaluable, particularly in settings where domain knowledge may be limited, positioning Drift Detection as a versatile evaluative tool within the anomaly detection and interpretability landscape.

5 Results and Discussion

This chapter provides a detailed examination of the results derived from the application of anomaly detection models and methods to generate counterfactual explanations to time-series data. Both SCANIA (Industrial) and SKAB datasets serve as the foundation for this analysis, offering a diverse range of scenarios to test the efficacy and robustness of the proposed methods. A multi-faceted evaluation framework is employed, combining quantitative and qualitative metrics to thoroughly assess model performance in detecting anomalies and the methods' ability in generating meaningful counterfactual explanations. Additionally, this chapter includes a meaningful discussion on the methodology, focusing on the strengths and limitations of the approaches used. The chapter also explores the implications and limitations of feature selector on enhancing model interpretability and the consistency of counterfactual generation across different data environments.

5.1 Industrial Data

5.1.1 Anomaly Detection Evaluation Metrics

The anomaly detection models, C-AE (Table 5.1) and C-VAE (Table 5.2), have been evaluated using F1-score, Recall, and False Positive Rate (FPR) across two types of anomalies.

For Anomaly Type 1, C-AE and C-VAE models show excellent detection ability with a Recall of 1 and an F1-score of 0.99, meaning they effectively identify all anomalies. However, the FPR is 1, indicating all normal instances were incorrectly flagged as anomalies. However, it is to be noted that this dataset consisting of 4746 windows of sequences has only a mere 5 non-anomalous windows based on ground truth. This high FPR reflects the dataset's nature, predominantly composed of anomalies, with only a few normal instances present by design.

In Anomaly Type 2, both models demonstrate good performance, with C-AE achieving an F1-score of 0.87 and a Recall of 0.79, and C-VAE showing a slightly better Recall of 0.80. The FPR is low at 0.06, indicating a reliable detection of anomalies with minimal false alarms. Yet, the challenge remains to adjust the models to better differentiate between normal and anomalous behavior in datasets with an inherent bias towards anomalies, especially in datasets like Anomaly Type 1.

Dataset	F1-score	Recall	FPR
Industrial Data - Anomaly Type 1	0.99	1	1
Industrial Data - Anomaly Type 2	0.87	0.79	0.06

Table 5.1: Evaluation Metrics for the C-AE based Anomaly Detection for the Industrial Dataset

Dataset	F1-score	Recall	FPR
Industrial Data - Anomaly Type 1	0.99	1	1
Industrial Data - Anomaly Type 2	0.87	0.80	0.06

Table 5.2: Evaluation Metrics for the C-VAE based Anomaly Detection for the Industrial Dataset

5.1.2 Counterfactual Explanation Evaluation Metrics

This section presents the evaluation of counterfactual explanations, segmented further into quantitative and qualitative metrics.

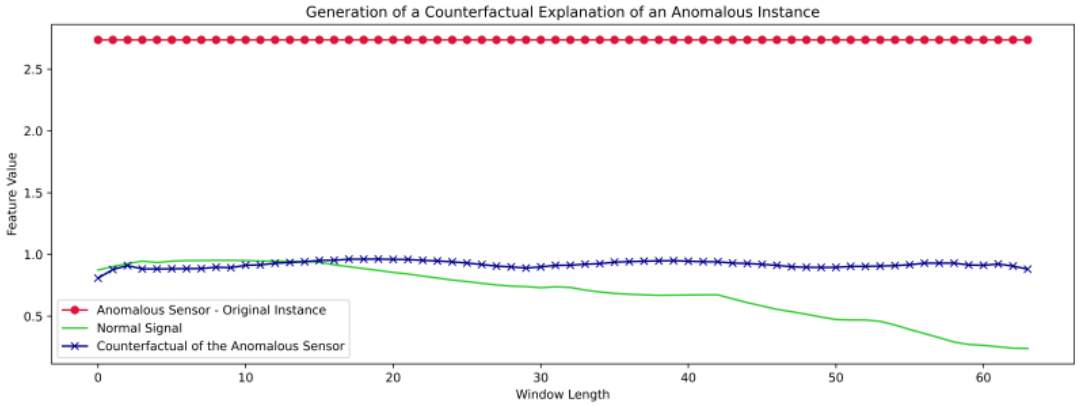


Figure 5.1: This chart illustrates the generation of a counterfactual explanation for an anomalous sensor reading. The red line represents the original instance of the anomaly, consistently high across the window length. The blue line shows the minimal adjustments needed to transform the anomalous readings into normal, depicted by the green line. This visualization highlights how small changes to the sensor values can effectively prevent anomalies, offering a clear, actionable path to maintaining normal operational conditions.

5.1.2.1 Quantitative Evaluation Metrics

Quantitative metrics, including validity, distance, and sparsity, offer objective measures to gauge the explanations' effectiveness in altering the model's predictions, their distance to the original instances, and the minimalism of modifications proposed.

	Validity ↑	Sparsity ↓	Proximity ↓
C-AE	0.9651	0.1444	0.6216
C-VAE	0.9803	0.1852	0.6263

Table 5.3: Gradient approach generated counterfactual explanations' quantitative metrics using the C-AE and C-VAE for the Industrial dataset. The up arrow indicates that a higher value is preferrable and the down arrow indicates that a lower value is preferrable.

The evaluation of counterfactual explanations through quantitative metrics (Figure 5.2, Tables 5.3 and 5.4) presents a clear distinction between the genetic algorithm and gradient-

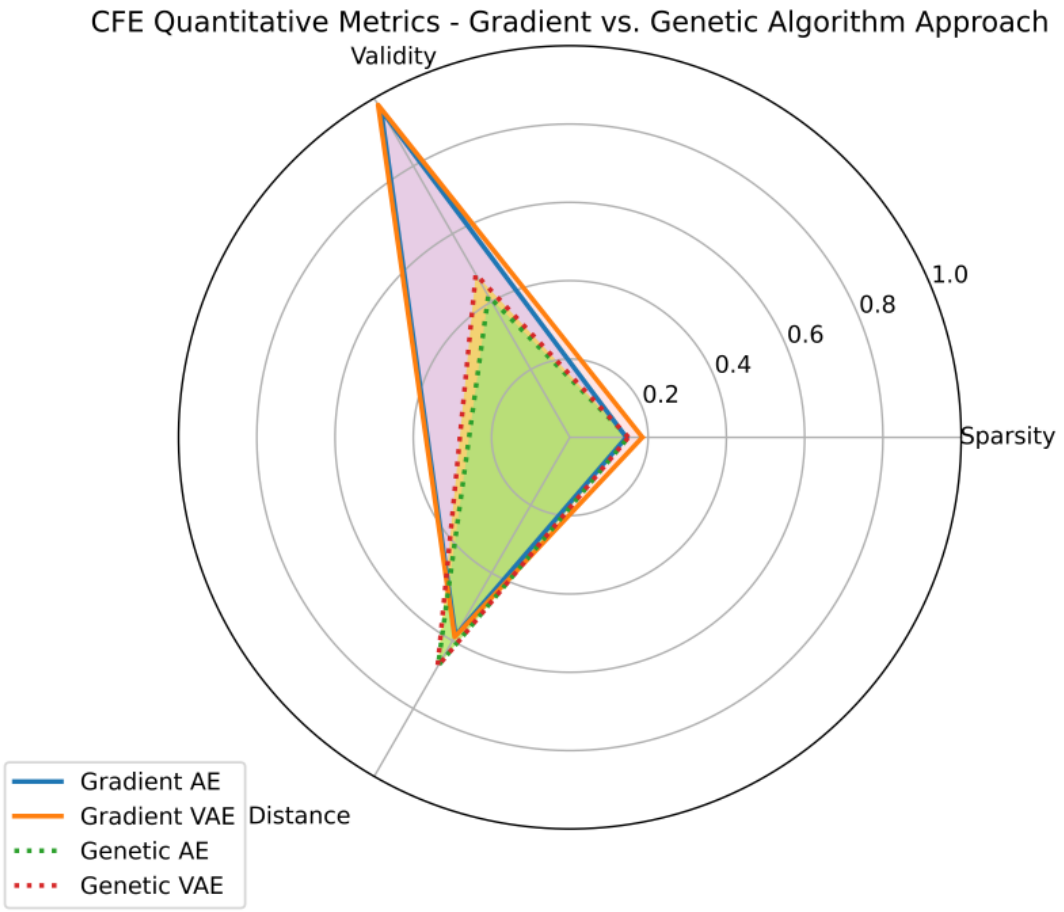


Figure 5.2: Comparative radar charts displaying counterfactual explanation (CFE) metrics—Sparsity, Validity, and Distance—for Convolutional Autoencoder (C-AE) and Variational Autoencoder (C-VAE) models using Gradient-Based and Genetic Algorithm approaches for the Industrial dataset.

	Validity ↑	Sparsity ↓	Proximity ↓
C-AE	0.4165	0.1494	0.6737
C-VAE	0.4810	0.1497	0.6760

Table 5.4: Genetic algorithm approach generated counterfactual explanations' quantitative metrics using the C-AE and C-VAE for the Industrial dataset. The up arrow indicates that a higher value is preferrable and the down arrow indicates that a lower value is preferrable.

based approaches in terms of *validity*, while there is not much of a difference in terms of *distance* & *sparsity*.

The gradient-based approach showcases a superior validity metric for both C-AE and C-VAE models, indicating a higher success rate in altering the original model's predictions to achieve the desired outcomes. On the other hand, the genetic algorithm approach, which explores a broader solution space as suggested by the slightly greater distance metric, falls short in validity. This implies that despite the genetic algorithm's counterfactuals venturing further from the original data points, they are less effective in changing the model's decision.

The trade-off between distance and validity is more pronounced in the genetic algorithm approach, where the search for a valid counterfactual solution results in more substantial alterations to the original instance, yet does not correspondingly increase the chance of achiev-

ing a different outcome. On the other hand, the gradient-Based approach achieves higher validity with closer adherence to the original data structure, highlighting its efficiency.

Upon comparing the C-AE and C-VAE models (Figure 5.2), the C-VAE displays a slightly higher sparsity across both methods, indicating a preference for slightly higher feature modifications while achieving almost similar validity metric in the gradient-based approach and in the genetic algorithm approach, the counterfactuals generated by using the C-VAE model achieves a higher validity metric. This suggests that C-VAE's counterfactuals, exhibiting a slightly higher sparsity across the gradient based approach, may be more intricate due to alterations in more features. This complexity could potentially affect the interpretability and actionability of the counterfactuals, as simplicity is often a key factor in understanding and implementing changes based on these explanations.

Concluding from the quantitative analysis, the gradient-based approach emerges as the more effective method for generating counterfactuals when compared to the genetic algorithm. It not only ensures a higher likelihood of altering the model's predictions but also maintains closer proximity to the original instances and demonstrates a more minimalistic alteration of features. This establishes the gradient-based approach as a preferable choice based on the evaluated metrics.

