FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

XTadGAN: Generative Adversarial Networks to Detect Extremely Rare Anomalies

Nuno Beleza Vasconcelos

Mestrado em Engenharia e Ciência de Dados

Supervisor: Carlos Soares, PhD

Supervisor: Vítor Cerqueira, PhD

October 11, 2023

XTadGAN:
Generative Adversarial Networks to Detect Extremely Rare Anomalies

Nuno Beleza Vasconcelos

Mestrado em Engenharia e Ciência de Dados

Approved in oral examination by the committee:

Chair: Prof. Doctor António Pedro Rodrigues Aguiar, University of Porto
External Examiner: Prof. Doctor Albert Bifet, University of Waikato
Supervisor: Prof. Doctor Carlos Milheiro de Oliveira Pinto Soares, University of Porto

October 11, 2023

Abstract

Machine learning methods have been widely employed for anomaly detection in time series data, but often struggle to identify rare anomalies in high-dimensional or non-stationary data. Generative Adversarial Networks (GANs) have shown promise in addressing this limitation, but their effectiveness in detecting extremely rare anomalies remains a challenge. Additionally, the lack of systematic comparison methods for evaluating anomaly detection algorithms, particularly in relation to varying anomaly frequencies, has hindered progress in this field.

This thesis addresses these challenges by introducing novel contributions to the realm of anomaly detection in time series data. Firstly, two new GAN-based architectures, TadGAN-DT and XTadGAN, are proposed to handle scenarios with extremely rare anomalies. The former, TadGAN-DT, incorporates non-parametric dynamic thresholding and pruning methods. The latter, XTadGAN, leverages meta-information on expected anomaly frequencies to establish rarity-based dynamic thresholding and pruning strategies. Our experimental results demonstrate that both algorithms outperform other relevant approaches in rare anomaly detection.

Furthermore, a comprehensive framework for evaluating anomaly detection models is introduced. This framework uses Monte Carlo sampling to generate an arbitrary number of time series from a small set of original datasets, simulating various controlled scenarios. It enables systematic assessments across various time series attributes, specifically considering varying levels of anomaly rarity. This establishes a standardized test bench, facilitating a deeper understanding of model strengths and limitations. To enhance model comparisons, a novel sensitivity index, the *x-score*, is introduced. This metric provides an objective measure to evaluate the performance of different anomaly detection algorithms across a spectrum of attributes, particularly varying anomaly frequencies.

This research contributes to the field of time series anomaly detection by advancing the understanding of rare anomaly detection using GANs. It introduces a robust framework for systematic model evaluation, including a sensitivity index that enhances the reliability of model comparisons, guiding future research and improving the applicability of anomaly detection algorithms in real-world scenarios.

Keywords: anomaly detection, time series, generative adversarial networks (GANs), TadGAN, XTadGAN, rare anomalies, sensitivity analysis, evaluation framework, model comparison, Monte Carlo sampling, rarity-based thresholding, dynamic thresholding

Resumo

Métodos de *machine learning* têm sido amplamente utilizados para a deteção de anomalias em séries temporais, mas apresentam dificuldade em identificar anomalias raras em dados de alta-dimensionalidade ou não-estacionários. As Redes Generativas Adversariais (GANs) mostram-se uma alternativa promissora para ultrapassar essa limitação, mas a sua eficácia na deteção de anomalias extremamente raras ainda é um desafio. Adicionalmente, a falta de métodos sistemáticos de comparação para avaliar algoritmos de deteção de anomalias, particularmente em relação a diferentes frequências de anomalias, tem prejudicado o progresso neste campo.

Esta dissertação aborda esses desafios ao introduzir novas contribuições no domínio da deteção de anomalias em séries temporais. Em primeiro lugar, são propostas duas novas arquiteturas baseadas em GANs, TadGAN-DT e XTadGAN, para lidar com cenários de anomalias extremamente raras. A primeira, TadGAN-DT, incorpora métodos de *thresholding* dinâmico não-paramétrico. A segunda, XTadGAN, explora a utilização de meta-informações sobre a frequência esperada de anomalias para condicionar o *thresholding* e criar um método de *pruning* contextual. Os nossos resultados experimentais demonstram que ambos os algoritmos superam outras abordagens relevantes na deteção de anomalias raras.

Adicionalmente, é introduzido um novo *framework* para avaliar modelos de deteção de anomalias. Esta abordagem utiliza o método de amostragem de Monte Carlo para gerar um número arbitrário de séries temporais a partir de um conjunto limitado de séries originais, permitindo a simulação de vários cenários controlados. Este método permite a avaliação sistemática de várias características de séries temporais, em particular o nível de raridade de anomalias que a compõem. A sua utilização assenta na criação de um banco de testes padronizado, permitindo uma compreensão mais profunda das vantagens e limitações de cada algoritmo. De forma a melhorar as comparações entre modelos, é ainda introduzido um novo índice de sensibilidade, o *x-score*. Esta métrica fornece uma medida objetiva para avaliar o desempenho de diferentes algoritmos de deteção de anomalias considerando um espectro de características relevantes, com especial destaque para diferentes frequências de anomalias.

Este estudo apresenta novas contribuições para o campo da deteção de anomalias em séries temporais. São apresentados avanços na deteção de anomalias raras usando GANs e introduzida uma metodologia robusta para a avaliação sistemática de algoritmos em condições controladas. O índice de sensibilidade desenvolvido melhora a confiabilidade das comparações entre modelos, orientando futuras pesquisas e melhorando a aplicabilidade de algoritmos de deteção de anomalias em cenários do mundo real.

Keywords: deteção de anomalias, séries temporais, redes adversariais generativas (GANs), TadGAN, XTadGAN, anomalias raras, análise de sensibilidade, métodos de avaliação, comparação de modelos, amostragem de Monte Carlo, thresholding condicionado, thresholding dinâmico

Acknowledgements

The completion of this work represents the culmination of a valuable and fulfilling journey, and I am deeply grateful to the many individuals who have supported and guided me along the way.

I am profoundly thankful to my advisors, Carlos Soares and Vítor Cerqueira, whose expertise and mentorship have been instrumental in shaping the trajectory of this research. Their guidance and valuable insights challenged me to think critically and greatly enhanced the quality of this work.

I am indebted to my teachers, colleagues and peers, who provided a stimulating intellectual environment for discussions, brainstorming sessions, and collaborations. I must make special mention of André Afonso, Gabriel Carvalhal and Wagner Ceulin, with whom I have forged deep friendships. Your camaraderie has been a source of inspiration and growth.

Alberto Bastos and Tiago Sanchez, of Kaizen Institute Western Europe, played a pivotal role in enabling me to conduct this journey. Their investment and trust gave greater significance to my career.

To my family, who have given me their shoulders to stand upon, providing me with the tools and support that have enabled me to reach this point. To my girlfriend, for the unwavering encouragement and putting up with me and my sleepless nights.

My gratitude also extends to all unsung heroes, individual and collective, who generously volunteered their time, resources, and data for the experiments conducted in this thesis. A mention is due to K. Hundman (NASA), Numenta, and the custodians of the UCR archive, whose publicly available data provided essential building blocks for this research. This recognition is also extended to LIACC, embodied by Gil Rocha and Miguel Abreu, without whom this research would have been cut short. A word of appreciation goes to Sarah Alnegheimish from MIT for the remarkable resource that is Orion and for engaging in invaluable discussions on the project repository.

Finally, I would like to express my heartfelt appreciation to Rui Moreira, whose casual introduction to the realm of artificial intelligence five years ago ignited my passion and set me on this journey.

Thank you all for your contributions and support.

Nuno

Contents

List of Figures

List of Tables

Abbreviations

AE	Auto-Encoder
ARIMA	Autoregressive Integrated Moving Average
COF	Connectivity-based Outlier Factor
DT	Dynamic Thresholding
DTW	Dynamic Time Warping
EWMA	Exponentially-Weighted Moving Average
FDA	Functional Data Analysis
FIT	Failure In Time
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
HTM	Hierarchical Temporal Memory
KNN	K-Nearest Neighbors
LOF	Local Outlier Factor
LSTM	Long Short Term Memory Recurrent Neural Networks
MTBF	Mean Time Between Failures
PCA	Principal Component Analysis
ROC	Receiver Operating Curve
TadGAN	Time Series Anomaly Detection using Generative Adversarial Networks
TN	True Negative
TP	True Positive
VAE	Variational Auto-Encoder

Chapter 1

Introduction

Time series data are often used to monitor and analyze complex systems, such as financial markets, biological systems, and industrial processes. In many cases, it is of paramount importance to be able to identify anomalous data points or patterns in these time series, as these anomalies may indicate errors or unusual events that are worth investigating further. This ability can aid in identifying fraudulent activities, detecting system failures, and monitoring the health of complex systems.

1.1 Motivation

Traditional methods for anomaly detection in time series data, such as statistical tests and machine learning algorithms, have been widely used for this purpose. However, these methods can be limited in their ability to detect complex or rare anomalies, particularly in high-dimensional or non-stationary time series data [Chandola et al., 2009].

Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] are a class of machine learning algorithms that have been shown to be effective for learning the underlying distributions of data and generating synthetic data samples that are similar to the real data. Although GANs have been primarily used in the context of image and video datasets to generate new samples, recently there has been some research in using them for time series data as well. One notable example is the TadGAN paper by Geiger et al. [2020], which has shown that GANs can be effective in detecting anomalies in time series data when compared to traditional methods.

The results of this research have demonstrated that adversarial training has great potential as a powerful tool for anomaly detection in time series data, and has sparked further exploration in this area. However, it has also revealed that similarly to other methods, GANs tend to struggle in identifying extremely rare anomalies, which is a critical aspect in many fields such as fraud detection, medical diagnosis, and equipment monitoring [Blázquez-García et al., 2021]. This finding highlights the need for continued research and development in this area to improve the

ability of GANs to detect extremely rare anomalies and make them a more robust tool for anomaly detection in time series data.

Another important aspect to consider is that this research has been conducted using a variety of real and synthetic datasets that encompass limited ranges of anomaly types and anomaly frequencies. This experimental setup as a comparison method has been met with criticism by some domain experts in recent investigation [Wu and Keogh, 2021]. One of the main criticisms is that the data generation process in synthetic datasets can be biased, leading to unrealistic results. Additionally, there are a lot of factors at play in each dataset, which makes it very difficult to understand why each model behaves as it does. This can lead to unreliable conclusions about the performance of the models being tested.

In this thesis we argue that there has been a lack of systematization in the process of comparing the performance of different anomaly detection methods, specifically in regards to how sensitive they are to variations in the frequency of anomalies.

1.2 Objectives

This thesis aims to address this issue by devising a method to assess how well different models perform as anomalies become rarer. The purpose of this analysis is to shed light on how GANs can be optimized to better detect extremely rare anomalies in time series data, exploring the potential of extending or modifying known GAN architectures to achieve this goal, and understanding the limitations that come with this approach.

Particularly, we will propose two new GAN-based architectures to handle rarer anomaly contexts: *TadGAN-DT* evolves the original TadGAN formulation by integrating an anomaly identification method inspired by Dynamic Thresholding [Hundman et al., 2018], which acknowledges the non-parametric nature of reconstruction errors to find better detection thresholds; *XTadGAN* harnesses meta-information regarding the expected anomaly frequency within a series and applies rarity-based dynamic thresholding and pruning techniques.

The developed framework, referred to as Monte Carlo sampling, generates an arbitrarily large array of time series from a limited set of original datasets, establishing controlled test environments for experiments. It enables a more comprehensive understanding of how algorithms perform under varying conditions and enables a more accurate comparison between state-of-the-art anomaly detection methods. Using this framework, we concentrate on anomaly rarity and systematically evaluate the performance of several algorithms with increasing levels of anomaly rarity. The results of this analysis, coupled with the framework itself, contribute to the field of anomaly detection by establishing a methodological foundation for evaluating detector performance. No single detector is expected to excel under all conditions [Wolpert, 2002], making this a valuable tool for guiding algorithm selection and deployment in specific contexts.

This work also introduces a sensitivity measure for different orders of anomaly rarity – the *rarity-spectrum score* (x_r*-score*). This metric, which can be generalized for any attribute, enables an objective and unbiased assessment of different anomaly detection algorithms across a spectrum

of anomaly frequencies. To the best of our knowledge, the development of this score is significant because it addresses a gap in current research by providing a robust metric that can be used to conduct standardized comparison tests between models. The proposed *sensitivity score* allows for a more accurate and reliable comparison of different anomaly detection models, which can ultimately lead to a better understanding of the strengths and limitations of each model. This can guide future research in the field and ultimately help to improve the performance of anomaly detection algorithms in real-world applications.

The systematic analysis of algorithmic performance in varying anomaly frequencies is interesting on another level as well. Due to the fact that traditional and GAN-based models work fundamentally differently, it is not clear if decreasing the frequency of anomalies would worsen or improve each method's performance. While a case could be made for either outcome, one argument is that decreasing the frequency of anomalies could improve the ability to detect anomalies by allowing it to better understand the normal behavior of the series (especially relevant in the case of generative or statistical methods, like Autoregressive Integrated Moving Average (ARIMA) [Yaacob et al., 2010]). On the other hand, decreasing the frequency of anomalies could also result in decreased performance as the model has fewer examples to learn from (which could be the case for classification or proximity-based methods, such as K-Nearest Neighbors (KNN) [Angiulli and Pizzuti, 2002] or Local Outlier Factor (LOF) [Breunig et al., 2000]).

1.3 Contributions

This thesis focuses on the challenge of detecting extremely rare anomalies with Generative Adversarial Networks. In addition, we also address the lack of systematic comparison in the performance of different anomaly detection methods, specifically regarding how sensitive each one is to variations in the frequency of anomalies.

- *New GAN-Based architectures*. The main contribution of this thesis is the introduction of two new GAN-based architectures to handle extremely rare anomaly scenarios. *TadGAN-DT* evolves the original TadGAN formulation by integrating a non-parametric dynamic thresholding and pruning method. *XTadGAN* harnesses meta-information regarding the expected anomaly frequency within a series and applies rarity-based dynamic thresholding and pruning techniques;

- *Novel Framework for Sensitivity Analysis*. Another significant contribution is the creation of a model-agnostic framework for assessing anomaly detection algorithms across a landscape of time series attributes, particularly to varying levels of anomaly rarity;

- *New Sensitivity Score*. The previous framework is complemented with the development of a *sensitivity score* to evaluate the performance of different anomaly detection algorithms across a range of anomaly frequencies, filling a gap in current research by providing an objective comparison metric;

- *Rarity Sensitivity Analysis*. We use the two previous contributions to conduct a detailed analysis of how different state-of-the-art detection algorithms behave when confronted with variations in anomaly frequency.

The aim is to improve the understanding of the strengths and limitations of each approach, ultimately guiding future research in the field and improving the performance of anomaly detection algorithms in real-world applications.

1.4 Structure

The rest of the document is organized as follows. Chapter 2 summarizes the existing literature on anomaly detection in time series data, highlighting the specific challenges of this field and how GANs can address them, and identifies the gaps that this thesis aims to fill.

In Chapter 3, we introduce a novel approach leveraging Monte Carlo sampling to construct a versatile test bench from pre-existing benchmark datasets, enabling comprehensive model assessments across diverse controlled scenarios. We also propose an innovative metric designed to succinctly encapsulate a model's performance across varying levels of anomaly rarity.

In Chapter 4, we test and evaluate several state-of-the-art machine learning models across a spectrum of anomaly frequencies. Leveraging the insights gained from the previous study, we implement and evaluate two new GAN-based architectures tailored for extremely rare anomalies in Chapter 5. Figure 1.1 presents a condensed schematic view of this pipeline.

Finally, Chapter 6 encapsulates the most significant findings and insights gained through this work and outlines potential pathways for future research that emerge as a result of this work.

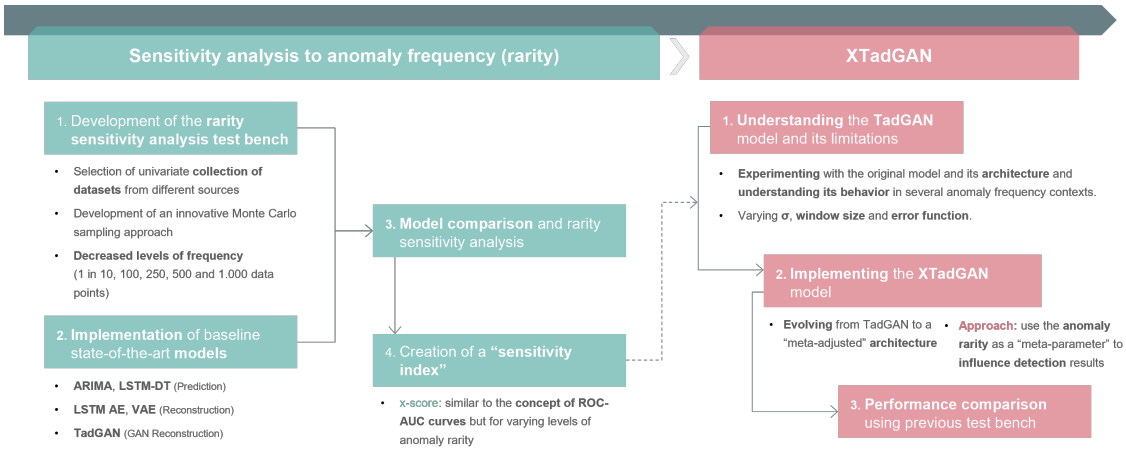

Figure 1.1: Main development pipeline

Chapter 2

Literature Review

The use of GANs for anomaly detection in time series data has attracted attention in recent years due to their ability to model complex distributions. In this literature review, we will provide an overview of anomaly detection in time series data, highlight the specific challenges in this field, and how GANs can address those challenges.