5.1.2.2 Qualitative Evaluation Metrics

Complementing the quantitative analysis, qualitative metrics through methods like UMAP visualization and correlation analysis between sensor readings provide a nuanced understanding of the explanations' practicality and intuitiveness. By visualizing the distribution of original, anomalous, and counterfactual instances and assessing the restoration of normal operational patterns, these evaluations underscore the counterfactuals' relevance and applicability in real-world settings.

UMAP

Here the UMAPs of the counterfactual explanations generated by gradient and genetic algorithm based approaches using C-AE and C-VAE models are presented.

For Anomaly Type 1 (Table 5.5), the gradient approach's UMAP visualization demonstrates C-AE and C-VAE counterfactuals clustering with the normal data, indicative of a clear decision boundary. The consistency of this clustering aligns with the high validity scores, reflecting the approach's effectiveness in generating desirable counterfactual outcomes.

Conversely, the genetic algorithm approach's counterfactuals for Anomaly Type 1 (Table 5.5), despite the lower validity score, largely cluster with normal data points. This visual observation is inconsistent with the expected outcome based on the quantitative analysis, suggesting that this qualitative evaluation via the UMAPs does not highlight the shortcomings of the genetic algorithm based approach.

Extending the analysis to Anomaly Type 2 (Table 5.6), the gradient approach maintains its performance with C-AE and C-VAE models, as shown by the counterfactuals' proximity to normal data clusters in the UMAP visualization. This underlines the robustness of the gradient approach across different anomaly contexts.

For the genetic algorithm with Anomaly Type 2 (Table 5.6), the counterfactuals largely exhibit a tendency to align with normal data regions which is inconsistent with the genetic algorithm's lower validity metric. The intuition is that a number of counterfactual instances in this case should have been clustered with anomalous instances so as to be consistent with the lower validity scores seen in Figure 5.2, however this is not the case and suggests that a UMAP-based evaluation alone may not be sufficient to compare the gradient and genetic algorithm based approaches for generating counterfactual explanations.

The UMAP visualizations for both types of anomalies emphasize the gradient approach's capacity to generate counterfactuals that not only statistically alter the model's predictions

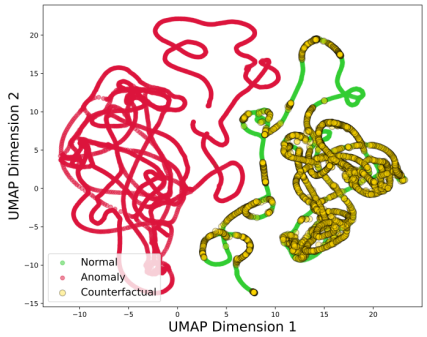
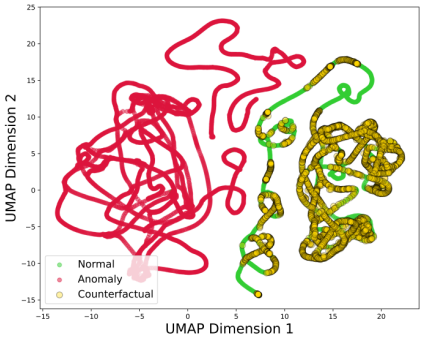
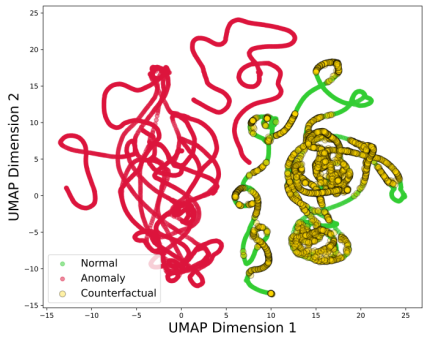
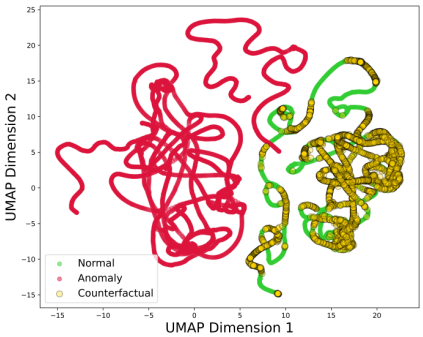
Model / Approach	C-AE	C-VAE
Gradient		
Genetic Algorithm		

Table 5.5: UMAP of the Counterfactual Explanation for C-AE and C-VAE Models generated by Gradient and Genetic Algorithm based approaches for Anomaly Type 1.

but are also qualitatively consistent with the normal operational state. On the contrary, the genetic algorithm's performance, which is quantitatively poorer is not reflected in the UMAP-based qualitative evaluation, and hence requires additional scrutiny through more definitive tools like correlation loss-based anomaly detection and data drift detection to provide a comprehensive understanding that corroborates with the quantitative metrics.

Naive Anomaly Detector based on Correlation Loss

In the realm of sensor data analysis, understanding the relationship between different sensor readings is vital. For Anomaly Type 1, domain knowledge indicates that *Sensor 5* and *Sensor 6* should exhibit a high correlation under normal operating conditions. This relationship is leveraged as a benchmark for detecting anomalies and evaluating the efficacy of counterfactual explanations. High correlation values suggest normal operation, while low values may indicate anomalous behavior.

Gradient Based Counterfactual Explanations - C-AE and C-VAE

The histograms (Figure 5.3) from the gradient-based approach for C-AE and C-VAE models displays the distribution of correlation coefficients between *Sensor 5* and *Sensor 6* and the red dashed line marks the 0.9 correlation value. Under normal conditions, a strong positive correlation can be observed, depicted by the peak towards the right of the histograms. The counterfactual data generated by both models, intuitively, should tend to shift the distribution towards this normal correlation range, and though not as tightly clustered, suggests

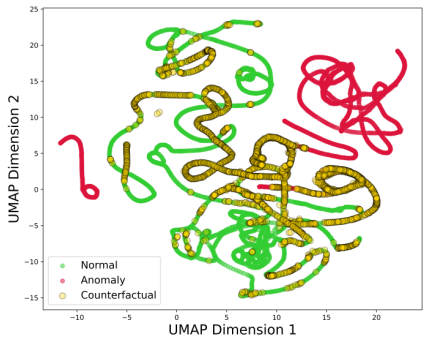
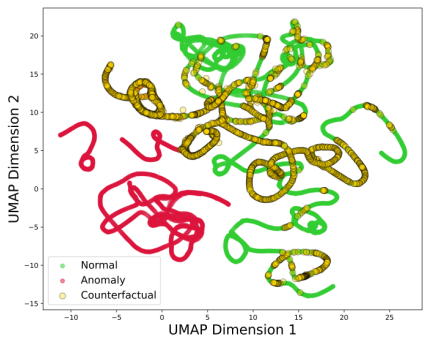
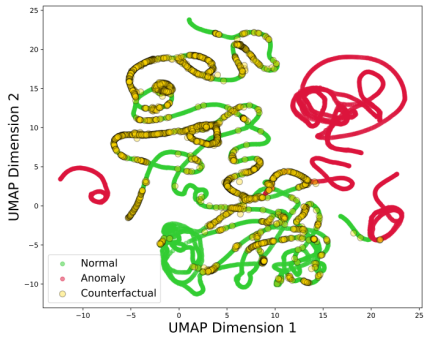
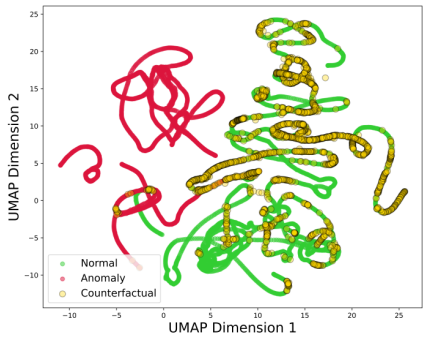
Model / Approach	C-AE	C-VAE
Gradient		
Genetic Algorithm		

Table 5.6: UMAP of the Counterfactual Explanation for C-AE and C-VAE Models generated by Gradient and Genetic Algorithm based approaches for Anomaly Type 2.

some success in generation of counterfactual explanations that promotes return to normal sensor behavior with a majority (in terms of frequency) of the correlation values being greater than 0.9.

Genetic Algorithm Based Counterfactual Explanations - C-AE and C-VAE

In contrast, the genetic algorithm approach (Figure 5.4) presents a more dispersed distribution of counterfactual correlation values, with a noticeable number of counterfactuals failing to align with the high correlation characteristic of normal operational data. This dispersion aligns with the low validity scores observed in the quantitative analysis, substantiating the qualitative evaluation's effectiveness in capturing counterfactuals that do not convincingly revert to a state of normalcy.

Comparatively, the gradient-based approach demonstrates a tighter concentration of counterfactual explanations around the high-correlation peak with correlation values mostly greater than 0.9, indicative of normal operation, unlike the genetic algorithm's broader spread. This observation suggests that the gradient-based method is more consistent in generating plausible counterfactual explanations that align with domain knowledge of sensor behavior.

The correlation loss-based anomaly detection illustrates the limitations of the UMAP visualizations. While UMAP projections suggested that most counterfactuals were clustering with normal data for the genetic algorithm approach, the correlation histograms reveal that many of these counterfactuals did not effectively capture the high correlation between *Sensor 5* and *Sensor 6*, thus failing to simulate normal operational conditions accurately.

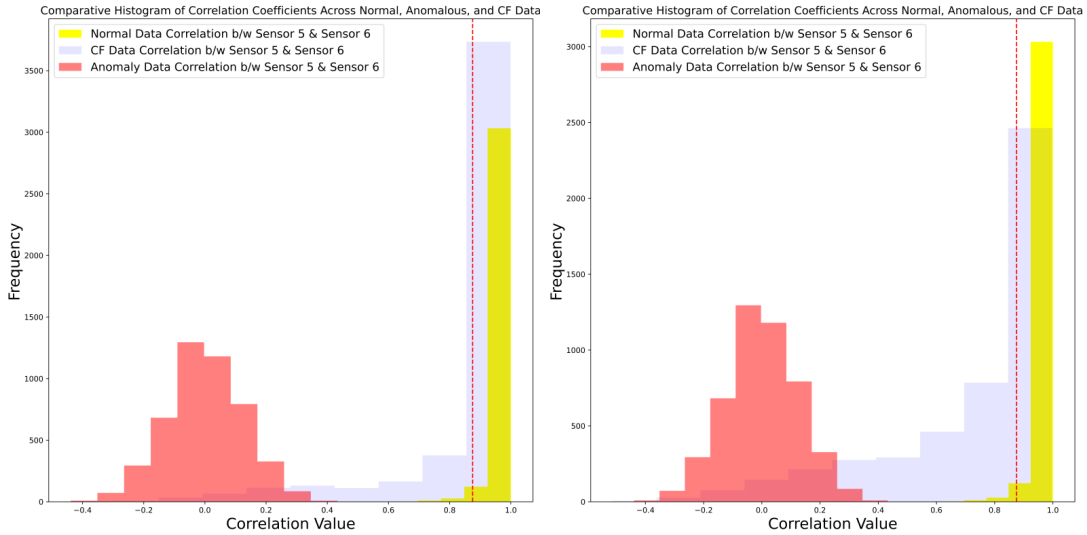


Figure 5.3: Correlation coefficient distributions for *Sensor 5* and *Sensor 6* across normal, anomalous, and counterfactual data generated from the Gradient-based approach for the C-AE (left) and C-VAE (right) models.