We will cover the basic concepts of anomaly detection in time series, present the most prominent state-of-the-art methods used in real-world applications, and explore the concept and mechanism of using GANs for anomaly detection. Furthermore, we will look into the metrics used to evaluate the performance of anomaly detection methods, and the current limitations and challenges in this particular scope.

By the end of this chapter, we aim to provide a clear understanding of the current state-of-the-art in using GANs for anomaly detection in time series data and to identify the gaps in the literature that this thesis aims to fill. Specifically, the absence of a standard framework for analyzing the sensitivity of each method to extremely rare anomalies, and how this study can help devise better models and architectures to improve overall performance.

2.1 Anomaly Detection in Time Series

Time series anomaly detection refers to the process of identifying instances in a time series that deviate from the expected or normal behavior [Chandola et al., 2009]. These instances, known as anomalies, can indicate important events or changes in the system being analyzed, such as failures, sudden spikes in demand, or deviations from the normal patterns [Wu, 2016]. Anomaly detection in time series can be applied in multiple domains, including finance, healthcare [Pereira and Silveira, 2019; Liu et al., 2015], cyber-security, and energy, to monitor and detect unusual events. Its importance stems from its capability to uncover valuable (and frequently vital) information that can be acted upon by organizations [Chandola et al., 2009].

2.1.1 Taxonomy and disambiguation

The term *anomaly* is widely used in the field of anomaly detection, however, it lacks a clear and agreed-upon definition. This ambiguity in definition has resulted in the term being used interchangeably with related concepts such as *outlier* detection, *novelty* detection, and *rare event* detection. The lack of a standardized definition has made it challenging to understand the processes that generate them and accurately compare and evaluate different anomaly detection methods and techniques [Carreño et al., 2020].

Although some authors such as Aggarwal [2017] use those terms as synonyms, in the context of this thesis we will make a clear distinction between all those concepts. We will adopt the definition used by Grubbs [1969] of an *outlier*: a data point that stands out significantly from the rest of the observations in a sample is referred to as an *outlier*. This can result from variations in measurement, natural data fluctuations, or the result of an experimental mistake. Although not completely explicit, on the basis of this definition are some *a priori* assumptions about the data, such as that it follows a statistical distribution. In other words, an outlier can be seen as a legitimate data point whose characteristics are far away from the mean or median in a distribution. This concept is related to the notion of *point anomaly* presented by Chandola et al. [2009], which is explored in further detail in Section 2.1.2.

On the other hand, an *anomaly* is an instance (or sequence of instances) that does not conform to an expected pattern of the other instances in the data. This, in turn, may suggest that it was generated by a different process than the one that generated the normal data [Pimentel et al., 2014]. For instance, it is not inherently incorrect (although highly unlikely) to claim that a specific set of observations consists of 25% anomalies. However, the same cannot be asserted for outliers since they are, by definition, rare.

In essence, this definition of an *anomaly* generalizes the concept of an *outlier* as an abnormality in the data, as it no longer encompasses only points that are deviant from the distribution, but extends it to include context: an instance or pattern that may be inside the expected values of the normal data, but whose presence in a specific setting (in the case of time series, a specific time interval) is abnormal (this concept is also further explored in Section 2.1.2). In other words, although the two concepts are related, an outlier can be an anomaly, but not every anomaly is an outlier.

Novelty detection is also sometimes used in the context of anomaly detection [Bishop, 1994; Pimentel et al., 2014; Singh and Markou, 2003]. It is the task of identifying previously unseen (emergent, new) patterns in the data. The difference between novel patterns and anomalies is that once detected, novel patterns are often integrated into the normal model [Chandola et al., 2009].

It should be noted that this distinction and taxonomy used henceforth in this thesis differs from Carreño et al. [2020], where the authors try to unify these terms based on the nature of the problem (supervised *versus* unsupervised) and the data (temporal *versus* static).

2.1.2 Types of Anomalies

Anomalies in data can come in many forms, depending on the source and its domain. These are usually classified into three classes [Chandola et al., 2009]. Figure 2.1 provides a visual example of each.

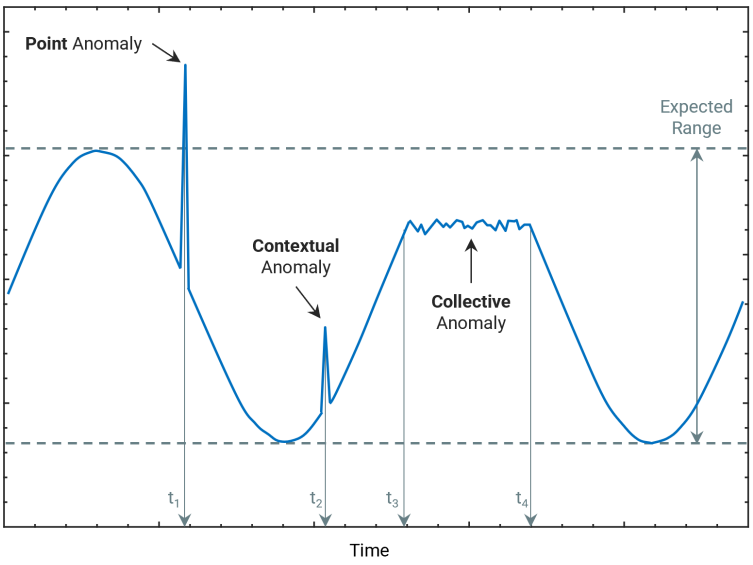

Figure 2.1: Point, Collective and Contextual anomalies in time series. Adapted from Chandola et al. [2009]

- **Point anomalies**: individual instances in a dataset that are significantly different from the rest of the data, for example, an unexpectedly high or low value in a time series [Shaukat et al., 2021]. These are the most basic type of anomaly and their identification can usually be conducted without considering any other input data points. This is intimately related to the previously discussed concept of *outlier* [Wu, 2016]. In the context of time series, point anomalies are flagged by a single timestamp.
- **Collective anomalies**: if a sequence of consecutive related anomalous observations is observed, it is referred to as a collective anomaly. Each data point in a collective anomaly may not be considered anomalous individually, but their occurrence as a group is anomalous [Foorthuis, 2021]. Such anomalies are identified by an interval of start and end timestamps in time series anomaly detection.
- **Contextual anomalies**: a data point that is anomalous only within a particular context is known as a contextual anomaly [Chandola et al., 2009] or a conditional anomaly [Song et al., 2007]. The context is defined as part of the problem formulation: each instance is defined using a contextual attribute (often time) to define the neighborhood, and behavioral attributes that define the characteristics/values of the series. The anomaly is thus determined using the values for the behavioral attributes within a specific local context.

2.1.3 Labels and Training Strategies

Labels associated with a data instance indicate if that observation is considered normal or anomalous. However, these labels may not always be available for training. Thus, anomaly detection learning techniques can be categorized into three modes based on the availability of labels: supervised, semi-supervised, and unsupervised [Chandola et al., 2009].

2.1.3.1 Supervised Learning

Supervised learning involves training an algorithm with labeled data, where anomalies have already been identified and labeled as such. This model can then be used to make predictions on new, unlabeled data and identify anomalies. The typical approach is to build a binary classification model for normal *versus* anomalous classes.

This approach is best employed when there is ample and reliable labeled data available for training. However, this is hardly the case in most real-world applications for anomaly detection, where it is usually difficult to obtain accurate labels [Steinwart et al., 2005]. Another major issue is the inherent imbalanced nature of this type of problems [Joshi et al., 2001], with far fewer anomalous observations than normal instances, which brings challenges to training and evaluating models.

2.1.3.2 Semi-supervised Learning

In semi-supervised learning, the model is usually trained on a dataset that contains both labeled (albeit in a smaller portion) and unlabeled observations. While supervised learning relies exclusively on labeled data, semi-supervised learning incorporates both labeled and unlabeled data to improve model performance [Ruff et al., 2019]. However, some literature also considers some scenarios where only one of the classes–either the normal or the anomalous class–is used for training [Akcay et al., 2019].

This approach is particularly useful when labeled data is scarce or expensive to obtain. Additionally, it can also be useful when the anomalies are rare and difficult to identify, as the model can use both labeled and unlabeled data to identify these anomalies.

2.1.3.3 Unsupervised Learning

Unsupervised learning is a type of machine learning approach where the model is trained on unlabeled data only. Its goal is to identify patterns and relationships within the data, without the guidance of labels [Shaukat et al., 2021].

Behind this learning strategy, there is the implicit assumption that normal instances are far more frequent than anomalies [Goswami et al., 2022]. When this is not the case, models usually suffer from a high false positive rate.

One advantage of unsupervised learning is that it does not require labeled data, and is thus most widely applicable in real-world applications. Furthermore, unsupervised models can identify

anomalies that may be different from what was seen in past data, making it a useful approach for identifying previously unseen anomalous patterns.

Given the scope of this thesis, a more thorough and in-depth analysis of unsupervised techniques in the context of time series anomaly detection is presented.

2.2 Unsupervised Approaches for Anomaly Detection in Time Series

The diverse range of anomalies, data types, and use cases has led to a proliferation of very different anomaly detection methods [Hodge and Austin, 2004]. As the main subject of interest behind this thesis, this section focuses on unsupervised approaches.

2.2.1 General Formulation for the Unsupervised Context

The goal of unsupervised time series anomaly detection is to identify anomalous sub-sequences of varying lengths within a time series. Let $X = (x_1, x_2, \ldots, x_T)$ where $x_i \in \mathbb{R}^N$ denote a multivariate time series of length T with N measurements at time step i. The goal is to find a set of n anomalous time segments $A = (a_{ij}^1, a_{ij}^2, \ldots, a_{ij}^n)$, where a_{ij}^k represents a continuous sequence of time steps of varying length starting at time i and ending at time j that shows anomalous behavior.

This process involves determining the level of dissimilarity or difference between the data points in the time series X and its normal/expected behavior. This is usually done by computing a measure of distance between the two and flagging instances where this measure exceeds a predefined limit [Goswami et al., 2022].

The most straightforward approach is the *out-of-bounds* method, which identifies instances where values exceed a predefined threshold. However, this method is not very flexible and, most importantly, not able to detect contextual anomalies [Goldstein and Uchida, 2016]. To address this limitation, more sophisticated techniques have been developed, such as proximity-based, prediction-based, and reconstruction-based methods. In the following sections, we outline each of these approaches and mention some of the more notable models.

2.2.2 Proximity-based methods

Proximity-based anomaly detection uses a distance measure to evaluate similarities between data instances — either single points for point anomalies or fixed-length sequences for collective anomalies. In this context, instances that are far from the main bulk of the data are considered anomalies. This approach can be further divided into two sub-categories:

- Distance-based methods, such as K-Nearest Neighbors (KNN) [Angiulli and Pizzuti, 2002], which uses a predefined radius to identify an instance's neighbors and a count of neighbors to determine an anomaly score.
- Density-based methods, such as Local Outlier Factor (LOF) [Breunig et al., 2000] or Connectivity-based Outlier Factor (COF) [He et al., 2003], consider the density of the instance as well

as that of its neighbors. These methods assume that normal data instances occur in dense neighborhoods, whereas anomalies tend to be isolated from their nearest neighbors.

According to Chandola et al. [2009] and Wu [2016], there are two significant limitations to applying proximity-based methods to time series data: (1) prior knowledge of anomaly duration is necessary, and (2) these methods cannot account for temporal correlations. As such, this family of approaches has many limitations in dealing with contextual anomalies.

2.2.3 Prediction-based methods

Prediction-based approaches build a predictive model using the given time series data and then use it to predict future values. An instance is considered anomalous if the discrepancy between its predicted value and the actual value exceeds a predetermined threshold [Kozitsin et al., 2021].

Although usually very powerful, these statistical models such as ARIMA [Yaacob et al., 2010], Holt-Winters [Pena et al., 2013], and Functional Data Analysis (FDA) [Torres et al., 2011] often demand extensive domain expertise to correctly adjust their hyperparameters and strong assumptions about the data (stationarity, for example).

To address the limitations of earlier methods, recent machine learning-based techniques have been proposed. Malhotra et al. [2015] and Hundman et al. [2018] used Long Short Term Memory (LSTM) Recurrent Neural Networks (RNN) to predict future time steps and identify large deviations from predictions. Hierarchical Temporal Memory (HTM) was introduced by Ahmad et al. [2017], which converts the current online sequencial input into a hidden state and predicts the next hidden state, with the prediction error calculated by comparing the current and the predicted hidden states. This class of machine-learning based methods has been proven to outperform traditional models given sufficient training samples in forecasting problems [Cerqueira et al., 2019].

2.2.4 Reconstruction-based methods

Reconstruction-based techniques model the underlying structure (low-dimensional representation) of the provided time series data and then generate a synthetic reconstruction [Goldstein and Uchida, 2016]. On the basis of this method is the assumption that anomalous instances lose information when mapped to a lower dimensional space, and hence are not effectively reconstructed. In this setting, an anomaly is identified in sequences where reconstruction errors between the original data and the reconstructed data are high.

One of the most well-known dimensionality-reduction methods is Principal Component Analysis (PCA), which is used in some contexts for reconstruction. One limitation of this method, however, is that it is only applicable to linear reconstruction and Gaussian, highly correlated data [Hyndman et al., 2015].

As in the case of prediction-based approaches, recent breakthroughs in computational power have increased efforts to explore deep learning techniques. Notable mentions are the use of Auto-Encoders (AE) and Variational Auto-Encoders (VAE) [An and Cho, 2015]; and LSTM Encoder-Decoder architectures [Malhotra et al., 2016]. However, according to Geiger et al. [2020] these

approaches have a tendency for overfitting if not properly regularized. To address this issue, some authors have recently proposed the use of adversarial training to produce more robust models for time series reconstruction. A more detailed review and discussion about the use of GANs for time series anomaly detection is addressed in the following section.

2.3 GANs for Anomaly Detection in Time Series

A Generative Adversarial Network (GAN) is a type of deep learning architecture first proposed by Goodfellow et al. [2014] that involves two neural networks competing against each other in a zero-sum game[1]. One network, the generator, creates synthetic data, while the other network, the discriminator, evaluates the authenticity of the synthetic data compared to real data. The generator's goal is to create data that the discriminator cannot distinguish from the real data, while the discriminator's goal is to accurately identify the synthetic data [Creswell et al., 2018]. The two networks are trained simultaneously, and over time, the idea is that the generator improves its ability to produce realistic synthetic data.

Since its introduction, GANs have been used in a variety of applications, more notably in image synthesis and video generation. Only recently have some works been published that apply the concept of adversarial training to time series anomaly detection. The first proposed method, called MAD-GAN, was presented by Li et al. [2019]. The authors use a standard GAN to model the time series data, using the Discriminator to flag anomalies, and show that it outperforms traditional methods in terms of accuracy and computational efficiency.

Zhou et al. [2019] propose a GAN architecture as an anomaly detection method for heartbeat signals. The main paradigm shift was the introduction of the reconstruction error as a metric for detecting and signaling anomalies, instead of the output of the Discrmininator network.

In the TadGAN paper, Geiger et al. [2020] aim to improve on the previous works by training the model with cycle consistency loss [Zhu et al., 2017]. This loss encourages the model to produce translations that are not only realistic but also maintain the essential content and structure of the original data. Furthermore, they explore different approaches to combine reconstruction errors and Discriminator outputs to compute anomaly scores. The authors demonstrate the effectiveness of their approach through extensive experiments on various real and synthetic datasets and comparing the results with state-of-the-art methods. Since this publication, no relevant work has been done in pursuing advances on top of TadGAN, with only some application reports being released.

As the current state-of-the-art GAN-based method in time series anomaly detection, this approach will serve as the basis for our work. A more in-depth analysis of the architecture and inner-workings of the model is presented below.

[1] A zero-sum game is a mathematical concept in game theory, where one player's gain is exactly balanced by the losses of the other player(s). In other words, the total net benefit for all participants is zero. Examples of zero-sum games include chess or tic-tac-toe.

2.3.1 TadGAN

In this section, we will delve into the TadGAN architecture proposed by Geiger et al. [2020] and detail its components and implementation. In the following paragraphs, we will examine its problem formulation, objective function, and key advancements compared to previous work on using GANs for anomaly detection.

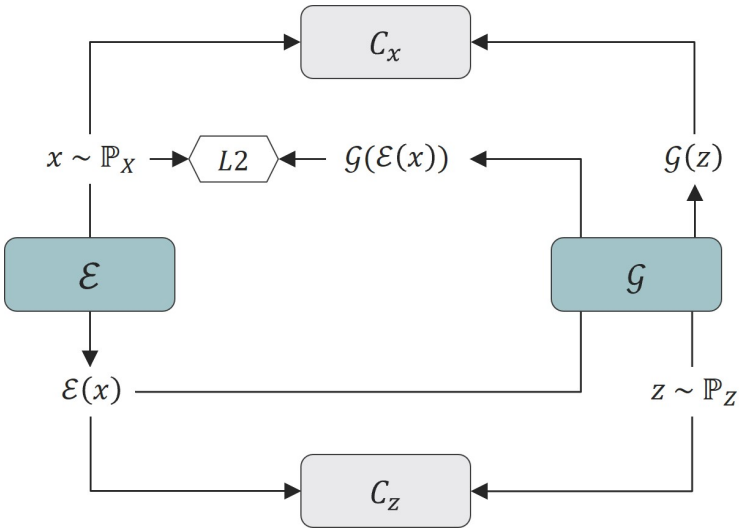

Figure 2.2: TadGAN Architecture as proposed by Geiger et al. [2020]. Adapted from the original paper.

As we can see in figure 2.2, the TadGAN model is composed of an inner *Encoder-Decoder* pair of networks, that learn two mapping functions between domains X (original) and Z (latent space). Both of these networks can be seen as *Generators*: $\mathcal{E}$ works as the Encoder that maps the series to the latent space Z, whereas $\mathcal{G}$ works as the Decoder that transforms the latent space into the reconstructed time series X. The original time series can thus be reconstructed by: $x_i \rightarrow \mathcal{E}(x_i) \rightarrow \mathcal{G}(\mathcal{E}(x_i)) \approx \hat{x}_i$. Random vectors z are sampled from Z following a standard normal distribution to represent white noise.

This Encoder-Decoder core is complemented by two adversarial *Discriminators* (denoted as *Critics* in the original paper), $\mathcal{C}_x$ and $\mathcal{C}_z$. The former is responsible for discerning between the original time series observations from X and generated time series instances from $\mathcal{G}(z)$; the latter evaluates the efficiency of the mapping into the latent space Z.

The end goal is for $\mathcal{G}$ to fool $\mathcal{C}_x$ by producing real-looking instances. To accomplish this, the following MinMax problem is formulated:

$$\min_{\{\mathcal{E},\mathcal{G}\}} \max_{\{\mathcal{C}_x,\mathcal{C}_z\}} V_X(\mathcal{C}_x,\mathcal{G}) + V_Z(\mathcal{C}_z,\mathcal{E}) + V_{L2}(\mathcal{E},\mathcal{G}) \tag{2.1}$$

The Wasserstein losses $V_X(\mathcal{C}_x,\mathcal{G})$ and $V_Z(\mathcal{C}_z,\mathcal{E})$ are here applied as the adversarial loss to train the

GANs in order to overcome the mode collapse problem[2]. The term $V_{L2}(\mathcal{E},\mathcal{G})$ is also used to adapt the concept of cycle-consistency loss (L2 norm of the difference between the original observations and the reconstructed instances) to decrease the possible search space for the mapping function.

This architecture offers some clear advantages in comparison to the previous proposals Li et al. [2019]; Zhou et al. [2019]: firstly, we can use $\mathcal{C}_x$ as a measure of anomalous instances, as it is trained to recognize between the original and reconstructed time series; secondly, because of the implementation of cycle-consistency loss, one can also use the difference between the original and decoded sequence as detection metric. The authors explore several combinations of these two measures in the original paper and report on their performance.

The most effective configuration employs a product combination of these two measures. It first calculates the reconstruction error using *Dynamic Time Warping* (DTW) [Berndt and Clifford, 1994] to compare the mean squared error between the reconstructed and original series values. Then, it utilizes outputs from the Critic networks to measure how anomalous a time segment is. These outputs are normalized, and the final anomaly scores are obtained by multiplying their z-scores using the formula:

$$score(x) = \alpha Z_{RE}(x) \odot Z_{\mathcal{C}_x}(x) \tag{2.2}$$

After computing the anomaly scores, the model employs a simple parametric thresholding technique to identify candidate anomalous sequences. This is done by setting a static threshold at 4 standard deviations from the window's mean. Sequences that score higher than this boundary are classified as anomalies.

Although beneficial for recall, this method tends to produce many false positives. The authors employ an anomaly pruning approach inspired by Hundman et al. [2018] to mitigate this risk. A detailed explanation of this method is described in section 5.1.

The complete algorithm for this architecture is outlined in Appendix A.4.

2.4 Extreme Anomalies

There is no formal, universally accepted definition of an extreme anomaly in the context of anomaly detection. The specific criteria for what constitutes an extreme anomaly can vary depending on the application and the data being analyzed.

One could use a *statistical* definition, where an extreme anomaly may be defined in terms of a specific statistical criterion, such as the probability of occurrence or the distance from the mean of the data [Aggarwal, 2017; Hawkins, 1980; Tukey, 1977]. For example, in some cases, an extreme

[2] A common issue in GANs, where the generator produces a limited diversity of samples that often converge to a few modes or patterns in the target distribution, ignoring other possible variations. In essence, the generator becomes too focused on generating specific data points, neglecting the richness of the entire data distribution it is supposed to capture. This can be mitigated by using the Wasserstein loss, which encourages the generator to produce a diverse range of samples and capture a more accurate representation of the target data distribution.

anomaly may be defined as a data point that is a certain number of standard deviations away from the mean of the data. Nevertheless, even in this case, the definition of an extreme anomaly is dependent on the threshold used to catalog a point or sequence as an anomaly: an extreme anomaly may be defined as a data point that is more than three standard deviations from the mean, while in other cases it may be defined as a data point that is outside of the 99th percentile of the data.

While interesting, the previous definition and scope will not be the focus of this investigation. In this thesis, an extreme anomaly will be defined based on the rarity or frequency of the data point or sequence. Defining extreme anomalies based on a rarity or frequency criteria can be much more useful in many real-world applications, as these instances often have a profound impact on the system being analyzed, and correctly identifying them can make the difference between a critical failure or successfully mitigating the situation.

This will lay the foundation for our study, where we will explore the detection of extreme anomalies at varying levels of frequency (such as one anomaly every 1 000, 10 000, or 100 000 data points), to gain a better understanding of the behavior of different anomaly detection approaches.

Defining extreme anomalies based on rarity provides a clear criterion for identifying these instances, but it is important to note that rarity or frequency alone may not always be a sufficient measure of comparison. In order to gain a comprehensive understanding of extreme anomalies, it is necessary to consider other factors, such as the context of the data and the specific system being analyzed. Nevertheless, by exploring the detection of extreme anomalies at different levels of frequency, this study aims to provide new insights into this field and contribute to the development of more effective anomaly detection methods.

2.5 Evaluation

In this section, we are going to discuss the different aspects relevant to evaluating the performance of an unsupervised anomaly detection model against the ground truth.

In an unsupervised context, the model only receives time series values at each time step as input, without accompanying labels. Using this experimental setup, it would be difficult to assess the real performance and accuracy of the model. Therefore, unsupervised models are usually evaluated on multiple pre-labeled time series datasets (more on this topic in Section 2.5.3). Using these labels as the ground truth, we can evaluate the results of our model and determine how well it is performing.

2.5.1 Evaluation Metrics

In the context of anomaly detection, the most widely adopted mechanism to evaluate the performance of a model is to use the concepts from binary classification, regarding each anomalous instance as the positive class [Goldstein and Uchida, 2016]. Thus, we have the following terminology:

- **True Positive (TP)**: an instance where an anomaly is correctly identified as such by the model;
- **False Positive (FP)**: an instance where a normal observation is mistakenly identified as an anomaly by the model;
- **True Negative (TN)**: an instance where a normal observation is correctly identified as such by the model;
- **False Negative (FN)**: an instance where an anomaly is wrongly identified as a normal observation by the model.

These terms can be used to compute several classification metrics such as precision, recall, and f1-score:

- **Precision** $[TP/(TP+FP)]$: number of true positives divided by the sum of true positives and false positives. Precision measures how many of the anomalies detected by the model are actually true anomalies. A high precision value indicates that the model has a low rate of false positive errors;
- **Recall** $[TP/(TP+FN)]$: number of true positives divided by the sum of true positives and false negatives. Recall measures how many of the true anomalies are detected by the model. A high recall value indicates that the model has a low rate of false negative errors;
- **F1-score** $[2 \times Pre \times Rec/(Pre+Rec)]$: harmonic mean of precision and recall. The F1 score balances precision and recall and provides a single value that represents the overall performance of the model. A high F1 score indicates that the model has a high level of precision and recall;

However clear and easy to implement, these measures have a major limitation when applied to time series: they do not take time into account. As such, some strategies have to be devised in order to extend the previous metrics to take time into account.

2.5.2 Evaluation Strategies

In order to evaluate the performance of an anomaly detection model in time series, two main strategies are usually deployed. The first approach, known as the *weighted segment* approach, involves comparing each detected anomalous segment with its corresponding segment in the ground truth [Lavin and Ahmad, 2015a].

The second approach, known as the *overlapping segment* approach, involves evaluating the performance of the model by examining the overlap between the detected anomalous segments and the ground truth anomalies [Hwang et al., 2019].

It is worth noting that each of these approaches evaluates a distinct objective, and emphasizes different characteristics of the model. Therefore, both are described in detail below. Figure 2.3 gives a visual perspective on the different outputs of each alternative.

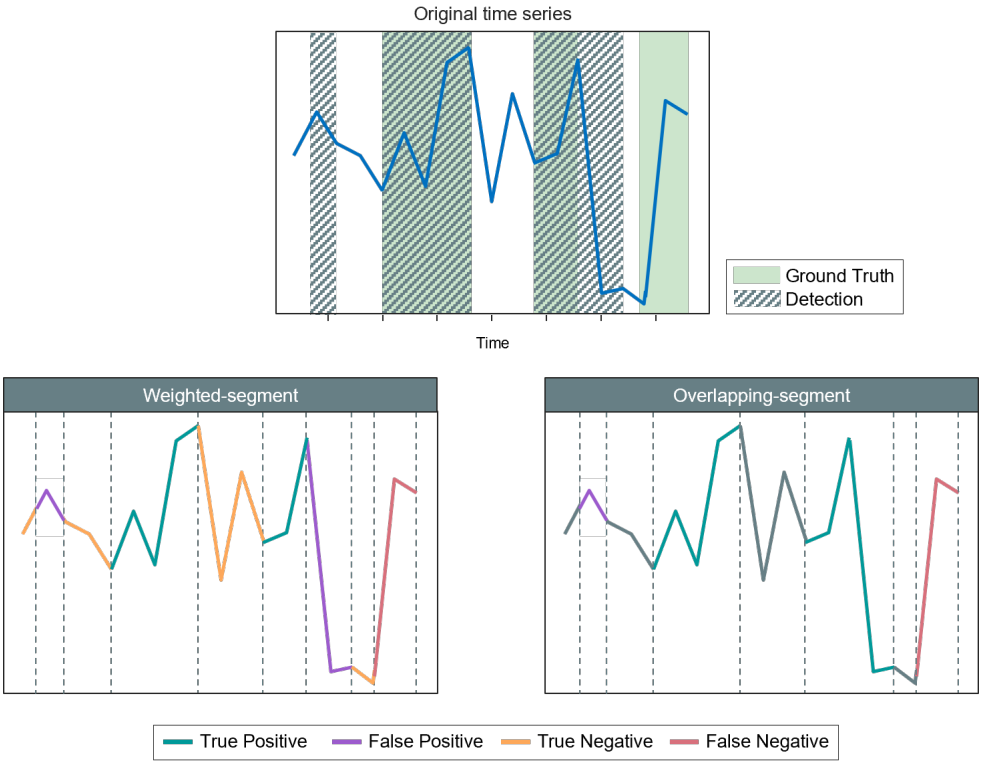

Figure 2.3: Comparison between the weighted-segment and overlapping-segment evaluation strategies

- **Weighted Segment**: this approach involves dividing the time series data into segments based on the ground truth and detected sequences. Each data point in the anomalies detected by the model is then compared with its counterpart in the ground truth, and a score is assigned based on the match between the two. The overall score is calculated by considering the duration (size) of each segment [Huet et al., 2022]. This method is stricter and is useful when we want to give equal importance to detecting anomalies and normal instances.

- **Overlapping Segment**: overlapping segment-based evaluation is a more flexible method and operates under the premise that if the model identifies a subset of an anomaly, it should still be rewarded [Tatbul et al., 2018]. In practice, this gives an incentive to the model even in cases where only a partial identification is correctly done: in a real-world application, assuming a set of users are monitoring the signals, even if the model only partially identifies the anomaly, the users would be able to examine the entire anomaly as the alert has brought it to their attention.

  In this evaluation approach, True Negatives are not recorded. The method considers a True Positive (TP) if a ground truth segment overlaps with the detected segment; a False Negative (FN) if the ground truth segment does not overlap any detected segments; a False Positive (FP) if a detected segment does not overlap any labeled anomalous region.

As one can infer from the previous analysis, the *overlapping segment* approach is expected to generally have higher values for precision in comparison to *weighted segment* since partial detections are recorded as true positives [Hwang et al., 2019].

2.5.3 Benchmark datasets

Almost all academic research in the field of anomaly detection in time series data has been performed using three main sources of data: NASA[3] [Hundman et al., 2018], Yahoo[4] and Numenta Anomaly Benchmark (NAB)[5] [Lavin and Ahmad, 2015b], in addition to private datasets whose confidential data is not publicly available for reproducibility.

This could be seen as a positive aspect, as it creates a level playing field for comparison between models and makes it easier to compare and validate results. However, recent studies have highlighted a variety of issues with these sources of data that make them far from ideal as a reliable basis for comparison [Wu and Keogh, 2021].

The authors argue that the current benchmarking methods suffer from several limitations, having categorized them in four *flaws*: *(1) triviality, (2) unrealistic anomaly density, (3) mislabeled ground truth* and *(4) run-to-failure bias*.

- **Triviality**: the authors describe the flaw of "triviality" as the lack of difficulty in the tasks presented by the benchmark datasets. They argue that many of the datasets used in the current benchmarking methods are too simple and do not reflect the complexity and difficulty of real-world data, leading to an underestimation of the difficulty of the task and an overestimation of the performance of the models. To prove the previous point, the authors have shown that a great portion of the available datasets can easily be solved using simple thresholding and clustering techniques;

- **Unrealistic anomaly density**: related to the discrepancy between the anomaly density in the benchmark datasets and the anomaly density in real-world data. They argue that many of the benchmark datasets used contain an unrealistic high density of anomalies, which leads to an overestimation of the models performance, as more anomalous instances are present to train. This unrealistic density of anomalies is mainly caused by the artificiality of the datasets, which are often generated in a controlled environment and do not accurately reflect the sparse and irregular distribution of anomalies in real-world data;

- **Mislabeled ground truth**: this is described as the incorrect labeling of anomalies in the benchmark datasets. They argue that many of the data sources contain errors in the labeling of anomalies, which can lead to incorrect evaluation of the performance of models. These

[3] A spacecraft telemetry signals dataset, available at https://github.com/khundman/telemanom

[4] A set of datasets consisting of real metrics and synthetic signals of Yahoo service. Available under request at https://webscope.sandbox.yahoo.com/#datasets

[5] A repository for anomaly detection in streaming, real-time applications with several real and artificial signals on different domains. Available at https://github.com/numenta/NAB

errors in labeling can be due to human error in the annotation process, a lack of clear definition or understanding of what constitutes an anomaly, or inconsistencies in the labeling of anomalies;

- **Run-to-failure bias**: defined as the tendency to only include a limited number of anomalies appearing towards the end of the datasets, because many real-world systems are run-to-failure. The authors argue that this can be a problem as it drastically affects the default rate and can lead to biased results, and that a naive model that only labels the last observation(s) as anomalous has a high chance of being correct. As such, using these datasets may not reflect the true performance of the model over an extended period of time.

These limitations make it difficult to draw meaningful conclusions from the results obtained using these sources and can lead to over-optimistic or unrealistic evaluations of the performance of anomaly detection models. In order to obtain a more accurate and comprehensive evaluation of the performance of these models, the authors propose and create a new set of benchmark datasets, The UCR Time Series Anomaly Archive[6] [Dau et al., 2018].

Having these limitations in mind, we will nevertheless use all of the benchmark data sources presented above for the development of our work. This should both serve as a means of exploring if we can reproduce the originally reported results for the methods we will use as a comparison for our XTadGAN model, but also to understand if we can see a difference in the behavior of each method when applied to the UCR archive.

2.6 Summary

The literature review conducted for this thesis has yielded several important insights, and has highlighted the need for continued research into the field of anomaly detection in time series.

Perhaps the more interesting conclusion is that no single research was found that made reference to exploring the behavior and performance of different approaches to varying levels of anomaly rarity in time series. Although many of the previous datasets have diverse characteristics and different anomaly frequencies, there has been no prior effort in devising an experimental setup that isolated and explored this variable. As such, we find this gap in the current literature as one of the main contributions to address during this thesis. The details for this experimental setup are discussed in Section 4.1.

The literature review also demonstrated the potential of using adversarial training to enhance the performance of conventional methods for anomaly detection in time series. Despite this potential, the use of GANs in this area is still in its early stages, providing ample opportunities for further research and improvement.