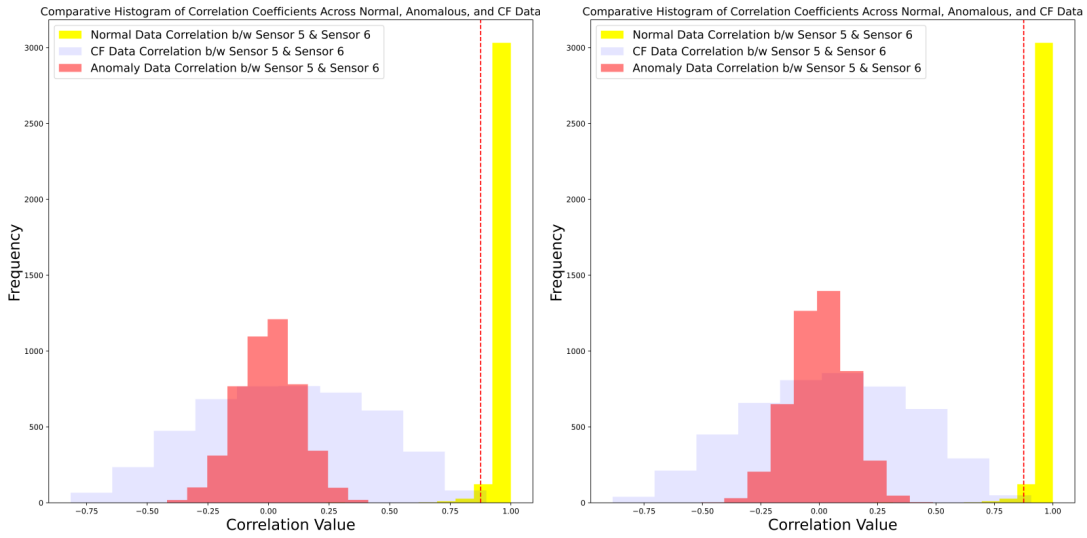


Figure 5.4: Correlation coefficient distributions for *Sensor 5* and *Sensor 6* across normal, anomalous, and counterfactual data generated from the Genetic Algorithm-based approach for the C-AE (left) and C-VAE (right) models.

This method of qualitative evaluation through correlation loss is potent when specific domain knowledge is available, as with Anomaly Type 2. However, for Anomaly Type 2, where such domain-specific relationships between sensors may not be well-defined or known, alternative methods like drift detection are employed.

Data Drift

Gradient Based Counterfactual Explanations - C-AE and C-VAE

In the absence of domain knowledge for Anomaly Type 2, data drift detection offers insight into the distributional changes particularly in the distribution patterns for *Sensor 0* and *Sensor 3*, prompted by anomalies and the corrective influence of counterfactuals. This approach is essential for understanding the adjustments made by counterfactual explanations to the data distribution, reflecting the restoration of typical operational conditions.

The left side of the Figure 5.5 presents a noteworthy shift in the distributions of normal (prior) and anomalous (post) data for both the sensors, and in the scatter plot as well the distinctive clustering of normal and anomalous data can be seen. The right side, that is the counterfactual (post) data points are now realigned closer to the training i.e., normal data's distribution, indicating the anomalies' influence has been substantially reduced. This convergence is particularly evident in the scatter plots for both the sensors, where the counterfactual data clusters closely with the normal data, indicating effective correction of the drift. This suggests that the counterfactuals generated by the C-AE model are capable of correcting the drift and restoring data points to a distribution akin to the normal operating state.

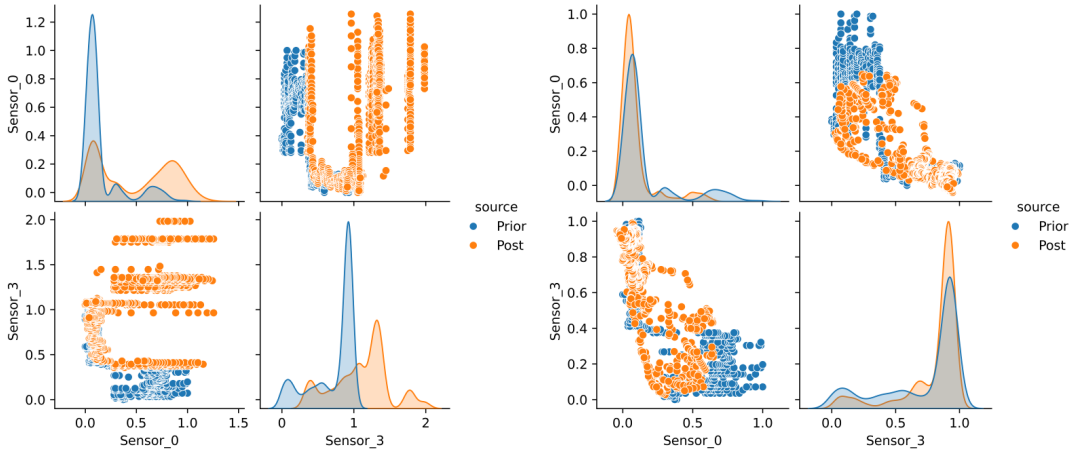


Figure 5.5: The image compares data drift in *Sensor 0* and *Sensor 3* between training and anomalous data (left) versus training and counterfactual data (right) generated by Gradient method using the C-AE.

The Figure 5.6 relates to drift detection between the sensors and the corrective action of the counterfactuals generated using the C-VAE model and gradient based approach. Similar to the C-AE model, in the left image a clear separation between the normal and anomalous data distribution and the scatter plots in both the sensors can be observed, signifying the presence of data drift due to anomalies. It can be claimed that counterfactuals generated are effective, since the right image displays a convergence of the post-counterfactual distribution with that of the normal operational data and additionally, the clustering in the scatter plot further substantiates this claim, echoing the corrective results seen with the C-AE model.

In the scatter plot comparisons for Anomaly Type 2, using the gradient-based approach for both C-AE and C-VAE models, one can discern how counterfactuals are clustered in relation to normal data.

With the C-AE model, the scatter plots reveal a substantial overlap between the counterfactual and normal data, indicating that the C-AE counterfactuals closely approximate the normal operational state. However, there's still a visible spread of counterfactual data points beyond the dense core of normal data, suggesting room for improvement in the generation of counterfactual explanations.

Shifting focus to the C-VAE model, the respective scatter plots show a more pronounced clustering of counterfactual data around the normal data points. The counterfactuals seem to encapsulate the normal data more tightly, demonstrating the C-VAE's stronger performance

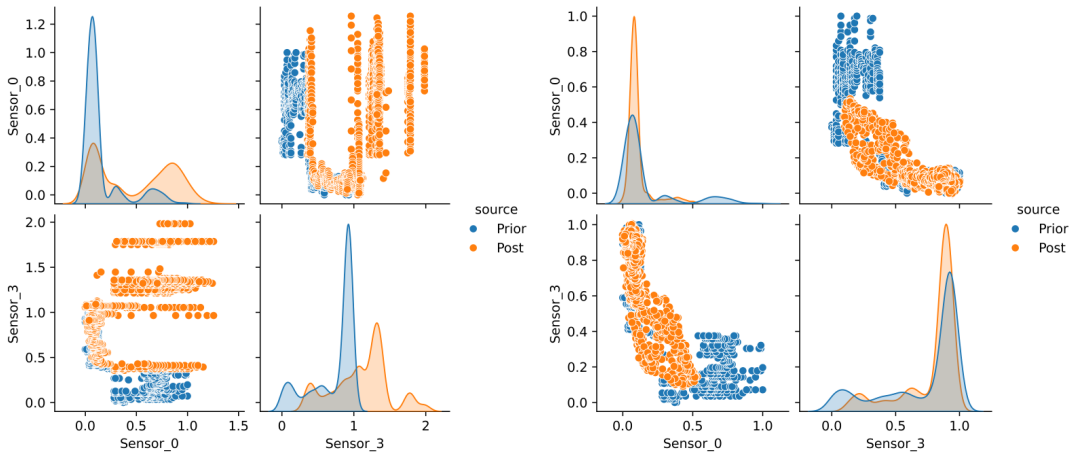


Figure 5.6: The image compares data drift in *Sensor 0* and *Sensor 3* between training and anomalous data (left) versus training and counterfactual data (right) generated by Gradient method using the C-VAE.

in generating counterfactual explanations that adhere to the expected sensor behavior in normal conditions.

Comparatively, the C-VAE's counterfactuals display a greater degree of alignment with normal data than those generated by the C-AE model. This tighter clustering in the C-VAE plots suggests that the counterfactuals not only counter the data drift effectively but also enhance the proximity to the typical data range, supporting the conclusions drawn from the quantitative validity metric. The gradient-based approach, especially with the C-VAE model, thus demonstrates a more precise capacity to generate counterfactual explanations that can restore sensor readings to a distribution indicative of normal operations.

Genetic Algorithm Based Counterfactual Explanations - C-AE and C-VAE

In the data drift detection assessment for Anomaly Type 2, the counterfactual explanations generated using the genetic algorithm approach for the C-AE model (Figure 5.7) reveal distinctive findings. The first image shows a notable divergence between the normal (prior) and anomalous (post) data, with the counterfactual (post) data points appearing to reduce this gap slightly, as indicated by a modest shift towards the distribution of the normal data. However, this shift is not as substantial as might be expected, suggesting that while there is some correction of the data drift, it is not as effective as desired.

Turning to the C-VAE model, the second image (Figure 5.8) displays a similar pattern. The genetic algorithm approach results in a slight convergence of the post-counterfactual distribution towards that of the normal data, yet, the correction remains partial, with a significant proportion of the distribution still indicating a drift from the normal operational state.

When comparing the genetic algorithm and gradient approaches, it underscores a clear distinction, not just in the realignment of data distributions but also in the clustering patterns observed in scatter plots. The gradient approach excels in both respects, demonstrating a more pronounced realignment of counterfactual data with the normal data distribution and exhibiting a dense clustering indicative of its more effective mitigation of the data drift caused by anomalies. This is evident from the scatter plots, where the gradient approach's counterfactuals display tight groupings around the normal operational state, signifying a high fidelity correction.