[6] Publicly available at https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

Finally, while the majority of academic research in this context uses three benchmark datasets, recent investigation has shown that these datasets have several shortcomings that can affect the validity of the results. This highlights the importance of using more diverse and realistic datasets in future research and development. Taking this into consideration, all models and developed work within this thesis will strive to address these limitations by incorporating more representative and diverse datasets.

Chapter 3

A Novel Framework for Sensitivity Analysis in Time Series

The main focus of this work is to understand how different machine learning models behave as a function of anomaly frequency in time series data, and how GAN-based architectures can be enhanced to detect extremely rare anomalous instances.

However, to the best of our knowledge, no relevant works have been identified that concentrate on this specific topic or engage in a systematic exploration of models' responsiveness concerning a spectrum of anomaly frequencies in time series data. One work by Emmott et al. [2013] briefly mentions this subject, recognizing anomaly frequency as an important dimension within anomaly detection benchmarks, but not concerning the context of time series.

Thus, a major contribution of this thesis is to address the current absence of a controlled and systematic comparison framework regarding the sensitivity of the most prominent detection methods to variations in the frequency of anomalies, and how this study can be used to improve current state-of-the-art approaches.

The current state of research in time series anomaly detection faces several challenges that hinder the development of robust and sustainable work. Many of these issues have been discussed in Chapter 2, as highlighted by the work conducted by Wu and Keogh [2021]. In the context of our specific problem, we would like to add and highlight two key challenges that significantly impact research:

- *Limited Data Availability*: the effective training and evaluation of machine learning algorithms relies on datasets with an adequate sample size. However, obtaining a sufficient number of samples often proves difficult due to inherent data scarcity in certain domains, the costs and practicality of data collection, and privacy concerns across some crucial applications;

- *Bias and Under-Representation*: even when a satisfactory sample size is available, benchmark datasets might be influenced by biases in data collection and may not adequately represent certain groups (for instance, only representing a narrow range of anomaly frequencies). By employing these datasets as benchmarks, there is an inadvertent encouragement for the development of algorithms that perpetuate or exploit existing biases.

Although, by definition, time series have a very specific and rigid structure which makes it particularly hard to extrapolate, issues regarding current evaluation procedures make recent results questionable [Wu and Keogh, 2021]. We argue that the aforementioned issues have contributed to a noticeable lack of generalization in current models.

Our objective was thus to develop a comprehensive framework that facilitates the creation of controlled and systematic experiments, even if starting from a limited set of initial datasets.

3.1 Monte Carlo sampling for time series

The proposed approach, named Monte Carlo sampling, is a method for constrained, systematic generation of semi-synthetic time series. It addresses the need of researchers to gain a deeper understanding of how various methods perform under specific conditions, such as different frequencies of anomalies in time series data. This approach allows the creation of semi-synthetic time series that meet predefined criteria, including constraints on key meta-features related to anomalies. Figure 3.1 condenses the Monte Carlo sampling pipeline.

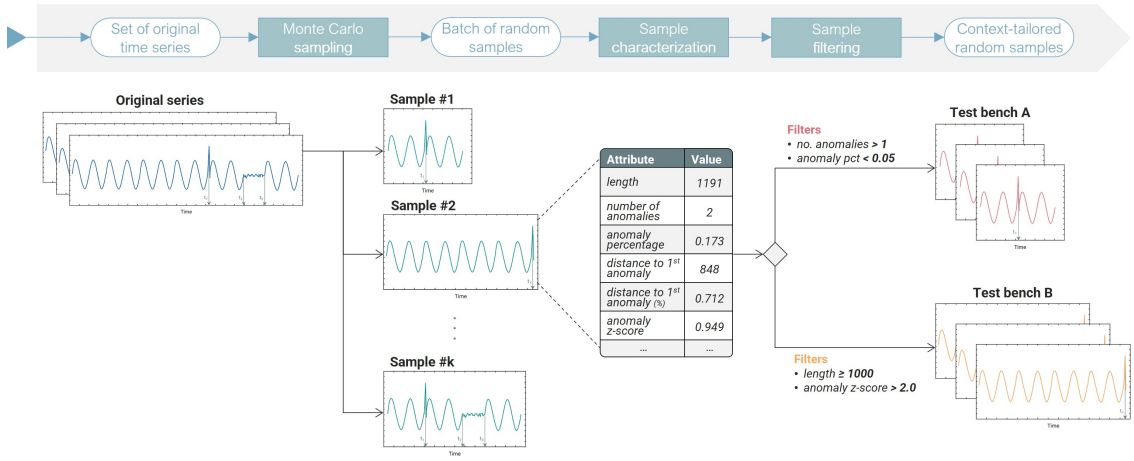

Figure 3.1: Schematic representation of the Monte Carlo sampling process

In simplified terms, the Monte Carlo approach is employed to create a substantial number of new time series by sampling from existing ones. Subsequently, these generated time series are filtered to extract a subset that aligns with the specific requirements and conditions needed for a specific experiment. A comprehensive explanation of each stage in the pipeline is provided below.

Sampling

We want to generate an arbitrarily large number of time series, each representing different controlled scenarios, derived from a relatively small set of original datasets.

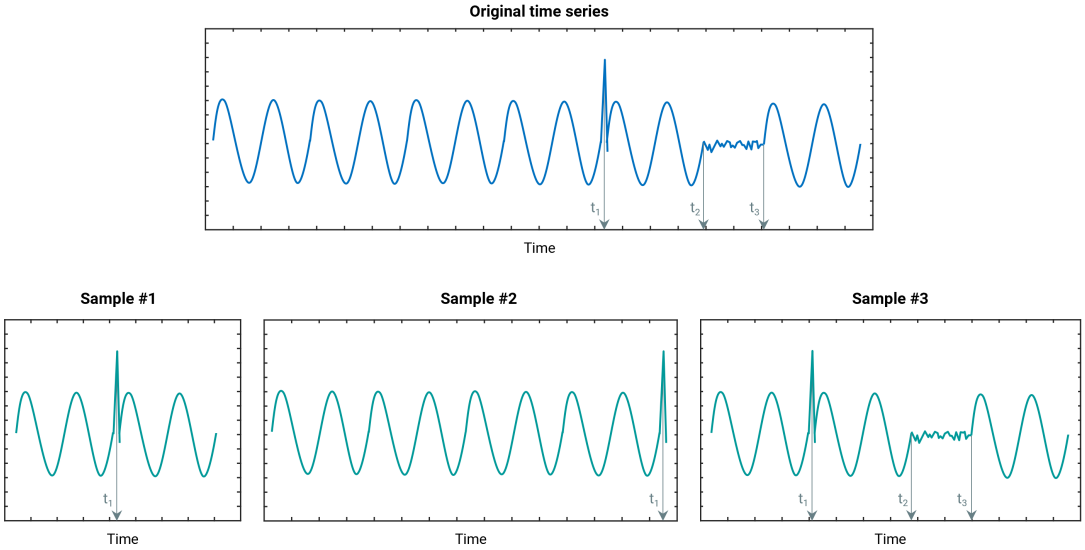

Figure 3.2: Generating samples using a Monte Carlo-inspired strategy

Our method draws inspiration from the Monte Carlo approach, taking a novel approach to address the challenges of controlled experimentation with time series data. Starting from a single or a set of time series, our method generates a large number of randomly trimmed copies of the original data. The output samples are created by extracting sub-sequences of varying lengths from the original series, resulting in several trimmed instances with distinct properties. To generate these sub-sequences, two points are randomly selected from the original series to mark the start and end of the sample. Figure 3.2 shows an example of this method.

Sample characterization

However, not all generated sub-sequences are useful: some may be too small, occur too late in the series, or not contain any anomaly at all. Hence, during this generation process, a wide range of attributes, or dimensions, is computed to characterize each resulting sample. These attributes can be tailored to the specific context of interest and may include metrics such as anomaly frequency, average distance between consecutive anomalies, mean anomaly distance from the series mean, and distance (time) to the first anomaly, among others. Appendix A.2 provides an extensive list of attributes computed by our implementation.

The previously calculated attributes provide a multidimensional profile for each generated sample, offering a rich description of its characteristics. Figure 3.3 depicts an example of some attributes histograms for 10 000 generated samples using random lengths. To enhance readability, only the first 9 attributes are displayed. The complete output can be seen in Appendix A.1.3.

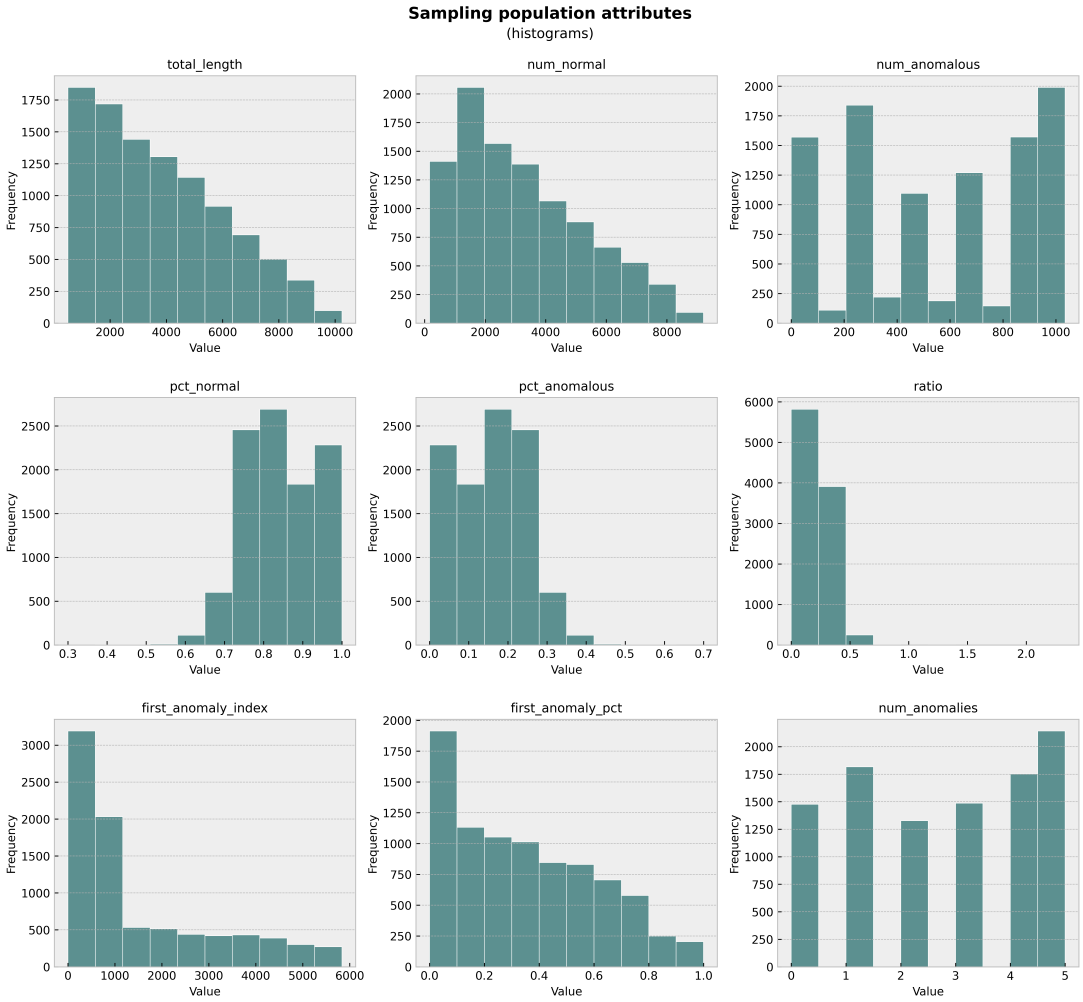

Figure 3.3: Histogram for each sampling population attribute (cropped)

Filtering

Using this pool of generated samples, our approach enables the application of simple filters to specific attributes, either individually or in combination. This capability allows the construction of controlled test environments for evaluating the sensitivity of algorithms. For instance, to assess how various models perform as anomalies approach the mean of the series, an effective experimental setup involves crafting a test bench with multiple samples featuring decreasing levels of some *out-of-distribution* measure (e.g., anomaly z-scores). Through our proposed method, this can be achieved by generating a multitude of trimmed series and then filtering for samples where anomalies fall within a designated z-score range or bin.

With an ample number of original series and a substantial volume of generated samples, these sub-sequences comprehensively cover nearly all conceivable scenarios arising from the interplay of the previously defined attributes. This approach provides researchers with a powerful tool to

systematically explore a wide array of scenarios that hold significance for their investigations, and investigate algorithm behavior under various conditions and parameters.

What sets our approach apart is its ability to initiate these investigations from a limited set of original datasets and time series. Instead of requiring an exhaustive dataset collection effort, researchers can leverage a foundational dataset as a starting point. From there, they can systematically manipulate the parameters within the framework, creating tailored experiments that emulate real-world conditions while maintaining control over variables.

A limitation of this method is that experiment requirements must be expressed in terms of computable data characteristics. Nevertheless, we contend that this limitation is substantially mitigated by the fact that the set of computed meta-features is extensible, which can accommodate a wide array of experiment scenarios.

This approach not only saves time and resources but also ensures that experiments remain controlled and consistent. This is particularly valuable as it allows researchers to assess algorithmic performance across a wide spectrum of scenarios, providing a deeper understanding of their strengths and limitations. It also reduces the dependence on synthetic datasets or real-world time series manipulation techniques, which can introduce bias and alter the underlying distribution of the analyzed signal.

3.2 Sensitivity score

As previously mentioned, another notable gap in the existing literature pertains to the absence of an objective metric capable of encapsulating algorithm performance across a range of experiments, specifically when varying a particular attribute's values. Our proposed approach, which enables the assessment of method performance under systematically varied conditions, presents an opportunity to establish such an aggregated measure.

Addressing this gap, we have introduced a novel family of metrics termed the *x-score*. This innovative suite of metrics can be seen as a *sensitivity index* to assess and quantify the performance of any anomaly detection model to a particular attribute.

3.2.1 The rarity-spectrum score: x_r-score

When designing our experimental setup (using the Monte Carlo sampling approach) for studying algorithm behavior as a function of anomaly rarity, we aimed to generate graphs with curves similar to ROC[1] curves, showing the performance of a specific model at every anomaly frequency threshold. Figure 3.4 displays a mock-up example of the proposed visualization.

[1] A Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classifier system as the discrimination threshold is varied. It plots the true positive rate (TPR) against the false positive rate (FPR).

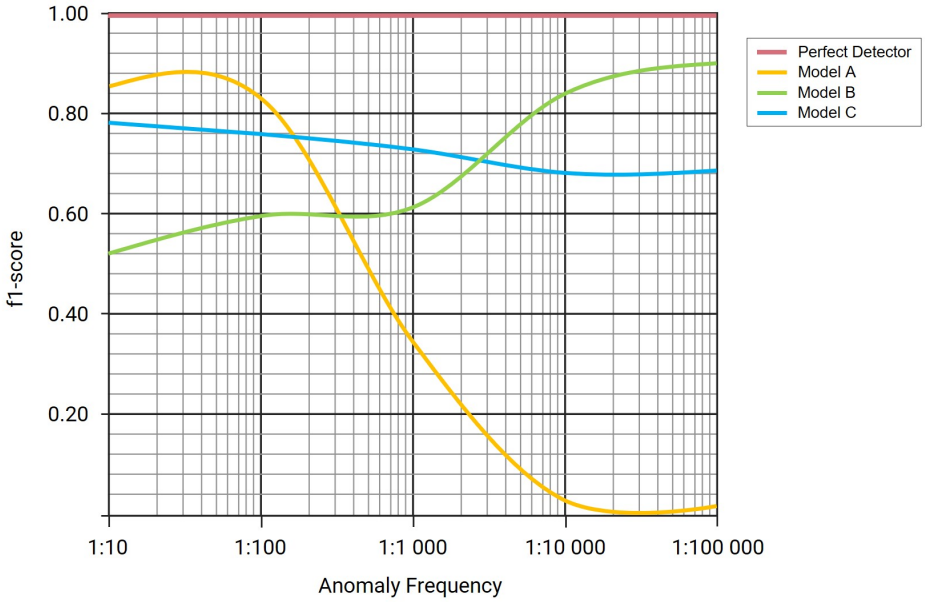

Figure 3.4: A mock-up example of the proposed x_r-*score* output

The graph plots two parameters: the x-axis represents the characteristic of interest – in our case, the frequency spectrum divided into log-scale thresholds; the y-axis represents the evaluation metric that best judges the performance of the model. We propose the *f1-score* as the default metric, as it captures both *precision* and *recall* in a single value. However, this metric could and should be replaced by *precision* or *recall* in cases where false positives or false negatives, respectively, are more costly.

By determining the area under the curve we can compute an aggregate measurement of the performance across the entire rarity spectrum – the x_r-*score*. This score should range from 0 to 1 (a perfect detector).

The plotted curves show different behaviors for three distinct dummy models: Model A is able to detect anomalies with very good results in a higher frequency spectrum, but has a steep decrease in performance for lower frequencies; Model B has the opposite behavior, with better scores in series with rarer anomalies; Model C displays a model that although not achieving outstanding performance, is able to maintain a consistent detection rate at every threshold.

Using the x_r-*score* for each model as shown in table 3.1, one is equipped with an effective metric to assess the most suitable model for a given scenario. In cases where the expected anomaly frequency is not known *a priori* for a particular context, model C emerges as the safest choice to deploy in a real-world setting (highest x_r-*score*). Moreover, one can redefine the spectrum for which the metric is calculated, centering the attention on the attribute's critical landscape. For instance, if the focus centers on extremely rare anomalies, one can configure the x_r-*score* to evaluate performance for an anomaly frequency below 1 every 500 data points (or 0.002). Using this sub-setting of the score tendered to rare anomalies, we discern that model B exhibits better performance within this particular range.

Table 3.1: Computed x_r-*scores* for the mock-up models

Model	$\mathbf{x_r}$	$\mathbf{x_{r\leqslant 1:500}}$
Model A	0.425	0.145
Model B	0.700	**0.800**
Model C	**0.729**	0.705

This example hopefully shows how this metric can be of paramount importance when choosing the best model to deploy in a particular context.

3.2.2 The general case of spectrum scores: x-score

While the previous explanation of the x_r-*score* has served as a foundation for comprehending the concept of sensitivity scores, it is crucial to recognize that this methodology's application extends well beyond its initial formulation. The beauty of this approach lies in its capacity to be generalized across various contexts.

Hence, we see *x-scores* as a class of sensitivity scores. Consider scenarios where the objective is to assess algorithmic performance concerning fluctuations in the number of anomalies or the distance until the first anomaly. By adapting the sensitivity score framework, analogous to the x_r-*score*, one can devise an x_n-*score* (for *number*) and an x_d-*score* (for *distance*).

We believe this to be a robust metric for model comparison and tuning, offering a unified yardstick for evaluating results across experiments with a spectrum of varying attribute values. As such, the introduction of the *x-score* metric represents a potentially interesting advancement in the field, providing researchers with a tool to objectively assess algorithm effectiveness within the context of varied attribute settings.

Implementation

A *python* implementation of the Monte Carlo sampling method is available as a public repository[2]. The codebase is open-source, and although the focus of this work is on anomaly rarity, an effort was made to make it as generalizable as possible. The implementation has been carefully structured in a modular fashion, facilitating the integration of supplementary approaches and extensions. For a comprehensive explanation of the implementation, please refer to Appendix A.

We have included a dedicated module in our repository to compute the *x-score* for a specific attribute. A detailed explanation of this module, along with an example, is available in Section A.3 of Appendix A.

[2] Available at https://github.com/nunobv/XTadGAN/

Chapter 4

Rarity Sensitivity Analysis

In this chapter, we present our second major contribution to the field of time series anomaly detection. Our focus is on exploring how state-of-the-art detection algorithms behave when confronted with variations in anomaly frequency. We will make use of the previously detailed Monte Carlo sampling method to conduct this study.

We begin by providing a comprehensive overview of our experimental setup. This includes a detailed description of our data sources, the architecture of our specially crafted test bench for studying rarity sensitivity using the previously proposed Monte Carlo sampling, the evaluation metrics employed, and details about our data preparation pipeline. Furthermore, we offer concise introductions to the state-of-the-art models under investigation.

Our research unfolds in two progressive steps using the established experimental framework. First, we establish a baseline rarity sensitivity analysis with the base algorithms. This initial exploration allows us to understand their behavior and establish a robust foundation for our study. Subsequently, we conduct experiments to fine-tune and recalibrate the original TadGAN formulation, pushing its capabilities to handle rarer anomaly scenarios.

We believe that this study is unique in its scope and significance. We hope that our contributions will serve to establish sensitivity analysis as a standard tool within the realm of time series anomaly detection, further advancing the field.

4.1 Experimental setup

We performed all experiments across two devices: an instance of LIACC computational resources with an Intel i7-3770K processor, 8 CPU cores (3.50GHz), 32GB RAM, and 1 Nvidia GeForce RTX 2080Ti GPU (11GB); and a machine featuring an AMD Ryzen 5 5600X processor, 6 CPU Cores (3.70 GHz), 16GB RAM, and 1 Nvidia GeForce RTX 3060Ti (8GB).

The experimental environment was built using Python 3.8 and TensorFlow 2.0. To maintain consistency and reliability in all model architectures and implementations, we have implemented

all models as primitives using Orion[1], accessible as part of MIT's Sintel project [Alnegheimish et al., 2022].

4.1.1 Data Sources

This research employs a selection of univariate time series sourced from three datasets: the UCR archive, NASA, and Numenta repositories. For simplicity, we will refer to the latter two collections as "Paper datasets" due to their inclusion in the original TadGAN paper. Section 2.5.3 provides further details on these collections.

Notwithstanding the previously discussed major criticism towards these last two data sources, they are included in this research to assess whether we can replicate the original authors' results for the anomaly detection methods used as benchmarks for our proposed new architectures. Additionally, this inclusion allows us to explore potential differences in algorithm behavior when applied to the UCR archive.

To align with the principle that real-world applicable algorithms should be trained on real-world generated data, we have excluded all synthetic series from our study. Consequently, the Yahoo datasets and the Numenta *artificialWithAnomaly* collection have been excluded from our data sources. Basic information about each dataset can be found in Appendix B.1.

In total, our sources comprise a collection of 7 datasets featuring 369 series, across a diverse landscape of anomaly properties. We believe this heterogeneous selection will help identify the strengths and limitations of each baseline model.

The locations of anomalies are known for each series. Since we are working in an unsupervised setting, this information is used solely for evaluation purposes and is never incorporated into the training process. Contrary to the experimental setup used in the TadGAN paper, and to maintain a true unsupervised approach, we will not perform a train/test split and instead train the model on the complete series.

4.1.2 Experiment Architecture: Rarity-Spectrum Test Bench

We use our novel Monte-Carlo framework to design a test bench for conducting sensitivity analysis on varying levels of anomaly rarity, as outlined in Figure 4.1.

We start by defining five different anomaly levels: 1:10, 1:100, 1:250, 1:500, and 1:1 000, which cover a realistic spectrum of anomaly frequencies for our study. We believe the 1:10 frequency serves as a practical lower limit for identifying anomalies, as frequencies higher than this become challenging to classify as "anomalies". Conversely, overwhelmingly rare anomaly frequencies below 1:1 000 present difficulties due to the limited availability of benchmark series with sufficient length to simulate such scenarios.

Although very uncommon, some series lacked any anomalous points and were thus unsuitable for our purposes. As such, to create this test bench, we applied a previous filtering step to the original 369 series to ensure that each time series had a minimum of 1 500 data points and at least

[1] https://github.com/sintel-dev/Orion

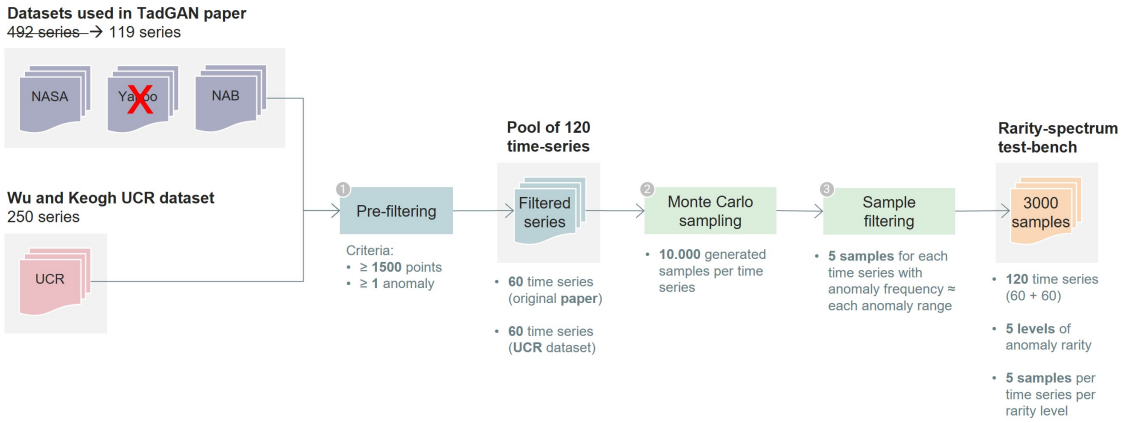

Figure 4.1: Condensed representation of the test bench creation process

one anomaly. These filters are enforced so that the remaining signals enable us to establish the desired range of anomaly frequencies.

From this filtered pool of series, we randomly selected 60 signals from the NASA and Numenta datasets, along with 60 signals from the UCR archives. For each of these signals, we used our Monte Carlo sampling technique to generate 10 000 samples per signal. The resulting samples underwent another filtering step, where we selected 5 samples for each of the 120 seed signals for each of the 5 anomaly ranges.

This process yielded 3 000 samples in total, with 600 samples for each anomaly level (300 from the UCR archives and 300 from the NASA and Numenta datasets). We chose this number to strike a balance between computational efficiency and obtaining statistically robust results.

4.1.3 Evaluation Metrics

Drawing inspiration from Hundman et al. [2018] and Geiger et al. [2020], this study employs unweighted contextual F1-scores as the chosen evaluation metric. Precision and Recall will also be presented as to give more insight into the behavior of the methods. This strategy – also known as *overlapping segment* – has been detailed in section 2.5.2.

As mentioned by the previous authors, the rationale behind this selection stems from the rarity of anomalies and the prevalence of window-based patterns in real-world scenarios. In such a context, users' primary goal is to efficiently detect true anomalies while keeping false positives to a minimum. Thus, this evaluation metric proves advantageous as it prioritizes the detection of any segment of an anomaly.

This approach for anomaly scoring is rooted in segment overlaps: a TP is registered when a known anomalous window intersects with a detected window, a FN arises when no overlap occurs between known anomalous windows and detected windows, and a FP emerges when a detected window does not align with any known anomalous region.

Additionally, when we provide summary metrics to evaluate algorithmic performance across multiple series or samples, we will employ macro-averages for the anomalous class only. This approach is chosen to offer a more accurate assessment of each algorithm's performance and ensure a fair evaluation across a diverse sample population. In situations where the F1-score is undefined due to Recall and Precision both being 0, we will set the F1-score to 0 as a default.

4.1.4 Data Preparation

Prior to feeding data to the models and beginning the training process, every series is subject to a data preparation pipeline. This process is standardized and uniformly applied to every sample across all experiments. Figure 4.2 shows a simplified overview of this process. Appendix B.2 contains a comprehensive diagram of the entire implementation pipeline.

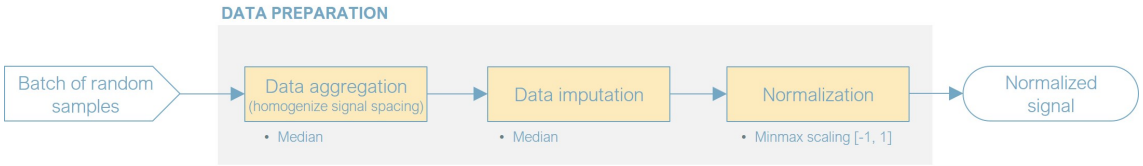

Figure 4.2: Condensed view of the *Data Preparation* pipeline

1. *Data Aggregation*. For each sample, we begin by standardizing the signal spacing to ensure uniform width across all timestamps. Most samples are already equally spaced but are nonetheless subjected to this step. The final timestamp value is determined by taking the median of the aggregated values.

   Let: S_i : original signal values at timestamp i
   T_i : timestamps associated with the original signal values S_i
   S'_i : new aggregated values for the i-th interval
   N : number of timestamps in the original signal

   Standardizing the signal spacing involves creating a new set of uniformly spaced timestamps T'_i such that:

   $$T'_i = i \cdot \Delta t \quad \text{for } i = 1, 2, \ldots, N \tag{4.1}$$

   where Δt is the uniform time interval. Then, for each right-open interval $[i \cdot \Delta t, (i+1) \cdot \Delta t[$ we compute S'_i as the median of the values of S_i whose timestamps T_i fall within this interval:

   $$S'_i = \text{median}\left(\{S_i \mid T_i \in [i \cdot \Delta t, (i+1) \cdot \Delta t[\}\right) \quad \text{for } i = 1, 2, \ldots, N \tag{4.2}$$

2. *Data Imputation*. Next, any missing values within the signal are filled using the series median value to ensure that every timestamp has a corresponding value during training.

3. *Normalization*. Finally, we normalize the data for each sample using a min-max scaling approach, fitting the values within the range of [-1, 1].

4.1.5 Baseline Algorithms

We deployed five state-of-the-art baseline and well-established unsupervised anomaly detection methods for comparison in our sensitivity study. The selection of these algorithms is based on their extensive use and prevalence in academic research. These algorithms will later serve as a benchmark for our proposed TadGAN-DT and XTadGAN algorithms explored in sections 5.1 and 5.2.

Table 4.1: Models selected for implementation

Model	Approach	Paper
ARIMA	Prediction	Yaacob et al. [2010]
LSTM Autoencoder	Reconstruction	Malhotra et al. [2015]
VAE (Variational Autoencoder)	Reconstruction	An and Cho [2015]
LSTM	Prediction	Hundman et al. [2018]
TadGAN	Reconstruction	Geiger et al. [2020]

To ensure uniformity and reliability across all model architectures and implementations, we adopted the architecture and implementation originally introduced by Geiger et al. [2020]. This decision enables us to establish direct comparisons while minimizing any potential variability arising from different architectural choices or implementation nuances.

A brief description of each method's architecture is presented below. The same training hyperparameters were used across all experiments: a batch size of 64, *Adam* optimizer, and 35 epochs of training (with the exception of TadGAN, for which a detailed explanation is given in section 4.1.5).

ARIMA

An Autoregressive Integrated Moving Average implemented with the *StatsModels* library. The hyperparameters are empirically set to $p=1$, $d=0$ and $q=0$. Point-wise prediction errors are used as the anomaly scores to detect anomalies.

The purpose of this ARIMA model was not to excel in anomaly detection, as evident from its vanilla configuration. Instead, it serves as a sanity check and baseline against which the other models were expected to outperform.

LSTM AE

The implemented LSTM auto-encoder uses two one-layer LSTM with 60 units, one for the encoder and the other for the decoder. A time-distributed layer with a dense one-unit layer is used to create the output. As with the previous method, a point-wise reconstruction error is used to detect anomalies.

LSTM VAE

For the LSTM variational auto-encoder, the encoder employs a single shared LSTM layer with 60 units and two separate dense layers, each with 60 units, to create the mean and standard deviation vectors. The decoder incorporates a repeat vector layer - an LSTM layer with 60 units - and a time-distributed layer with a dense one-unit layer for the output. The anomalies are scored using the same reconstruction error technique as described for the LSTM AE architecture.

LSTM

The LSTM neural network used in our experiments consists of two LSTM layers, each with 80 units and a dropout rate of 0.3. A final dense layer featuring a single unit is responsible for predicting the subsequent time step's value.

Following Hundman et al. [2018] work, the LSTM method uses a non-parametric Dynamic Threshold to mitigate false-positive predictions. Again, point-wise prediction errors are used for anomaly detection.

TadGAN

TadGAN comprises an encoder and a decoder, both utilizing bi-directional LSTM layers. The Generator $\mathcal{E}$ employs a one-layer bi-directional LSTM with 100 hidden units, while Generator $\mathcal{G}$ employs a two-layer bi-directional LSTM with 64 hidden units each, featuring a dropout rate of 0.3. The Discriminators (i.e. the *Critics* $\mathcal{C}_x$ and $\mathcal{C}_z$) are built with a 1D convolutional layer, designed to capture local temporal features and detect anomalous sequences.

The reconstruction error is calculated using *Dynamic Time Warping* [Berndt and Clifford, 1994]. The final anomaly scores are obtained by combining the reconstruction-based anomaly scores with the critic scores. In the original study by Geiger et al. [2020], different combinations, including *addition*, *product*, *critic-only*, and *reconstruction-only*, were explored. In our case, the *product* configuration was selected based on its superior performance as demonstrated by the same authors.

A contradictory aspect of the TadGAN paper is that while the GAN was initially trained for 70 epochs (2000 iterations in the original implementation, but updated to 70 epochs in TensorFlow 2.0), the other algorithms were trained for only 35 epochs. We aim to standardize this setup to ensure a fair comparison across all models.

4.1.5.1 TadGAN Sanity Check and Computational Load Analysis

Prior to starting our experiments, we conducted initial validation checks and tests on the NASA and Numenta datasets utilized in the original TadGAN paper. The aim was to verify the accuracy of the algorithm implementation and compare the results with those reported by the authors.

During our preliminary tests, we obtained the following average computational times (in seconds) for the 119 series:

Table 4.2: Computational load for each algorithm

Algorithm	**Epochs**	**Average time (s)**
ARIMA	35	135.9
LSTM AE	35	27.0
LSTM VAE	35	34.2
LSTM	35	50.5
TadGAN	70	999.2
TadGAN	35	511.8

There is a noticeable disparity in the computational demands between TadGAN and the other models, even when employing the same 35 epochs instead of the original 70 epochs. Generative Adversarial Networks are notoriously known for their computational load, and this is evident in the values condensed in table 4.2.