On the other hand, the genetic algorithm approach, shows progress with some degree of realignment; however, the scatter plot reveals that its counterfactuals are widely spread out, suggesting a less accurate restoration of normal sensor behavior. While it makes strides in

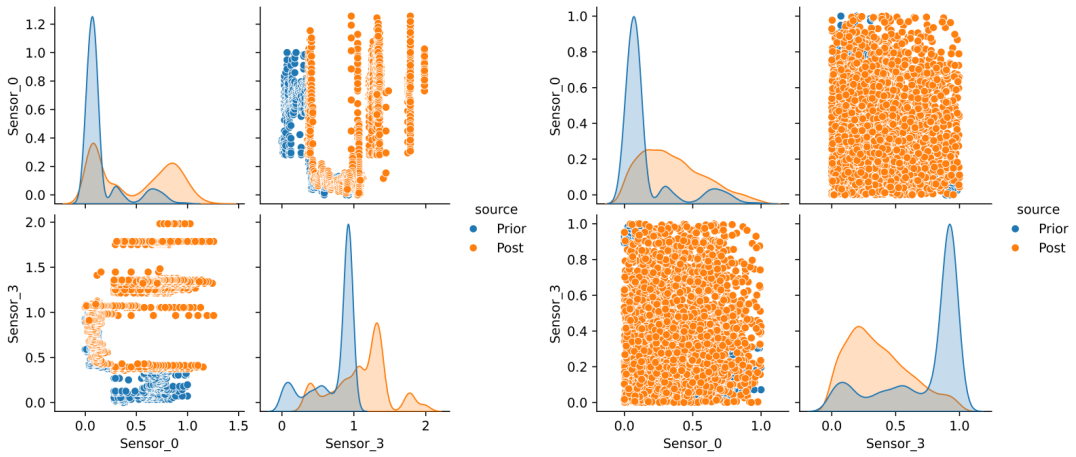


Figure 5.7: The image compares data drift in *Sensor 0* and *Sensor 3* between training and anomalous data (left) versus training and counterfactual data (right) generated by Genetic Algorithm method using the C-AE.

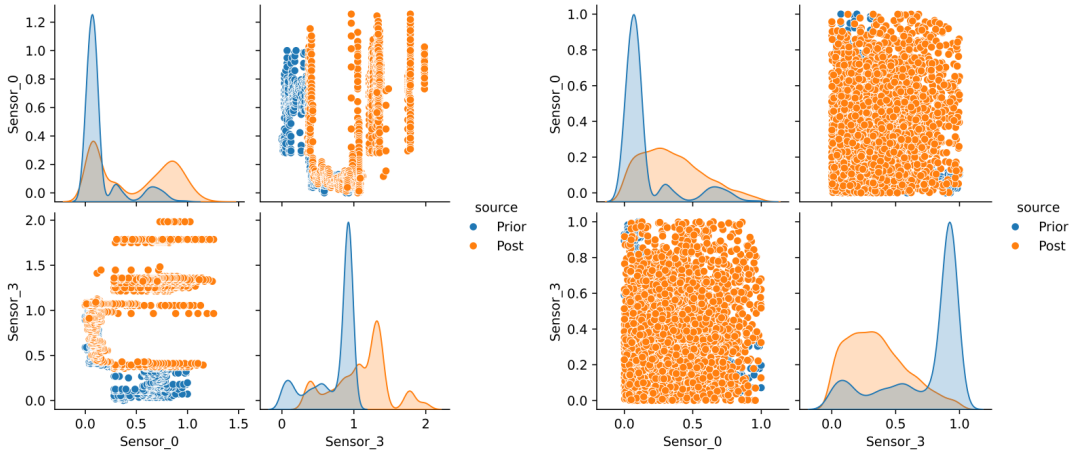


Figure 5.8: The image compares data drift in *Sensor 0* and *Sensor 3* between training and anomalous data (left) versus training and counterfactual data (right) generated by Genetic Algorithm method using the C-VAE.

reducing data drift, the approach does not achieve the same level of precision as the gradient method, with counterfactuals exhibiting greater scatter and less defined clustering.

The combined analysis of data distribution and scatter plot clustering solidifies the gradient method's superiority in generating counterfactual explanations that are closely aligned with the operational norm, effectively addressing anomalies and restoring the integrity of sensor data.

This qualitative assessment through data drift detection complements the quantitative validity metric, offering a visual representation of the counterfactuals' impact. Notably, the lower validity scores identified in the quantitative analysis for the genetic algorithm approach are again echoed here, revealing the approach's reduced efficacy in simulating a return to normalcy. This multidimensional evaluation, incorporating UMAP, correlation loss based anomaly detection and data drift detection, provides a comprehensive view of the counterfactual explanations' performance, affirming the gradient method's superiority in generating counterfactual explanations that are more closely aligned with the characteristics of normal sensor operation.

The results obtained here are consistent with the claim made by author [50], which suggests that heuristic strategies often yield efficiency by designing the solution to minimize a cost function at each iteration, albeit potentially at the expense of optimality. This is exemplified by the genetic algorithm approach that utilizes the model solely for cost function minimization to generate counterfactual explanations. On the other hand, the findings indicate that leveraging model gradients in the gradient approach not only aligns more closely with standard practices for generating counterfactual explanations but also produces higher-quality outcomes. The gradient approach's utilization of model gradients seems to guide the generation of counterfactual explanations more effectively, resulting in solutions that better simulate a return to normal sensor operation and enhance the interpretability of the counterfactual explanations. This distinction underscores the importance of method selection in the context of generation of counterfactual explanations and the potential trade-offs between computational efficiency and solution quality.

5.2 SKAB Dataset

This section extends the evaluation of the anomaly detection models and counterfactual explanations' metrics beyond the industrial data-set to include the SKAB data-set. The analysis herein follows the same evaluation metrics applied to the industrial dataset, ensuring a consistency. The following subsections will detail the anomaly detection performance and counterfactual explanation efficacy as applied to the SKAB data, providing insights into the strengths and limitations of the methods when faced with varying types of anomalies and data behaviors.

5.2.1 Anomaly Detection Evaluation Metrics

The evaluation of anomaly detection models using the SKAB dataset demonstrates the applicability of the C-AE and the C-VAE to a different domain, showcasing their capabilities to generalize across datasets with varied anomaly characteristics. The performance metrics, as presented in Table 5.7 and Table 5.8, reflect the effectiveness of each model under the distinctive anomaly conditions presented within the SKAB data.

Dataset	**F1-score**	**Recall**	**FPR**
SKAB Data - Fluid Leak Anomaly	0.55	1	1
SKAB Data - Rotor Imbalance Anomaly	0.73	0.77	0.08
SKAB Data - Slow/Sudden Increase in Water Anomaly	0.54	0.59	0.48
SKAB Data - Cavitation Anomaly	0.5	0.64	0.47
SKAB Data - High Temperature Anomaly	0.77	0.90	0.26

Table 5.7: Evaluation Metrics for the C-AE based Anomaly Detection

Dataset	**F1-score**	**Recall**	**FPR**
SKAB Data - Fluid Leak Anomaly	0.55	1	1
SKAB Data - Rotor Imbalance Anomaly	0.72	0.73	0.05
SKAB Data - Slow/Sudden Increase in Water Anomaly	0.31	0.24	0.05
SKAB Data - Cavitation Anomaly	0.38	0.33	0.15
SKAB Data - High Temperature Anomaly	0.82	0.77	0.06

Table 5.8: Evaluation Metrics for the C-VAE based Anomaly Detection

The C-AE model's performance, as depicted in Table 5.7, indicates a moderate to high capability in identifying anomalies with F1-scores ranging from 0.50 to 0.77 across different anomaly scenarios. The model shows perfect recall in cases of Fluid Leak and High Temperature anomalies, suggesting an excellent sensitivity in detecting these specific anomaly

types. However, the accompanying high False Positive Rates (FPR), particularly for Fluid Leak anomaly, suggest an over-generalization, flagging normal instances as anomalous.

In comparison, the C-VAE model, reported in Table 5.8, appears to balance the recall and precision metrics more effectively, as evidenced by the consistently lower FPRs across most anomaly scenarios while maintaining a high recall. The model's performance in the high temperature anomaly scenario is particularly noteworthy, with an F1-score of 0.85 and a recall of 0.84, coupled with a low FPR of 0.06.

When comparing the two models, the C-VAE's lower FPRs signify a more nuanced detection capability, potentially translating to a more reliable anomaly detection in practical applications. This suggests that while the C-AE model may be more aggressive in anomaly detection, the C-VAE model provides a more balanced approach, reducing the likelihood of false alarms, which is critical in maintaining operational efficiency and avoiding undue interventions. However, it is essential to note the exception in the fluid leak anomaly case, where both models exhibit a FPR of 1, indicating a universal misclassification of normal instances as anomalies. This specific instance highlights a critical vulnerability in the models' ability to discern subtleties in the data, warranting attention for improvement.

It is also apparent that the FPR is a crucial metric in evaluating the performance of anomaly detection models, especially when it comes to their practical deployment. An ideal model would not only detect all true anomalies (high recall) but also minimize the disruption caused by false alarms, a balance that seems to be better achieved by the C-VAE model in this instance.

5.2.2 Counterfactual Explanation Evaluation Metrics

5.2.2.1 Quantitative Evaluation Metrics

The quantitative assessment of counterfactual explanations for the SKAB dataset further establishes the performance consistency observed with the industrial data. The radar plots presented elucidate the capabilities of the gradient and genetic algorithm approaches in the context of this new dataset. Despite the shift from industrial sensors to a water supply system, the fundamental attributes of the generated counterfactual explanations by both methods remain in line with the findings from the industrial dataset.

	Validity ↑	Sparsity ↓	Proximity ↓
C-AE	0.8455	0.3603	0.2415
C-VAE	0.8432	0.3478	0.2821

Table 5.9: Gradient approach generated counterfactual explanations' quantitative metrics using the C-AE and C-VAE for the SKAB dataset. The up arrow indicates that a higher value is preferrable and the down arrow indicates that a lower value is preferrable.

	Validity ↑	Sparsity ↓	Proximity ↓
C-AE	0.0976	0.3619	0.4616
C-VAE	0.0983	0.3496	0.5475

Table 5.10: Genetic algorithm approach generated counterfactual explanations' quantitative metrics using the C-AE and C-VAE for the SKAB dataset. The up arrow indicates that a higher value is preferrable and the down arrow indicates that a lower value is preferrable.