This poses a substantial challenge for our research, as our available computational resources are inadequate to complete the experiments within a reasonable timeframe. As an illustrative example, running the original 70-epoch TadGAN model on a single experiment involving our 3 000 samples would require an uninterrupted span of 833 hours, or over 34 consecutive days, of computation.

Given this limitation, we conducted an additional series of tests to evaluate TadGAN's performance across various epoch counts. The resulting outcomes for the 119 series are summarized in the table below.

Table 4.3: TadGAN performance for various training epochs

Algorithm	**Epochs**	**Average Time (s)**	**Macro-Average F1-score**	**Macro-Average Precision**	**Macro-Average Recall**
TadGAN	5	98	0.391	0.333	0.474
TadGAN	10	166	**0.607**	**0.513**	0.740
TadGAN	20	309	0.555	0.461	0.711
TadGAN	35	511	0.597	0.507	0.729
TadGAN	50	718	0.582	0.434	0.671
TadGAN	70	999	0.577	0.470	**0.741**
Original paper reported results[2]			0.612	0.548	0.700

The findings in table 4.3 reveal an unexpected insight. Despite our initial intuition, the TadGAN algorithm does not exhibit substantial performance fluctuation across epochs, for values equal to or above 10 epochs. Remarkably, the model with 10 epochs emerged as the most effective, although this variance is not markedly significant.

This observation seems to imply that the networks converge quite efficiently. Given the unexpected nature of this outcome, we undertook an additional analysis to examine the progression of the generator loss and critic scores for these models. Since this topic is tangent to our primary focus, we have included the results and a concise explanation in Appendix C.

Moreover, upon reevaluating the original raw results from the TadGAN paper and computing the macro-average, we found no substantial deviation between our outcomes and the results of the original model. In our experiments, we show this to be consistent across all individual datasets. This emphasizes the robustness of our implementation, ensuring that direct comparisons are not influenced by variations arising from different architectural choices or implementation details.

As a result of this experiment, we can confidently affirm that diminishing the training epochs from the original 70 to a smaller value does not significantly impact performance. Thus, with the specific goal of reducing the computational demands necessary for TadGAN training and enabling an expanded number of experiments, we have used 10 training epochs for all forthcoming experiments detailed in this report.

4.2 Baseline Rarity Sensitivity Analysis

To start, we sought to grasp how each selected method performed concerning anomaly rarity. To achieve this, we subjected these algorithms to our test setup comprising 3 000 samples. This study establishes a baseline analysis for future experiments and forms the bedrock for the development of more effective anomaly detection architectures for extremely rare anomalies. The complete results table is available in Appendix B.4.

To delve deeper into the findings, we've segmented this analysis into three distinct stages. Firstly, we scrutinize the results from samples exclusively generated using the TadGAN paper datasets (NASA and Numenta). Subsequently, we turn our attention to the UCR samples exclusively, and finally, we examine the combined results from both datasets.

In this section, we will only show the F1-scores plotted against anomaly rarity, alongside the rarity spectrum score (x_r-*score*) for each model. For conciseness and ease of reading, the Precision and Recall plots are provided in Appendix B.3.

[2]While the authors employed a micro-averaging approach to compute the reported results, we used the original raw results from each signal and calculated the macro-average to align with our observations. The raw results can be accessed at https://sintel.dev/Orion/user_guides/benchmarking.html.

Samples generated using Paper datasets

Focusing on the left side of figure 4.3, as anticipated, all models surpass the performance of a basic *ARIMA(1,0,0)* model, which consistently exhibits poor performance with an x_r-score of 0.357 across the entire anomaly spectrum. Notably, TadGAN stands out as the top performer within the 1:10 range but experiences a decline in effectiveness as anomaly rarity increases. In fact, for the rarest anomalies, TadGAN is outperformed by every model except ARIMA.

Interestingly, the LSTM-DT approach demonstrates the most consistent performance across the entire spectrum, boasting the highest x_r-score among all models, 0.612. This is also true for both precision and recall, which do not seem to have a very significant impact on the algorithm's ability to detect anomalies.

Finally, with the exception of the 1:10 range, there is minimal disparity between the LSTM Autoencoder and Variational Autoencoder, both yielding similar average x_r-scores in this experiment (0.549).

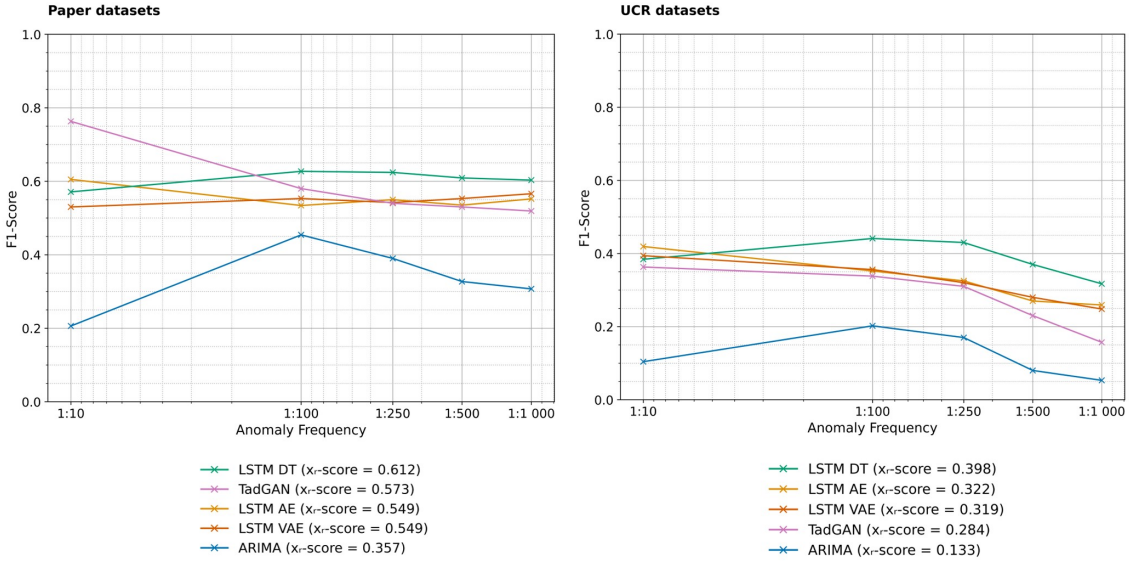

Figure 4.3: Rarity sensitivity analysis: model performance (F1-score) as a function of anomaly rarity for the Paper datasets (left) and UCR datasets (right)

Samples generated using UCR datasets

If we shift the focus to the right side of the previous figure, we can immediately notice a significant drop in performance for all models, with a particular emphasis on TadGAN in the 1:10 range. This reduction in performance closely aligns with recent results published by some of the original TadGAN authors, who applied these models to the UCR repository time series [Wong et al., 2022].

Further in-depth analysis is necessary to draw definitive conclusions from these results. Nevertheless, this discrepancy provides further evidence to the observations made by Wu and Keogh [2021] and their critiques regarding the construction of current time series anomaly detection benchmarks.

We believe that one noteworthy aspect contributing to this discrepancy is the nature of the UCR dataset, which mandates a single anomaly per series, making anomaly detection considerably more *hit or miss*. In contrast, the NASA and Numenta series lack such a restriction, often containing multiple anomalies within a single series. Due to our sampling technique, there is a higher likelihood that series in the more frequent rarity ranges consist of multiple anomalies, which is not the case for UCR-generated samples. A promising avenue for future research, building upon these results, would be to apply our Monte Carlo sampling method to explore algorithm behavior concerning the number of anomalies in fixed-length samples.

Similar to the samples generated from the paper datasets, but even more pronounced, all models face growing challenges in detecting anomalies as their rarity increases. LSTM-DT [Hundman et al., 2018] stands out as the most well-rounded approach across the entire spectrum, excelling particularly in identifying extremely rare anomalies, as the rarity spectrum scores in table 4.4 demonstrate.

All samples

When we consolidate the results across the entire sample population, a consistent trend emerges: performance diminishes notably as anomalies become rarer. Figure 4.4 illustrates this rarity-sensitivity analysis conducted on all the samples.

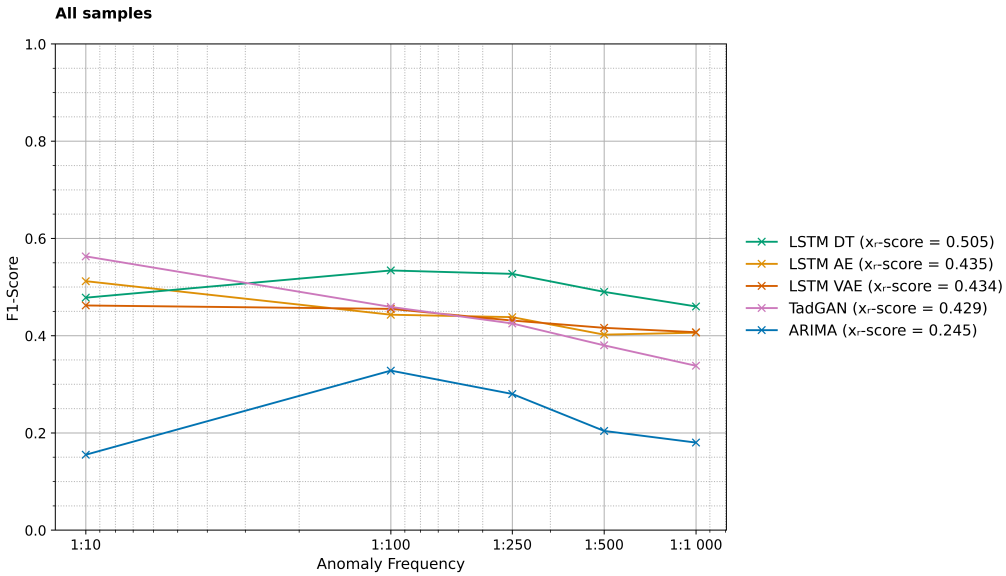

Figure 4.4: Rarity sensitivity analysis: model performance (F1-score) as a function of anomaly rarity for all samples

LSTM-DT emerges as the most balanced approach overall, demonstrating better performance even in the range of rarer anomalies. The post-processing approach used by Hundman et al. [2018]

for pruning anomalies in order to mitigate false positives is quite interesting, consistently maintaining high levels of both Recall and Precision, even when dealing with extremely rare ranges. This technique will be further explored in section 5.1 and refined in section 5.2.

Conversely, TadGAN is the model most affected by the frequency of anomalies within the samples. As detailed in section 2.3.1, after reconstructing the series, TadGAN applies a sliding window logic to calculate thresholds and identify anomalies. The parameters used are empirical (4 standard deviations and a window size equal to one-third of the total series length). The use of such a high value of σ, particularly when dealing with low anomaly frequencies, creates a significant smoothing effect, causing some anomalies to go undetected. We will delve deeper into the study of these parameters and their variations later in this work.

The behavior of TadGAN can also be attributed to the characteristics of the datasets used for its initial training and tuning. Specifically, the original NASA, Yahoo, and Numenta time series exhibit an average anomaly frequency of 6.00%, while the UCR archives have a significantly lower anomaly frequency of 0.26%. As such, one can infer that many of the architectural and parametric choices made during the development of the TadGAN model have contributed to this outcome. This further underscores the significance of the analysis and methodology proposed in this study.

Table 4.4: Baseline algorithm's x_r-scores by sample origin

Algorithm	**Paper datasets**		**UCR datasets**		**All samples**	
	$\mathbf{x_r}$	$\mathbf{x_{r\leqslant 1:500}}$	$\mathbf{x_r}$	$\mathbf{x_{r\leqslant 1:500}}$	$\mathbf{x_r}$	$\mathbf{x_{r\leqslant 1:500}}$
TadGAN	0.573	0.524	0.284	0.193	0.429	0.359
LSTM DT	**0.612**	**0.606**	**0.398**	**0.343**	**0.505**	**0.475**
LSTM AE	0.549	0.544	0.322	0.264	0.435	0.404
LSTM VAE	0.549	0.560	0.319	0.264	0.434	0.412
ARIMA	0.357	0.317	0.133	0.067	0.245	0.192

4.3 Re-calibrating TadGAN for extremely rare anomalies

Since the primary objective of this work is to develop a TadGAN variation tailored for rare anomalies, we conducted an empirical evaluation to assess how different hyperparameters influence the predictive performance of this algorithm.

Among these parameters, two stood out as particularly relevant for investigation: the number of standard deviations employed to establish the anomaly detection threshold and the rolling window size over which this threshold was applied. Both of these parameters directly influence the extent of the smoothing effect discussed earlier on the detection of anomalies by the original model, and were empirically set in the original paper.

In our preliminary experiments, the latter appeared to have minimal, if any, discernible effect on the original model's performance. In contrast, the standard deviation parameter exhibited a clear and significant impact on model performance.

4.3.1 Varying the anomaly threshold with values of σ

We wanted to gauge if, and the extent to which, we can influence TadGAN's ability to detect extremely rare anomalies by adjusting the parameters governing the detection threshold.

Our reasoning is as follows: theoretically, a lower σ value should help enhance recall, particularly in scenarios of extreme rarity, as it excludes fewer candidate anomalies. However, if this reduction is too greedy, it could result in an excessive number of false positives. Hence, our goal is to find a better trade-off between precision and recall at the more extreme range.

Figure 4.5 illustrates how altering the value of σ for the detection thresholds influences TadGAN's performance. In addition to the original TadGAN implementation with $\sigma = 4$, we have also incorporated the LSTM-DT algorithm, previously shown to be the top-performing among the baseline approaches.

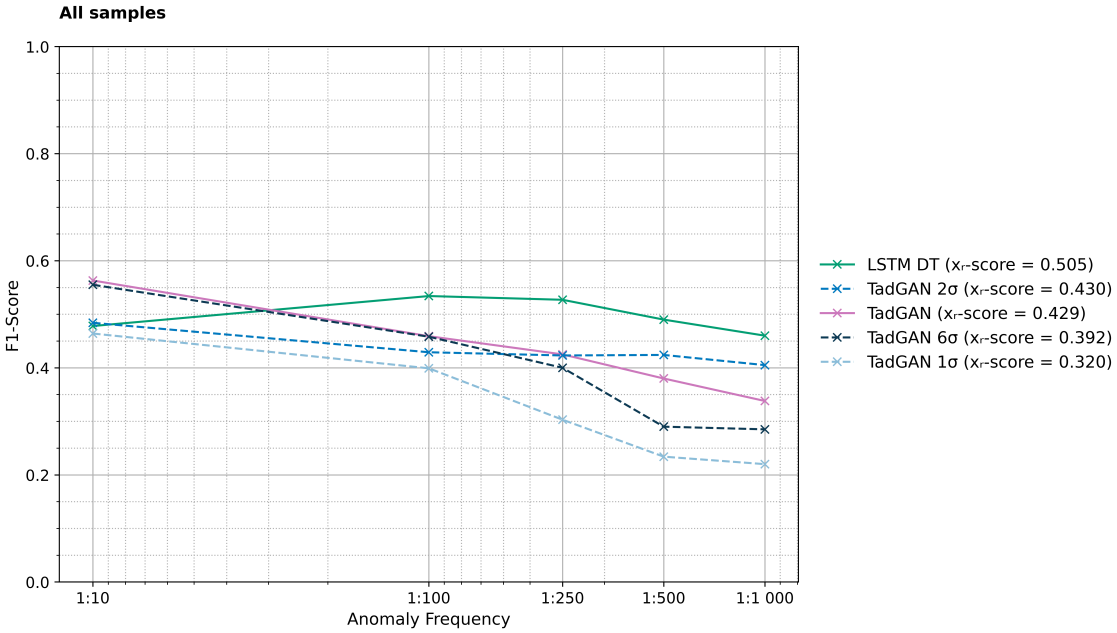

Figure 4.5: Rarity sensitivity analysis: model performance (F1-score) as a function of anomaly rarity for different values of σ (all samples)

As expected, changing to $\sigma = 2$ improves the algorithms' performance in the rarer anomaly ranges (1:500 and 1:1 000), while losing performance on the opposite side of the spectrum. The overall x_r-score does not vary significantly compared to the original TadGAN. Both models with $\sigma = 1$ (x_r-score = 0.320) and $\sigma = 6$ (x_r-score = 0.392) perform worse than the original model. LSTM-DT maintains superior performance in almost all rarity spectrum.

The difference in the previous results can be explained by the impact that the changes to the original configuration had on recall and precision. Both are summarized in figure 4.6.

As anticipated, a decrease in the σ value corresponds to an increase in recall. This is particularly evident for rare anomalies when $\sigma = 1$, reaching nearly 0.800 recall. This effect becomes more pronounced as the anomaly rarity escalates. In such cases, where there is a higher likelihood

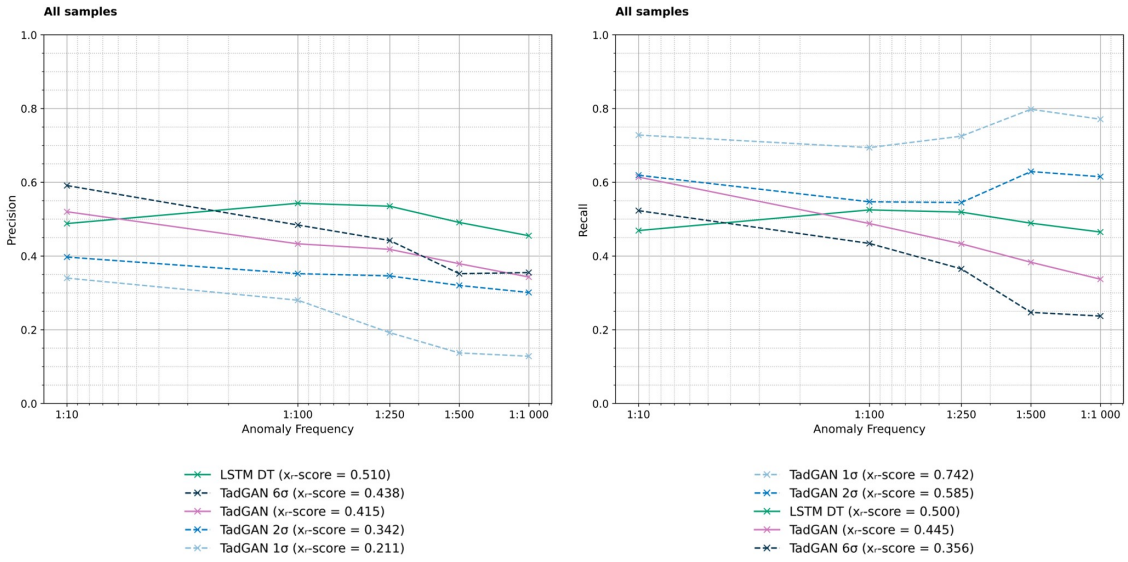

Figure 4.6: Precision (left) and Recall (right) as a function of anomaly rarity for TadGAN with varying values of σ (all samples)

of only one anomaly per sample, single anomaly detection gains greater relevance compared to the 1:10 frequency, where multiple anomalies exist per sample.

Still regarding recall, both the model with $\sigma = 1$ and the model with $\sigma = 2$ deliver impressive results, especially when compared to the original TadGAN model in samples with extremely rare anomalies. The same trend holds when compared to LSTM-DT.

Conversely, there was a decrease in precision in the models where σ was reduced, as expected. Notably, the LSTM model with Dynamic Thresholding achieves the highest precision in scenarios with rare anomalies, which further highlights the effectiveness of this pruning method in comparison to the simplified version employed by the TadGAN implementation. This achievement stands out, even though it does not excel in terms of recall.

This analysis highlights the pivotal role of precision. To enhance generative adversarial networks for contexts with extremely rare anomalies, the key lies in architectural modifications that elevate precision without notably sacrificing the recall achieved through σ value adjustments. In the next chapter, we propose two new algorithms that leverage this property.

Chapter 5

Detecting Extremely Rare Anomalies

Following the initial investigations outlined in the preceding chapter, we introduce two novel architectures tailored explicitly for detecting extremely rare anomalies. The first is TadGAN-DT (section 5.1), where we revamp the original post-processing pipeline by incorporating an anomaly identification method inspired by Hundman et al. [2018]'s Dynamic Thresholding.

Next, we present XTadGAN (section 5.2), a new approach that leverages meta-information about the expected anomaly frequency within each series. XTadGAN utilizes rarity-based dynamic thresholding and pruning techniques to improve performance in the context of extremely rare anomalies.

5.1 TadGAN-DT

The original TadGAN algorithm employs two distinct evaluators to estimate anomaly scores, as discussed in section 2.3.1. Firstly, it computes a DTW reconstruction error by measuring the mean squared error between the reconstructed series and the original series values. Additionally, it utilizes the outputs from the Critic networks as a measure of how anomalous a time segment is. These outputs are then normalized and the final anomaly scores result from a product combination of the *z-scores* of these two values using the following formula:

$$score(x) = \alpha Z_{RE}(x) \odot Z_{\mathcal{C}_x}(x) \tag{5.1}$$

After computing the anomaly scores, the model applies a parametric thresholding technique, employing a simple static threshold defined as 4 standard deviations from the mean of the window.

However, a notable issue with TadGAN's approach to anomaly detection is its assumption that the output error series follows a Gaussian distribution. Hundman et al. [2018], using the same NASA dataset, demonstrated through D'Agostino and Pearson's normality test that errors violate this assumption. Consequently, the author concluded that «the error information lost when using

Gaussian parameters results in suboptimal thresholds that negatively affect precision and recall and cannot be corrected by pruning».

Therefore, for our first method, we propose a complete overhaul of the original TadGAN post-processing pipeline. Instead, we suggest adopting an approach inspired by the Dynamic Thresholding and pruning architecture [Hundman et al., 2018], employing a non-parametric threshold.

Moreover, the decision to combine the reconstruction error with the critic score raises some debate. The reconstruction error gauges how closely the reconstructed signal resembles the original one, while the critic score can be perceived as a regularization term, preventing the series from overfitting anomalies. This serves a crucial role in achieving effective adversarial training.

However, mathematically justifying its utilization as the final anomaly score is far more challenging. To begin with, each score is distinct and has its own dimension (or scale). Furthermore, critic scores are unbounded due to the Wasserstein distance. Given the unbounded nature of the critic score, it can potentially significantly influence and dominate over the reconstruction loss. Perhaps to mitigate this impact, the authors empirically set the contribution of the reconstruction error to be ten times that of the critic scores ($\alpha = 10$). Nonetheless, even if the critic scores were bounded, it is apparent that its influence would be greater as the GAN improves its series reconstruction (implying a lower mean square error and its contribution to the overall error score).

To this end, we have also revised the scoring function used. While we still employ the product combination of both the reconstruction error and critic scores during model training to ensure that the reconstructed series does not overfit, we will rely solely on the reconstruction error to compute anomaly scores.

5.1.1 Model architecture

In light of the preceding arguments, as previously outlined, we propose a novel algorithm named TadGAN-DT. This algorithm draws inspiration from the original TadGAN but introduces significant architectural changes:

- Rather than employing the original methodology that combined reconstruction error and critic scores into a single metric for anomaly scores, we will exclusively use the reconstruction error as our anomaly score. Higher errors should suggest a higher likelihood of anomalies. This reconstruction error is computed as the point-wise mean squared error between the GAN-reconstructed series and the original series;

- We will discard the original parametric thresholding technique and, in its place, implement a non-parametric dynamic thresholding approach influenced by the LSTM-DT architecture, which has consistently demonstrated robust performance across our experiments.

A schematic view of the main differences between TadGAN and TadGAN-DT can be consulted in Appendix B.6. The algorithm for this architecture is detailed in Appendix A.5. Due to their importance in the new architecture, we will briefly detail the two stages of the post-processing pipeline of TadGAN-DT: *dynamic thresholding* and *pruning*.

5.1.1.1 Dynamic Thresholding

The dynamic thresholding technique, as proposed by Hundman et al. [2018], involves two primary steps:

1. *Error computation and smoothing*. The reconstruction error e_r is computed as a one-dimensional vector based on the point-wise difference between the original and reconstructed values for each timestamp i, $e_r^{(i)} = |y^{(i)} - \hat{y}^{(i)}|$:

$$e_r = [e_r^{(i-w)}, \ldots, e_r^{(i-1)}, e_r^{(i)}] \tag{5.2}$$

where w is the width of the context window used to evaluate errors at timestamp i. Subsequently, this error vector undergoes smoothing using an exponentially-weighted average (EWMA) over the same w previous error values, resulting in an array of smoothed errors e_s.

2. *Threshold calculation and anomaly scoring*. This step seeks to identify an optimal threshold value such that removing all values above it results in the most significant percentage decrease in the mean and standard deviation of the smoothed errors (e_s). The threshold t is selected from the set T:

$$T = \mu(e_s) + z\sigma(e_s) \tag{5.3}$$

where the t is determined by:

$$t = argmax(T) = \frac{\Delta\mu(e_s)/\mu(e_s) + \Delta\sigma(e_s)/\sigma(e_s)}{|e_a| + |E_{seq}|^2} \tag{5.4}$$

$$\begin{aligned} \text{where:} \quad \Delta\mu(e_s) &= \mu(e_s) - \mu(e \in e_s | e < t) \\ \Delta\sigma(e_s) &= \sigma(e_s) - \sigma(e \in e_s | e < t) \\ e_a &= e \in e_s | e < t \\ E_{seq} &= \text{continuous sequences of } e \in e_a \end{aligned}$$

This optimization process explores a range of potential standard deviation values, specifically values of z between 2 and 10, which contrasts with the fixed approach with $\sigma = 4$ of TadGAN. Once this threshold is determined, the normalized score for the highest smoothed error in each sequence of anomalous errors is calculated based on its distance from the selected threshold:

$$score^{(i)} = \frac{max(e_s^i) - t}{\mu(e_s) + \sigma(e_s)} \tag{5.5}$$

5.1.1.2 Pruning

To reduce false positives, a pruning procedure is introduced, derived from the previous pool of smoothed errors that passed the initial threshold (e_t). Here is how it works:

- The e_t vector is sorted in descending order, to which the maximum smoothed error that is not anomalous is also included.
- We step through this sequence of errors incrementally and calculate the percentage decrease ($d^{(i)}$) between consecutive instances, as shown in the equation below:

$$d^{(i)} = \frac{e_t^{(i-1)} - e_t^{(i)}}{e_t^{(i-1)}} \tag{5.6}$$

- At each step i, if a minimum percentage decrease p is exceeded by $d^{(i)}$, all previous candidate anomaly sequences remain classified as anomalies. If this is not the case, that and all the next smoothed error sequences are reclassified as belonging to the normal class.

The value for p is set at 0.13, in line with the original LSTM-DT implementation, whereas a value of 0.10 was used for the TadGAN algorithm.

5.1.2 Results and Discussion

We subjected this new method to the same experimental setup as described in Chapter 4. Figure 5.1 presents the results of this experiment. This new architecture significantly improves both the original model and the base model with $\sigma = 2$ in contexts of rare anomalies, although being less effective in cases with frequent anomalies.

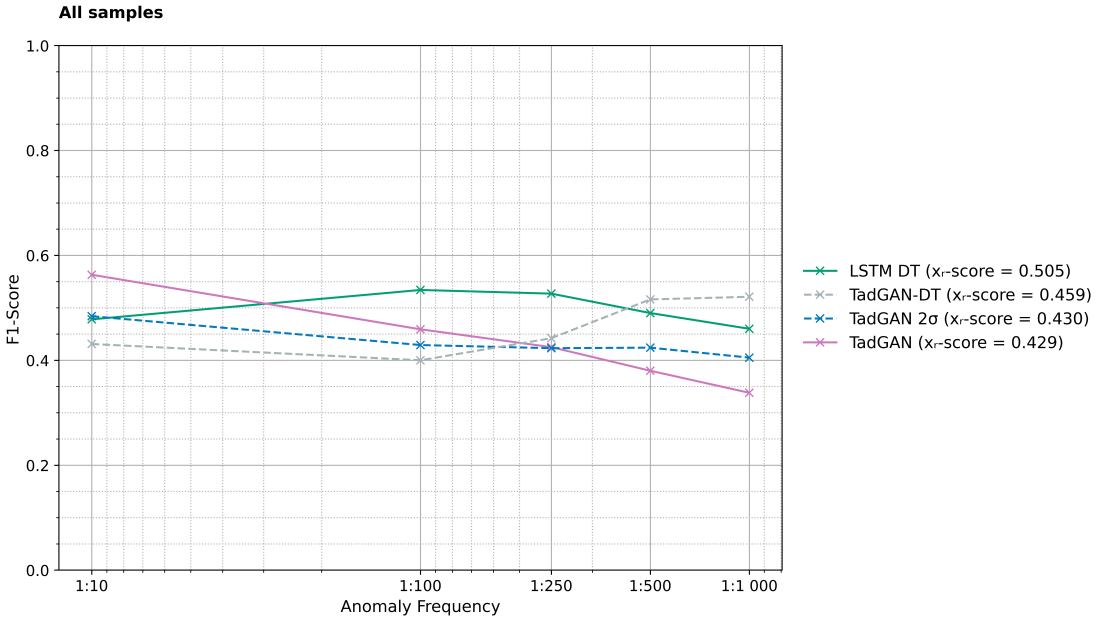

Figure 5.1: Rarity sensitivity analysis: TadGAN-DT performance (F1-score) as a function of anomaly rarity (all samples)

This occurs because, as the rarity of anomalies increases, the number of anomalies tends to approach 1. This mitigates some of the negative impacts expected on recall due to the more aggressive pruning, while bolstering the model's precision, as intended. To gain a deeper understanding of this effect, we can refer to Figure 5.2.

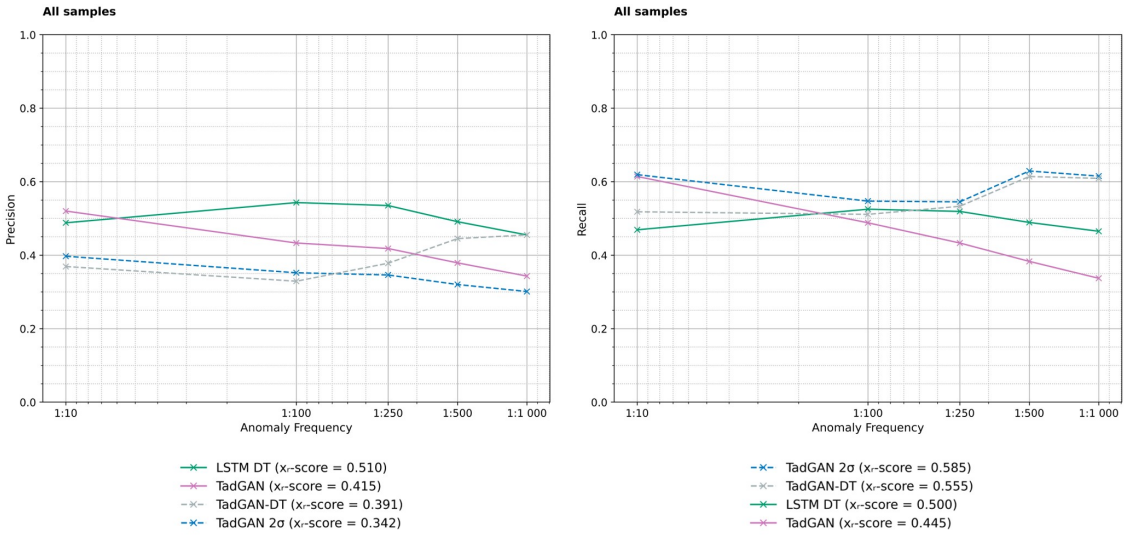

Figure 5.2: Precision (left) and Recall (right) for TadGAN-DT as a function of anomaly rarity (all samples)

Recall (on the right) is not severely affected by this change, especially when compared to the base model with $\sigma = 2$. Any decrease in recall is primarily noticeable in the higher anomaly frequency ranges, as explained previously. In contrast, precision (on the left) experiences significant improvements in the extremely rare anomaly ranges, converging with the results obtained with LSTM-DT as anomaly frequency decreases.

The initial goal of improving precision without significantly compromising recall has been successfully achieved. The algorithm developed outperforms all the other evaluated models in contexts of rare anomalies, and represents a direct evolution of the original TadGAN architecture.

5.2 XTadGAN

While the previous algorithm yielded very interesting results, we sought to push a bit further and explore a different approach, one grounded in real-world applicability.

Here is our premise: given that we are dealing with extremely rare anomaly contexts, we can leverage this knowledge to adjust the previous architecture and influence detection. In practical situations, real-world data often provides information about the expected anomaly frequency in advance. This information can be estimated using various methods, such as Mean Time Between Failures (MTBF) for industrial equipment, predicted lifespan cycles, component failure rates, or Failure In Time (FIT) in the context of electronics, among others.

Therefore, we can use the expected anomaly frequency for a specific sample or series as a *meta-parameter* to condition the detection and pruning of anomalies.

5.2.1 Model architecture

Building upon the previous TadGAN-DT architecture, we introduced significant changes. A visual representation of the updated architecture is also available in Appendix B.6. The complete algorithm for this architecture is presented in Appendix A.6.

- In the *smoothing* and *dynamic-thresholding* stages, the algorithm creates rarity-adjusted contextual intervals for the EWMA and thresholding calculation, respectively. The size of this contextual window is defined by the parameter ν, which defines the expected anomaly frequency for a given time series. For instance, if we anticipate an anomaly every 1 000 data points (an anomaly frequency of 0.001), the contextual window size is adjusted to $1/\nu = 1\,000$ data points – hence the term "rarity-adjusted".

- Instead of using a static value of p to prune candidate anomalies, we employ a dynamic approach, using the following expression:

$$p = p_0 \times e^{(1-\nu\cdot\Delta t)} \tag{5.7}$$

where: p : threshold used to classify subsequent sequences as normal
p_0 : base value for the parameter p
ν : expected anomaly frequency
Δt : distance between candidate anomalies

In addition to calculating the decrease in error among candidate sequences, this equation factors in the distance between previously identified anomalies through an exponential function: $e^{(1-\nu\cdot\Delta t)}$. This expression conditions the number of anomalies identified within each contextual window, approximating, but not forcing, the presence of $1/\nu$ anomalies per window. Figure 5.3 shows the range of values for this parameter as a function of anomaly distance.