For the SKAB data and its corresponding radar plot (Figure 5.9, Tables 5.9 and 5.10), the gradient approach maintains its superior validity score, demonstrating its robustness in producing desired outcomes across varying datasets and anomaly types. It remains evident that the gradient-based counterfactuals consistently align closely with the operational norm, ensuring that their alterations to the model's predictions are both significant and precise. The

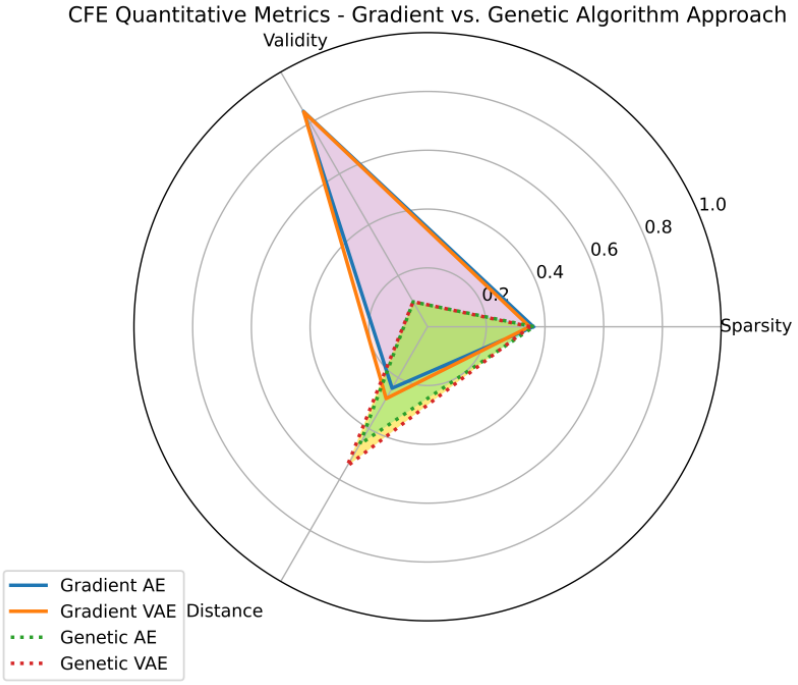


Figure 5.9: Comparative radar charts displaying counterfactual explanation (CFE) metrics—Sparsity, Validity, and Distance—for Convolutional Autoencoder (C-AE) and Variational Autoencoder (C-VAE) models using Gradient-Based and Genetic Algorithm approaches for the SKAB dataset.

genetic algorithm approach, while showing an expansive search through its higher distance metric, continues to struggle with validity. This reinforces the narrative that the genetic algorithm method, though potent in exploring diverse solutions, does not guarantee optimal results, particularly when it comes to effectively shifting the model's output towards the targeted prediction.

Comparatively, the gradient approach outperforms the genetic algorithm in the SKAB dataset, similar to its success in the industrial dataset (Figure 5.2), underscoring the gradient method's consistency and reliability. This comparative analysis emphasizes the gradient approach's potential for wider application and sets a benchmark for generation of counterfactual explanations across different domains. These observations affirm that the methodologies developed are not confined to a specific dataset but rather exhibit a degree of generalizability. The close resemblance in performance between the two distinct datasets suggests that the approaches taken are robust and could be considered reliable techniques in the generation of counterfactual explanations across various industrial domains.

5.2.2.2 Qualitative Evaluation Metrics

UMAP

The UMAP visualizations for the SKAB dataset (Table 5.11) present a complex scenario. In contrast to the quantitative metrics, the visual overlap of counterfactual and normal data points in the UMAPs does not provide a clear differentiation for either the gradient or genetic algorithm methods. Specifically, the UMAP projections suggest a blending of counterfactual explanations with the normal data, which contradicts the much lower validity scores identified through quantitative evaluation, particularly for the genetic algorithm method. Furthermore, the UMAPs fail to deliver the insights observed in the Industrial dataset, where

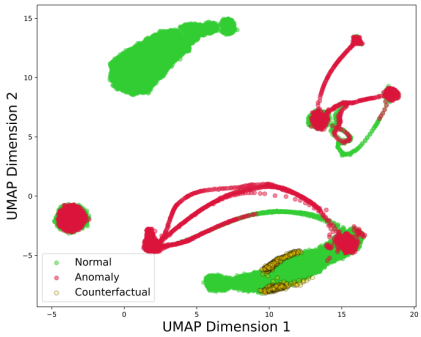
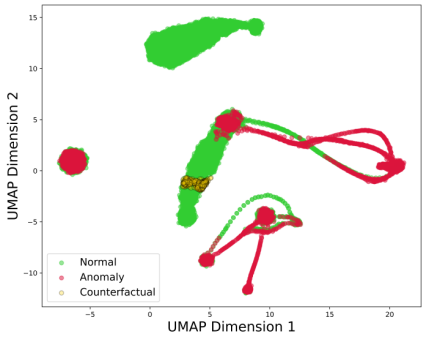
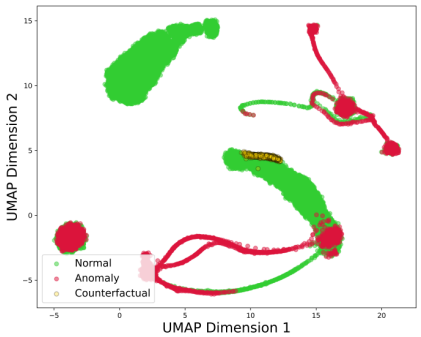
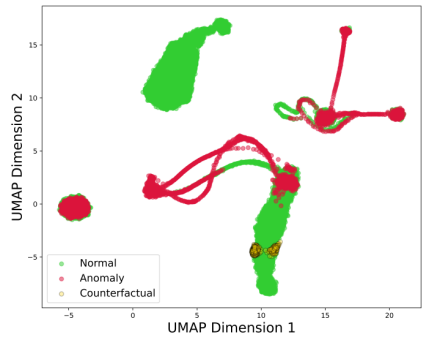
Model / Approach	C-AE	C-VAE
Gradient		
Genetic Algorithm		

Table 5.11: UMAP of the Counterfactual Explanation for C-AE and C-VAE Models generated by Gradient and Genetic Algorithm based approaches for SKAB data.

visualizations exhibited distinct separations between the normal and anomalous data that aligned with expected patterns, the SKAB dataset's UMAPs do not reveal such discernible trends.

The mismatch between the UMAP visualizations and the quantitative evaluation outcomes underlines the limitations of using UMAPs alone for assessing the quality of counterfactual explanations. This divergence highlights the critical need for additional qualitative assessments to validate counterfactual efficacy. To address the gap left by UMAPs, other qualitative methods, like data drift detection, could be employed. These alternative approaches may provide a more comprehensive understanding of how well counterfactuals can realign anomalous data points with the normal data distribution, offering a more rounded evaluation of their effectiveness.

Data Drift Detection

Gradient Based Counterfactual Explanations - C-AE and C-VAE

In evaluating counterfactual explanations, it is critical to consider data drift detection, particularly when other qualitative metrics like UMAPs provide inconsistent insights. The images provided illustrate the data drift phenomenon. In the Figure 5.10 (left), the distribution shifts from the normal state (prior) to the anomaly state (post) are evident, indicating a disruption from the established patterns. Conversely, the Figure 5.10 (right) portrays the generated counterfactual explanations by gradient based approach using the C-AE model and its

efforts to correct this drift, bringing the data distribution closer to the normal state as seen in the training data.

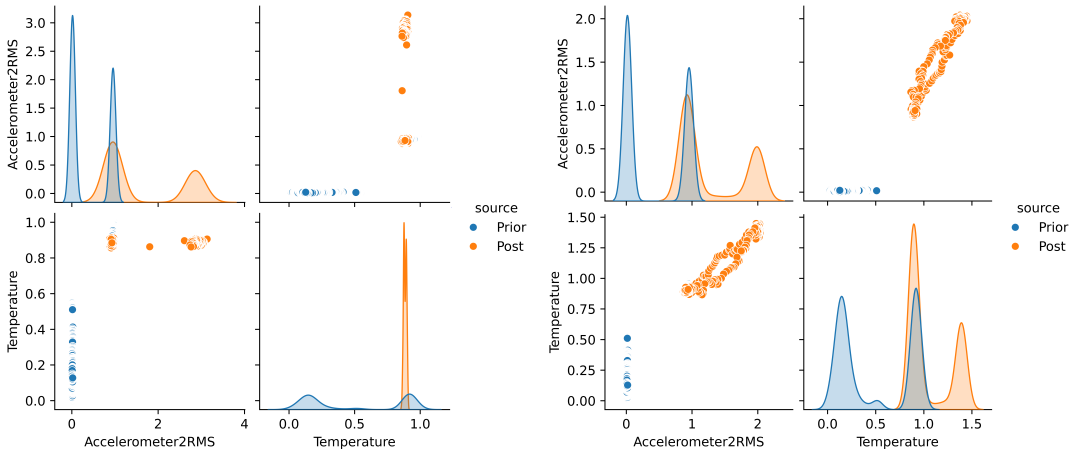


Figure 5.10: The image compares data drift in 'Accelerometer' and 'Temperature' between training and anomalous data (left) versus training and counterfactual data (right) generated by Gradient method using the C-AE for the SKAB dataset.

The visuals for the gradient approach using the C-VAE (Figure 5.11) exhibit similar trends, with the counterfactual feature distributions showing movement towards the baseline normal state. The overlap between post-counterfactual and normal distributions is more pronounced, indicating a better correction of the drift caused by the anomalies.

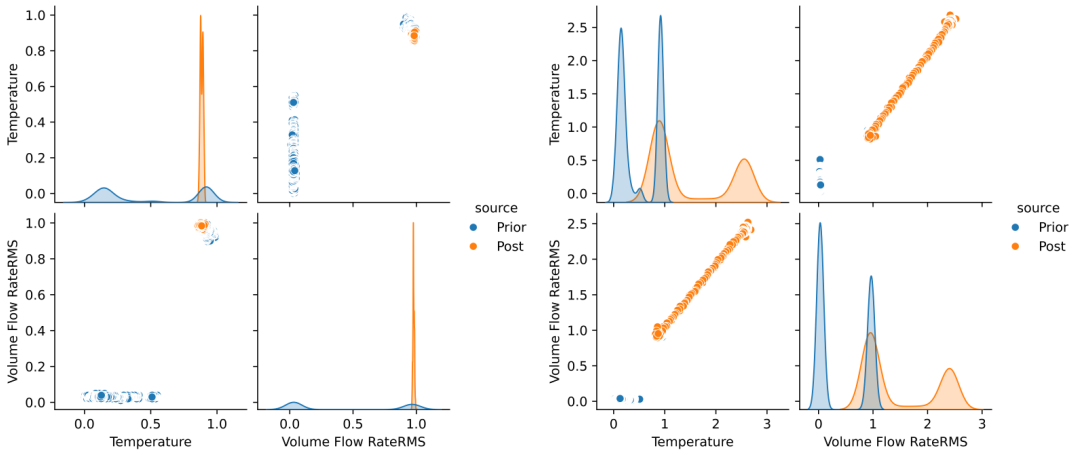


Figure 5.11: The image compares data drift in 'Temperature' and 'Volume Flow Rate' between training and anomalous data (left) versus training and counterfactual data (right) generated by Gradient method using the C-VAE for the SKAB dataset.

These observations reveal that while UMAPs provide a broad overview, data drift detection offers a more detailed evaluation of the counterfactuals' effectiveness, particularly in their ability to restore features to a state resembling normal operating conditions. It is important to note, however, that despite some improvement, a complete reversal of data drift is not always achieved.

This limitation is compounded by the relatively high False Positive Rate (FPR) observed in the models when applied to the SKAB dataset compared to the industrial data. This diminished model performance negatively impacts the downstream generation of counterfactual

explanations. The reliance on the LOF to model reconstruction errors and identify anomalous features may further amplify the effects of the models' compromised detection ability. As the generation of counterfactual explanations leverages the same flawed model gradients to guide the search through the loss landscape, the challenges in transitioning from an anomalous instance to a counterfactual (normal instance) are intensified.

The compounded nature of these issues originates from the initial detection stage, where a high FPR implies a foundational weakness in distinguishing the truly anomalous features. Subsequent steps, including the utilization of the LOF for feature selection and the generation of counterfactual explanations that are supposed to correct for detected anomalies, are hence built on unstable grounds. The propagation of errors through the framework can lead to counterfactual explanations that, while they may shift data points in the direction of normalcy, fail to convincingly simulate the conditions of a typical operational state due to their reliance on an imprecise mapping of anomalies. This cascade of compounded errors from initial detection to counterfactual explanations underlines the necessity of ensuring robust initial model performance. It underscores the importance of rigorous initial anomaly detection as a precursor to the effective generation of counterfactual explanations, without which the entire explanatory process may be undermined.

Furthermore, it's important to note that the features depicted in these visualizations are those identified by the LOF-based feature selector as significant contributors to the anomalies. The correction of drift in these features is particularly meaningful, as it implies that the counterfactuals are not arbitrary but targeted towards the very attributes that were most impacted by the anomalies. This targeted approach can be valuable for practitioners seeking to understand the underlying factors of detected anomalies and to devise appropriate corrective measures.

Genetic Algorithm Based Counterfactual Explanations - C-AE and C-VAE

For generated counterfactual explanations by the genetic algorithm using the C-AE model (Figure 5.12), the data drift visualization indicates an inability to realign the post-counterfactual distribution closely with the normal operational state. The drift remains pronounced, suggesting the genetic approach's counterfactuals for the C-AE model are comparatively less effective in mitigating the anomalies' impact on data distribution.

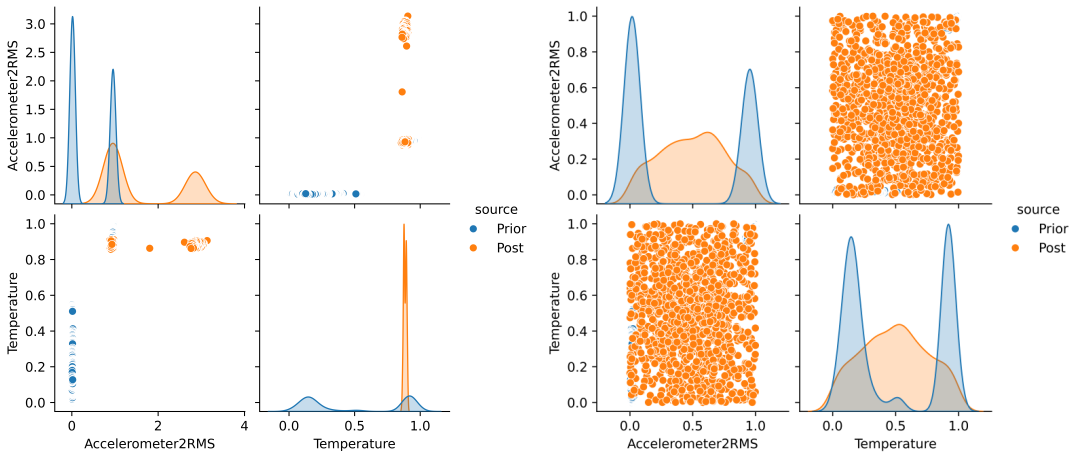


Figure 5.12: The image compares data drift in 'Accelerometer' and 'Temperature' between training and anomalous data (left) versus training and counterfactual data (right) generated by Genetic Algorithm method using the C-AE for the SKAB dataset.

When comparing the effectiveness of the genetic algorithm approach to generating counterfactual explanations using the C-VAE model (Figure 5.13), the results are similarly un-

derwhelming. The approach struggles to correct the drift caused by anomalies, with the post-counterfactual distribution failing to converge satisfactorily with the normal data. The counterfactuals do not appear to offer a significant improvement over those generated by the C-AE model, underlining the genetic algorithm's limitations in generating robust counterfactual explanations for both models in the context of the SKAB dataset.

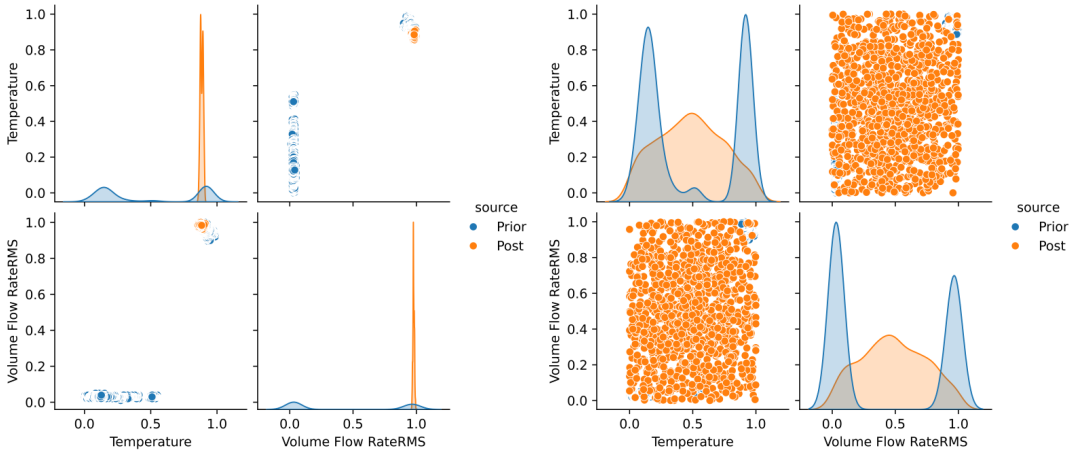


Figure 5.13: The image compares data drift in 'Temperature' and 'Volume Flow Rate' between training and anomalous data (left) versus training and counterfactual data (right) generated by Genetic Algorithm method using the C-VAE for the SKAB dataset.

The data drift detection based qualitative evaluation offers a nuanced perspective on counterfactual explanations that UMAP analysis does not, particularly for the genetic algorithm approach. While UMAPs might superficially indicate a successful clustering of counterfactual instances with normal data, data drift detection reveals a different story, aligning with the quantitative metrics which have already pointed to the superior performance of the gradient approach. The genetic algorithm based counterfactual explanations, both for the C-AE and C-VAE models, fail to effectively realign the data distribution post-counterfactual with the normal state. This discrepancy is captured through data drift detection, which visualizes the persistent distributional differences, thereby confirming the genetic algorithm's lower quantitative validity scores. It emphasizes the importance of using a combination of evaluation methods to gain a comprehensive understanding of the method's performance. In this case, data drift detection serves as a more reliable method for corroborating quantitative assessments and revealing the limitations of the implemented methods that UMAPs may miss, thus offering a more complete validation of the counterfactual explanation methods applied to the SKAB dataset.

5.3 Discussion On Method

5.3.1 Limitations Of The Anomaly Detection Models

In reflecting on the models applied for anomaly detection and in adopting a self-critical perspective to acknowledge the limitations and potential impacts on the results. One notable limitation encountered with the current approach involves the handling of binary and categorical features. The reconstruction of such features has presented significant challenges, where the models used have shown sub-optimal performance. This is partially due to the inherent characteristics of binary and categorical data, which do not align well with methods optimized for continuous variables. As these features are essential in many practical applications, their inadequate reconstruction could lead to misleading interpretations of anomalies and affect the trustworthiness of the predictive models.

Additionally, the window size selected for the analysis has a profound effect on the model's ability to detect anomalies. A window that is too small may not capture sufficient context, leading to higher false positives or false negatives. Conversely, a window that is too large might dilute the anomalous signals, rendering them indistinguishable from normal variations in the data. The challenge lies in finding an optimal balance that accurately reflects the temporal dynamics of the dataset without compromising the detection sensitivity.

5.3.2 The Effect Of The Feature Selector

The Feature Selector stands as a critical component within the anomaly detection and counterfactual explanations framework, as delineated by the flowchart of the framework (Figure 4.1). It plays a pivotal role in streamlining the feature space, which directly influences the interpretability and efficacy of the subsequent counterfactual explanations.

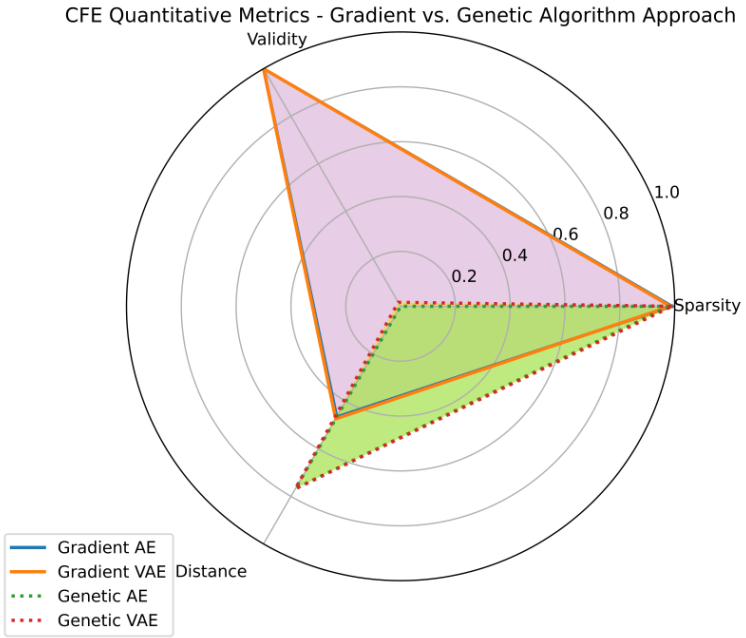


Figure 5.14: Comparative radar charts displaying counterfactual explanation (CFE) metrics—Sparsity, Validity, and Distance—for Convolutional Autoencoder (C-AE) and Variational Autoencoder (C-VAE) models using Gradient-Based and Genetic Algorithm approaches.

	Validity ↑	Sparsity ↓	Proximity ↓
C-AE	1	0.985	0.4737
C-VAE	1	0.99	0.4660

Table 5.12: The table visualizes effect on the quantitative metrics of the generated counterfactual explanations using the gradient approach and when no feature selector is used.

	Validity ↑	Sparsity ↓	Proximity ↓
C-AE	0	0.98	0.7672
C-VAE	0.0169	0.99	0.7686

Table 5.13: The table visualizes effect on the quantitative metrics of the generated counterfactual explanations using the genetic algorithm approach and when no feature selector is used.

The ablation study conducted to assess the Feature Selector's impact on counterfactual explanations provides insightful revelations. As shown in the radar plot of the quantitative metrics (Figure 5.2, Tables 5.12 and 5.13), the utilization of the Feature Selector contributes to a reasonable profile across validity and sparsity metrics. This implies that the feature selector effectively identifies and isolates the most relevant features contributing to the anomalies, which in turn allows for the generation of counterfactual explanations that are precise, minimally divergent, and exhibit essential sparsity—qualities that enhance their interpretability and actionability.

In contrast, the quantitative metrics without the feature selector (Figure 5.14), exhibits an unreasonable profile with respect to the sparsity metric for the gradient based approach, that is, while it achieves a similar validity score, the generated counterfactual explanations are needlessly more complex. Additionally, another interesting observation is that the performance of the genetic algorithm based approach is much worse with a validity that is almost zero. This highlights the pivotal role of the feature selector in fine-tuning the counterfactual generation process. Without it, the explanations may become unnecessarily complex, less sparse, and potentially less valid, which can obfuscate the interpretative process for end-users.

The evidence from the ablation study underscores the feature selector's significance in producing high-quality counterfactuals. It acts as a funnel that distills the anomaly detection results into a more manageable and interpretable form, enabling the counterfactual explanations methods to operate with a focused and relevant feature set and this, in turn, aligns with the goal of achieving high model interpretability.

5.3.2.1 Limitations Of The Feature Selector

The LOF based feature selector's effectiveness in crafting sparse counterfactuals is evaluated critically within the context of Anomaly Type 1, involving deviations in *Sensor 5*. Despite its potential, the following experiments, which incorporated models trained under varied initial conditions, highlighted a significant limitation in the reliability of the feature selector. Different initialization seeds produced a lack of uniformity in feature selection, signaling an underlying unreliability in the method's capacity to consistently identify features.

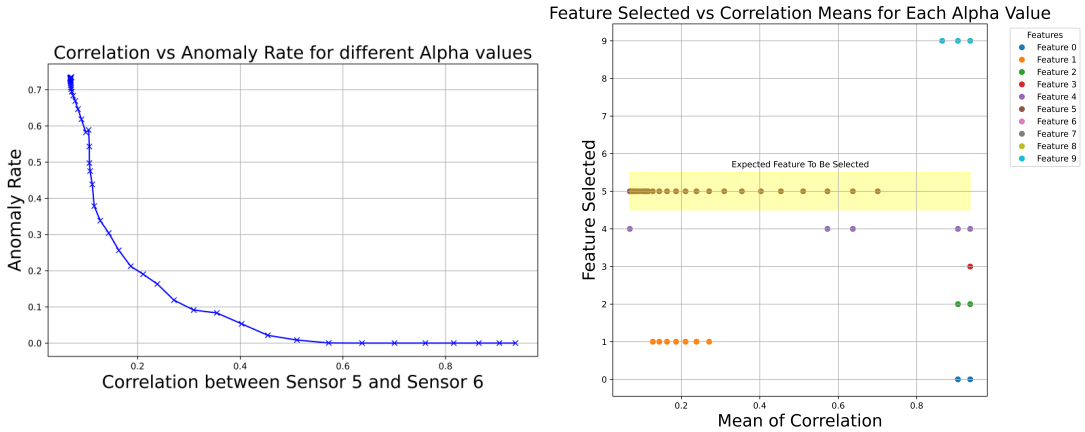


Figure 5.15: The right image is a scatter plot depiction of the feature selection across different alpha (smoothing parameter) values, with the highlighted band indicating the expected feature to be selected based on domain knowledge and the left image is a depiction of correlation loss versus the anomaly rate.

Figure 5.15 (right) illustrates the effect of varying anomaly rates and correlation between sensors in the context of Anomaly Type 1, on the feature selection process. Smoothing, in this context, refers to the application of exponential weighted moving averages to recent obser-

vations. By adjusting the smoothing parameter (alpha), the influence of recent data points on the overall trend can be controlled. The figure shows how varying the smoothing parameter that is, inducing anomalies, alters the selected features, demonstrating the sensitivity of feature selection to different levels of anomaly rates. By smoothing the sensor readings, anomalies are induced, which helps in testing the robustness of the feature selection process.

The highlighted band indicates the expected feature that should be selected based on domain-specific knowledge about *Sensor 5*. This band represents the desired outcome, yet the scatter plot reveals the selection of features both within and outside of this expected range, illustrating the effect of prevalent levels of anomalies to the feature being selected, that is, as the correlation increases and the anomaly rate decreases (Figure 5.15 (left)), the feature selector selects features that may not actually be anomalous and vice-versa.

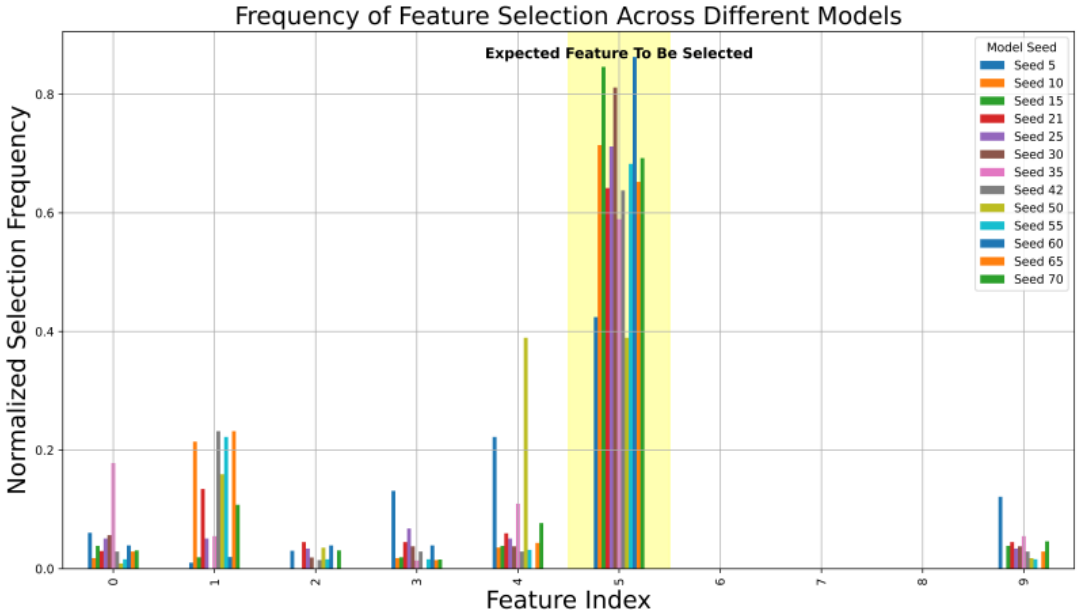


Figure 5.16: Bar chart showing normalized frequency of feature selection variability across models initialized with different seeds, highlighting the expected feature within the yellow band.

Further experimentation revealed that the feature selector's determinations are highly susceptible to the initialization state of the model. This susceptibility raises concerns regarding the method's ability to reliably identify pertinent anomalous features in diverse training scenarios. First, synthetic anomalies are introduced via a smoothing parameter and as this parameter is modulated to affect the synthetic correlation, a corresponding impact on the anomaly rates and the feature selector's performance can be noted. The feature selection showed significant variability as the synthetic correlation increased, indicating a reduction in anomaly rates. This variability underlines the dependency of the method on both the data characteristics and the model parameters.

Figure 5.16 further illustrates the inconsistency of feature selector with respect to model initialization, presenting the normalized frequency of feature selection across models initialized with different seeds. The expected feature is marked within a yellow band, and the bars represent the selection frequency of each feature. The variability is evident as not all models consistently select the expected feature as seen in the varying levels of frequency of the expected feature to be selected, corroborating the selector's instability.

Such variability poses a direct challenge to the replicability and reliability of the method. A feature selector yielding disparate results under various initializations could lead to fluc-

tuations in the counterfactual explanations' quality, thereby affecting their credibility and the actionable insights they can provide.

Given these findings, the study highlights the need for a more stable feature selection mechanism, one that can withstand changes in model initialization and data characteristics. This would ensure that key features are reliably and consistently earmarked for modification in the counterfactual generation process.

5.4 The Work In A Wider Context

This thesis explores the generation of counterfactual explanations within the realm of neural network-based anomaly detection, a field that is pivotal for enhancing transparency and accountability in automated systems. The integration of counterfactual explanations addresses significant ethical concerns associated with AI and ML, particularly in terms of explainability, fairness, and potential biases in automated decision-making processes.

The adoption of counterfactual explanations in neural networks aids in demystifying the decision-making processes of AI systems, promoting a greater understanding among users. By providing insights into how certain outcomes are derived, these explanations empower users to contest and correct decisions, thereby adhering to ethical standards of fairness and accountability by aligning with regulatory frameworks like the European Union's General Data Protection Regulation (GDPR), which advocates for the "right to explanation". This is particularly crucial in sectors where AI decisions may have significant repercussions, such as healthcare, finance, and criminal justice. On a societal level, enhancing the transparency of AI systems via counterfactual explanations could lead to broader acceptance and trust in emerging technologies. This transparency not only facilitates user trust but also encourages broader societal engagement with AI technologies, promoting informed discussions about their role and impact in society. In conclusion, the integration of counterfactual explanations into neural network-based anomaly detection represents a significant step towards ethical AI practices. It not only enhances the interpretability and fairness of automated systems but also aligns with broader educational and policy-driven goals aimed at fostering an ethically conscious digital society.

6 Conclusion

This thesis aimed to generate counterfactual explanations in the domain of anomaly detection for multivariate time-series data by first focusing on the development and evaluation of C-AE and C-VAE models. Next, these models were tasked with not only identifying anomalies but also providing interpretable counterfactual explanations that offer actionable insights. The research objectives were encapsulated through several research questions, which sought to understand the role of counterfactual explanations in interpretability, compare methods for generating these explanations, and develop robust evaluation metrics.

6.1 Achievement of Thesis Aims and Research Questions

1. **Interpretation of Deep Anomaly Detection Models**: The thesis successfully demonstrated that counterfactual explanations significantly enhance the interpretability of deep learning models for anomaly detection. These explanations provide a clear, actionable pathway for technicians and operators, aligning with the need for explainability in predictive maintenance applications. Counterfactual explanations in deep anomaly detection models offer actionable insights by clearly illustrating the minimal adjustments needed in sensor outputs to prevent detected anomalies, as depicted in the Figure 5.1 where original high anomalous readings are contrasted with their adjusted counterfactual values to achieve normal classification.

2. **Comparison of Explanation Methods**: Between the optimization (gradient-based) and heuristic (genetic algorithm-based) methods for generating counterfactual explanations, the gradient-based approach consistently outperformed in terms of producing explanations with higher validity, closer proximity to original data points, and maintaining simplicity in alterations.

3. **Development of Evaluation Metrics**: A comprehensive suite of quantitative and qualitative metrics was established, facilitating a thorough assessment of counterfactual explanations. These metrics ensured that the explanations were not only statistically sound but also aligned closely with practical, operational needs.

6.2 Impact and Implications for Target Audience

For practitioners and researchers in the field of predictive maintenance, this work provides robust methodologies for enhancing the transparency and accountability of deep learning models. By integrating counterfactual explanations, the models not only predict anomalies but also suggest preventative measures, potentially reducing downtime and improving maintenance strategies.

6.3 Future Work

The trajectory for future work is primarily influenced by the need to overcome limitations identified during the research. The forthcoming efforts should focus on the following areas:

- **Enhanced Feature Reconstruction**: To address the challenges with binary and categorical features, future research should be directed towards developing more sophisticated encoding and reconstruction techniques that can seamlessly integrate these data types within the anomaly detection models.
- **Stabilization of Feature Selection**: Given the variability in feature selection due to different initialization seeds, future work should explore more stable feature selection mechanisms that are less sensitive to model initialization and provide consistent identification of anomalous features.
- **Cascading Effects of Compromised Model Performance**: Addressing the cascading effect stemming from compromised initial model performance, which significantly impacts the generation of counterfactual explanations. This will involve revisiting the foundational stages of anomaly detection to ensure the integrity of subsequent counterfactual generation.
- **Investigation into Genetic Algorithm Performance**: Conducting in-depth research to understand the underlying reasons for the suboptimal performance of the genetic algorithm based approach in generating counterfactual explanations. The goal is to identify potential areas for algorithmic enhancements to improve its efficiency and effectiveness in producing high-quality counterfactual explanations for multi-variate time series data.

7 Ethical Considerations

The methodologies and frameworks developed and applied in this thesis for anomaly detection and counterfactual explanations have been designed with a strong emphasis on ethical considerations. The data used in this research is anonymized, ensuring compliance with GDPR regulations, and no personal data is recorded or transmitted. The data collection methods do not harm the environment or living beings, adhering to relevant safety standards. The research is conducted with strict adherence to ethical practices, ensuring transparency, fairness, and accountability. Counterfactual explanations provide clear insights into the decision-making processes of neural network models, enhancing trust and responsible use of AI in real-world applications. In conclusion, every aspect of this thesis is developed with a commitment to ethical considerations, including data privacy, environmental safety, and ethical research practices.

Bibliography

[1] *What is Predictive Maintenance?* `https://www.ibm.com/topics/predictive-maintenance`. Accessed: 03/2024.

[2] Frank E. Grubbs. "Procedures for Detecting Outlying Observations in Samples". In: *Technometrics* (1969).

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM Comput. Surv.* (2009).

[4] Victoria Hodge and Jim Austin. "A Survey of Outlier Detection Methodologies". In: (2004).

[5] Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu C. Aggarwal, and Mahsa Salehi. *Deep Learning for Time Series Anomaly Detection: A Survey*. 2022.

[6] Sandra Wachter, Brent Mittelstadt, and Chris Russell. *Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR*. 2018.

[7] Christoph Molnar. *Interpretable Machine Learning*. 2022.

[8] Iurii D. Katser and Vyacheslav O. Kozitsin. *Skoltech Anomaly Benchmark (SKAB)*. 2020.

[9] Francis Ysidro Edgeworth. "XLI. On discordant observations". In: *Philosophical Magazine Series 1* (1887).

[10] Charu C. Aggarwal. *Outlier Analysis*. 2016.

[11] Douglas M Hawkins. *Identification of Outliers*. Springer, 1980.

[12] Pierre Baldi. "Autoencoders, Unsupervised Learning, and Deep Architectures". In: 2012.

[13] Geoffrey E Hinton and Ruslan R Salakhutdinov. "Reducing the Dimensionality of Data With Neural Networks". In: *science* (2006).

[14] Umberto Michelucci. *An Introduction to Autoencoders*. 2022.

[15] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017.

[16] Jinwon An and Sungzoon Cho. "Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability". In: *Special lecture on IE* (2015).

[17] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. "Definitions, Methods, and Applications in Interpretable Machine Learning". In: *Proceedings of the National Academy of Sciences* (2019).

[18] DARPA. *DARPA - Explainable Artificial Intelligence (XAI) Program*. https://www.darpa.mil/program/explainable-artificial-intelligence. Accessed: 03/2024. 2017.

[19] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. "A Survey of Methods for Explaining Black Box Models". In: (2018).

[20] Amina Adadi and Mohammed Berrada. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* (2018).

[21] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. "Machine Learning Interpretability: A Survey on Methods and Metrics". In: *Electronics* (2019).

[22] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016.

[23] Raed Alharbi, Minh N. Vu, and My T. Thai. *Learning Interpretation with Explainable Knowledge Distillation*. 2021.

[24] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* (2015). DOI: 10.1371/journal.pone.0130140.

[25] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: 2017.

[26] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. "Explaining Machine Learning Classifiers Through Diverse Counterfactual Explanations". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020.

[27] Ziqi Zhao, Yucheng Shi, Shushan Wu, Fan Yang, Wenzhan Song, and Ninghao Liu. *Interpretation of Time-Series Deep Models: A Survey*. 2023.

[28] Yotam Hechtlinger. *Interpretation of Prediction Models Using the Input Gradient*. 2016.

[29] J. Blank and K. Deb. "pymoo: Multi-Objective Optimization in Python". In: *IEEE Access* (2020).

[30] *What Is the Genetic Algorithm?* https://se.mathworks.com/help/gads/what-is-the-genetic-algorithm.html. Accessed: 03/2024.

[31] Kalyanmoy Deb and Debayan Deb. "Analysing mutation schemes for real-parameter genetic algorithms". In: *International Journal of Artificial Intelligence and Soft Computing* (2014).

[32] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020.

[33] Nesryne Mejri, Laura Lopez-Fuentes, Kankana Roy, Pavel Chernakov, Enjie Ghorbel, and Djamila Aouada. *Unsupervised Anomaly Detection in Time-series: An Extensive Evaluation and Analysis of State-of-the-art Methods*. 2023.

[34] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. "A Deep Learning Approach for Network Intrusion Detection System". In: 2016.

[35] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. "1D convolutional neural networks and applications: A survey". In: *Mechanical Systems and Signal Processing* (2021).

[36] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. "Autoencoder-based Network Anomaly Detection". In: 2018.

[37] Abbas Golestani and Robin Gras. "Can We Predict The Unpredictable?" In: *Scientific reports* (2014).

[38] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts. "Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model". In: 2020.

[39] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. *A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data*. 2018.

[40] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. "Beatgan: Anomalous Rhythm Detection Using Adversarially Generated Time Series." In: *IJCAI*. 2019.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need". In: *Advances in neural information processing systems* (2017).

[42] Longyuan Li, Junchi Yan, Qingsong Wen, Yaohui Jin, and Xiaokang Yang. "Learning Robust Deep State Space for Unsupervised Anomaly Detection in Contaminated Time-series". In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[43] Xixuan Wang, Dechang Pi, Xiangyan Zhang, Hao Liu, and Chang Guo. "Variational Transformer-based Anomaly Detection Approach for Multivariate Time Series". In: *Measurement* (2022).

[44] Huan Song, Deepta Rajan, Jayaraman Thiagarajan, and Andreas Spanias. "Attend and Diagnose: Clinical Time Series Analysis using Attention Models". In: *Proceedings of the AAAI conference on artificial intelligence*. 2018.

[45] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. "Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges". In: *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*. 2019.

[46] A. Carlisle Scott, William J. Clancey, Randall Davis, and Edward H. Shortliffe. "Explanation Capabilities of Production-Based Consultation Systems". In: *American Journal of Computational Linguistics* (1977).

[47] Bryce Goodman and Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *AI magazine* (2017).

[48] State Council Chinese Government. *Development Plan for New Generation Artificial Intelligence*. `http://www.gov.cn/zhengce/content/2017-07/20/content5211996.htm`. Accessed: 03/2024. 2017.

[49] Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz-Rodríguez. "Explainable Artificial Intelligence (XAI) on Timeseries Data: A Survey". In: *arXiv preprint arXiv:2104.00950* (2021).

[50] Riccardo Guidotti. "Counterfactual explanations and how to find them: literature review and benchmarking". In: *Data Mining and Knowledge Discovery* (2022).

[51] Judea Pearl. "Causal Inference in Statistics: An Overview". In: *Statistics Surveys* (2009).

[52] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. "Explanations Based on the Missing: Towards Contrastive Explanations With Pertinent Negatives". In: *Advances in neural information processing systems* (2018).

[53] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM journal on imaging sciences* (2009).

[54] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. "Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models". In: *arXiv preprint arXiv:2101.00288* (2021).

[55] Deborah Sulem, Michele Donini, Muhammad Bilal Zafar, Francois-Xavier Aubet, Jan Gasthaus, Tim Januschowski, Sanjiv Das, Krishnaram Kenthapadi, and Cedric Archambeau. *Diverse Counterfactual Explanations for Anomaly Detection in Time Series*. 2022.

[56] Swastik Haldar, Philips George John, and Diptikalyan Saha. "Reliable Counterfactual Explanations for Autoencoder based Anomalies". In: *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*. 2021.

[57] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. "Multi-objective Counterfactual Explanations". In: *International Conference on Parallel Problem Solving from Nature*. 2020.

[58] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. "CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. AIES '20. 2020.

[59] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. "The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. 2019.

[60] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. "FACE: Feasible and Actionable Counterfactual Explanations". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. AIES '20. 2020.

[61] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E. Hines, John P. Dickerson, and Chirag Shah. *Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review*. 2022.

[62] Chan Sik Han and Keon Myung Lee. "Gradient-based Counterfactual Generation for Sparse and Diverse Counterfactual Explanations". In: *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. 2023.

[63] André Artelt and Barbara Hammer. *Convex Density Constraints for Computing Plausible Counterfactual Explanations*. 2020.

[64] Jana Lang, Martin Giese, Winfried Ilg, and Sebastian Otte. *Generating Sparse Counterfactual Explanations For Multivariate Time Series*. 2022.