Essentially, if two anomalies appear closely together, it becomes exponentially harder for the second one to be recognized as an anomaly. Conversely, the threshold decreases exponentially if there is a significant time gap between identified anomalies, allowing lower-scoring candidate anomalies to be classified as anomalous.

We have set the initial value of p_0 to 0.20, higher than the 0.13 used in the standard DT pruning method, to enforce a smaller number of anomalies identified in each contextual window.

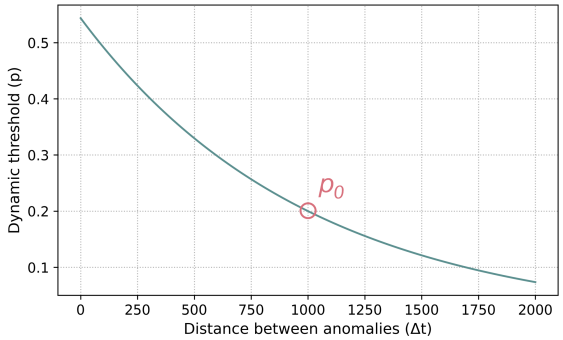

Figure 5.3: Values for p as a function of anomaly distance (Δt)

5.2.2 Results and Discussion

Figure 5.4 shows the results of this novel architecture, compared to the other top-performing methods and the baseline TadGAN algorithm.

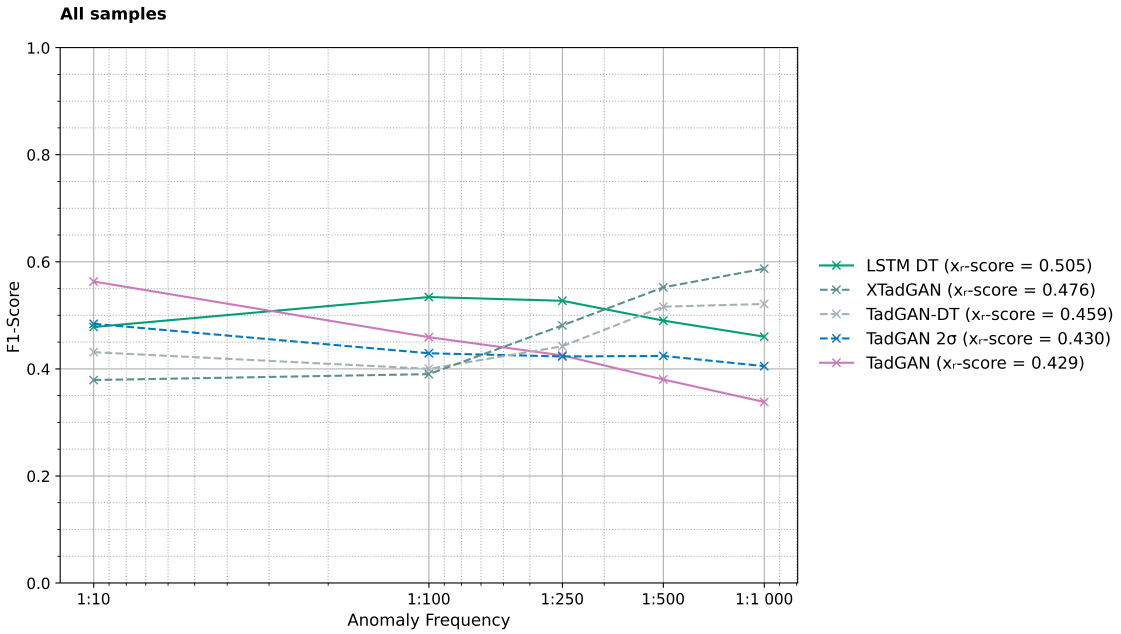

Figure 5.4: Rarity sensitivity analysis: XTadGAN performance (F1-score) as a function of anomaly rarity (all samples)

The algorithm demonstrates a substantial performance boost in the context of rare anomalies, outperforming the previous results achieved using Dynamic Thresholding. However, in scenarios with more frequent anomalies, its performance declines due to an elevated false negative rate, a consequence of the more aggressive pruning.

The higher p value coupled with the rarity-based contextual windows does not seem as effective in the higher frequency range as it is with rare anomalies. In cases where anomalies are

not uniformly distributed across the time series (which applies to most samples), this architecture results in fewer identified anomalies, which has a significant impact on recall.

As displayed in figure 5.5, XTadGAN's recall markedly decreases in comparison to the TadGAN-2σ model and TadGAN-DT. This drop is more pronounced in ranges with more frequent anomalies, as previously stated. The decrease arises from the rarity-based dynamic pruning, whose effect is more pronounced when there is a higher number of anomalies or when their proportion is greater.

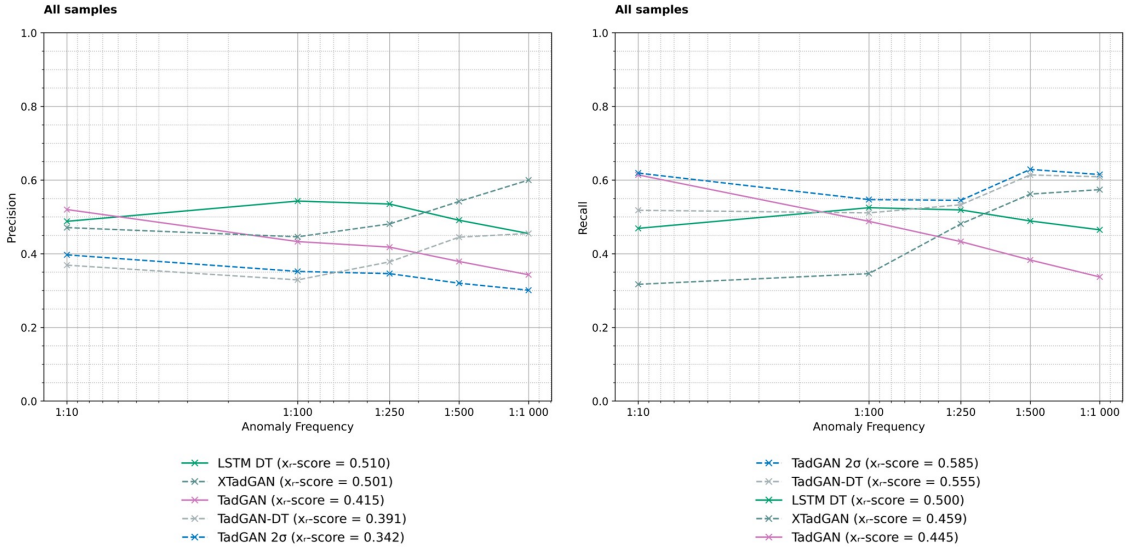

Figure 5.5: Precision (left) and Recall (right) for XTadGAN as a function of anomaly rarity (all samples)

However, the significant improvement in XTadGAN's performance, especially in contexts of rare anomalies, is explained by its heightened precision. As evident in the left-side plot of figure 5.5, the new architecture achieves the highest precision value among all developed methods, surpassing even the LSTM-DT algorithm. Table 5.1 condenses these results using the rarity-spectrum score.

Table 5.1: Rarity-spectrum scores for the top performing algorithms, across the entire anomaly spectrum (x_r) and for extremely rare anomalies ($x_{r\leqslant 1:500}$)

Algorithm	$\mathbf{x_r}$	$\mathbf{x_{r\leqslant 1:500}}$
LSTM DT	**0.505**	0.475
TadGAN	0.429	0.359
TadGAN-2σ	0.430	0.414
TadGAN-DT	0.459	0.518
XTadGAN	0.476	**0.570**

We have also performed an independent signed-rank Bayesian analysis[1] to compare the performance of each architecture [Benavoli et al., 2017]. This test computes three probabilities: the probability that the first method has higher scores than the second, the probability that differences are within a region of practical equivalence (set to 2.5%), or the probability that the second method has higher scores. The results are shown in table 5.2.

Table 5.2: Probabilities for all comparisons between algorithms using Bayesian signed-rank, for extremely rare anomalies ($\leqslant$1:500).

Algorithm A	**Algorithm B**	$p_{A>B}$	$p_{A\approx B}$	$p_{A<B}$
XTadGAN	TadGAN-DT	**0.871**	0.129	0.000
XTadGAN	TadGAN	**0.965**	0.035	0.000
XTadGAN	LSTM DT	**0.964**	0.036	0.000
TadGAN-DT	TadGAN	**0.963**	0.037	0.000
TadGAN-DT	LSTM DT	**0.875**	0.125	0.000
TadGAN	LSTM DT	0.000	0.036	**0.964**

These results underscore the effectiveness of the deliberate design choices in constructing this new architecture, particularly the use of the expected anomaly frequency as a *meta-parameter* and the subsequent dynamic rarity-based threshold to influence the anomaly identification process.

As a closing remark, we are confident that our newly proposed methods have room for further refinement and that there is potential to develop even more advanced architectures based on these findings. This underscores the significance of conducting sensitivity analyses as the one proposed, providing valuable insights into algorithms' behavior and limitations. It serves as a foundation for constructing superior, more robust, and attribute-agnostic algorithms.

[1] Bayesian analysis involves the use of Bayesian statistics and probability theory to estimate and compare the performance of multiple classifiers based on available data. It is used as a more robust tool compared to traditional null hypothesis significance testing.

Chapter 6

Conclusions and Future Work

In this study, we addressed the challenging task of detecting extremely rare anomalies in time series data using Generative Adversarial Networks. Our main objective was to develop robust solutions tailored to scenarios where anomalies are exceptionally scarce, pushing the boundaries of existing anomaly detection algorithms.

The foremost contribution of this research lies in the introduction of two novel GAN-based architectures designed specifically to handle the detection of rare anomalies. First, TadGAN-DT, an evolution of the original TadGAN formulation. This architecture integrates non-parametric dynamic thresholding and pruning techniques, enhancing the precision and reliability of anomaly detection in extreme rarity scenarios. This new approach significantly improves the original TadGAN model in contexts of rare anomalies, although being less effective in cases with frequent anomalies.

Second, XTadGAN leverages meta-information regarding expected anomaly frequencies within time series data. This innovation resulted in rarity-based dynamic thresholding and pruning techniques, further improving the model's performance in detecting extremely rare anomalies. Notably, this architecture outperforms all other tested or developed models in the rare anomaly detection context.

Another significant contribution of this research is the development of a comprehensive framework for evaluating anomaly detection models. This framework allows for systematic assessments across a spectrum of time series attributes, particularly with respect to varying levels of anomaly rarity. We introduced a standardized test bench, leveraging a newly proposed Monte Carlo sampling method from pre-existing benchmark datasets. This approach enabled systematic model assessments across diverse controlled scenarios, fostering a deeper understanding of their strengths and limitations.

To facilitate objective and reliable model comparisons, we also introduced a sensitivity index, *x-score*. This metric addresses a critical gap in current research, providing a quantitative measure to evaluate the performance of different anomaly detection algorithms across a spectrum of

attributes, particularly various anomaly frequencies.

Our research opens the door to several promising avenues for future exploration. One of the limitations of our work is its exclusive focus on univariate time series data. Future work in this area could explore multivariate settings, offering ample opportunities for innovation and improvement in the realm of time series analysis.

As demonstrated in our study, current adversarial models exhibit very slow training times and substantial computational demands when compared to alternative approaches. These characteristics may present deployment challenges in real-world applications that require lightweight and resource-efficient models. Future research could delve into techniques aimed at accelerating the training process of these models while preserving the advantages conferred by adversarial training.

Expanding and enhancing our sensitivity analysis framework offers potential avenues for further investigation. Firstly, increasing the number of samples across a wider range of anomaly levels would bolster the robustness and comprehensiveness of our evaluations. Secondly, building upon our Monte Carlo sampling method, future research could extend its application to explore algorithm behavior in different scenarios. One promising avenue could involve investigating the impact of varying the number of anomalies in fixed-length samples.

In addition to our current rarity-sensitivity experiment, a valuable extension would involve quantifying the impact of anomaly rarity on model performance. Rather than solely examining the behavior of various algorithms concerning anomaly rarity, we can explore how changes in anomaly frequency affect model outcomes. By subjecting models trained on specific rarity values to samples with different anomaly frequencies, we could quantify how sensitive a particular model is to abrupt shifts in real-world conditions, essentially uncovering the "shadow price" of rarity.

In closing, this work has contributed to the field of time series anomaly detection using Generative Adversarial Networks. We have introduced novel architectures, developed a comprehensive framework for sensitivity analysis, and provided valuable insights into the detection of extremely rare anomalies. It is our hope that this work will guide and inspire future research, ultimately leading to improved anomaly detection algorithms and their practical application in various domains.

Appendix A

Implementation Details

In this Appendix, we provide detailed explanations of the various methods, models, and architectures that have been developed throughout the course of this research. Here, we provide comprehensive descriptions of how these methods operate, their implementation details, and their potential utility for the broader time series anomaly detection community.

A.1 Monte Carlo Sampling Implementation

A.1.1 Sampling

At its core, the implementation starts with a base time series (or a set of time series). From this input, the method generates a user-specified number of samples, each capturing distinct combinations of attributes. This is achieved by introducing random cuts to the time series. The minimum and maximum lengths of the output samples can be specified by the user (either in absolute terms or as a percentage), and there is also a parameter to specify if the output should have a random length between the previous two values or a fixed length (equal to the minimum length). Users can call the *sample_time_series()* function to run the Monte Carlo sampling. An illustrative output is shown in figure A.1.

A.1.2 Characterizing the samples

After generating the pool of trimmed samples, the user can an array of pertinent attributes that serve to describe the characteristics of each sample by running the function *describe_samples()*. These attributes encapsulate different aspects of the time series data and its anomalies, providing valuable insights for subsequent experimentation and algorithm evaluation. As of the time of this writing, this method allows for the calculation of over 19 attributes, such as anomaly frequency, number of anomalies, distance to first anomaly and anomaly z-score. The full list of available attributes is condensed in table A.1 in Appendix A.2. Due to the modularity of the implementation, users can easily add additional attributes to be appended to the existing ones.

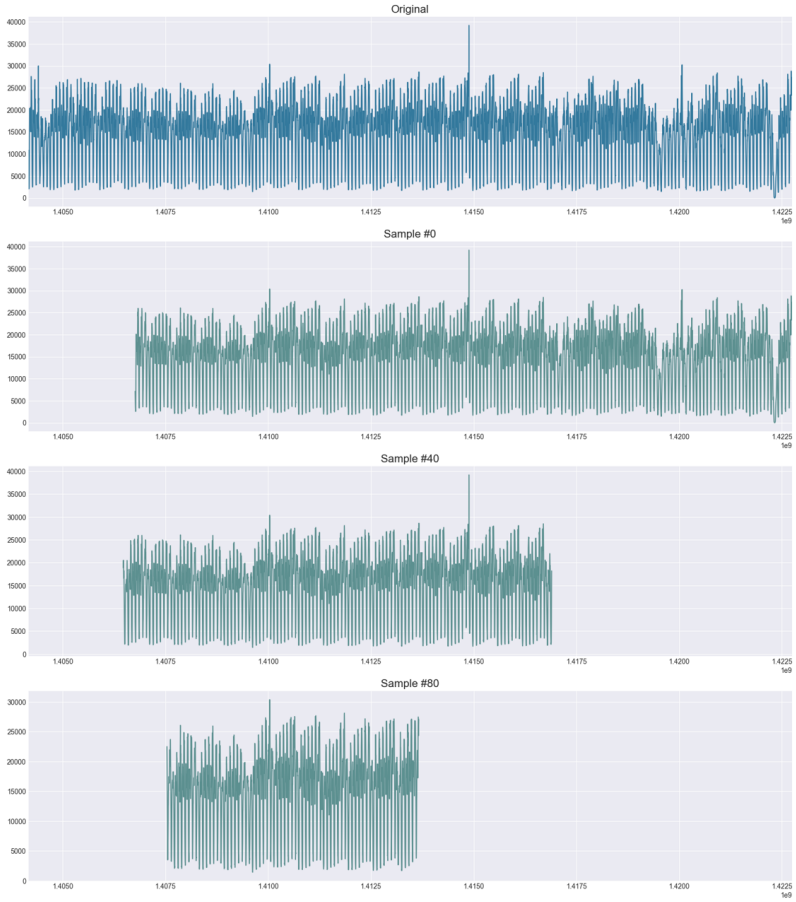

Figure A.1: Monte Carlo sampling example: original time series with 3 output samples

An auxiliary function *concatenate_labels()* has been provided to merge known anomalies with the original time series. This feature enables users working in an unsupervised frame to access some attributes that are otherwise impossible to calculate – particularly those associated with the anomalous class. It is important to note that these labels are solely intended for describing the generated samples and should be discarded during the model training process.

A.1.3 Analysing the entire sampling population

To see the condensed properties of the entire sample population, one can call two distinct functions: *describe_population()* and *plot_population_attributes()*. Upon calling *describe_population()*, a summary of the attribute distributions across the sample population is obtained. This function aggregates the essential statistical information to provide a compact representation of the samples attributes.

In tandem, the *plot_population_attributes()* function visualizes the distribution of each computed attribute using individual histograms. By offering a graphical depiction of the attribute

distributions, this function enables researchers to validate the outcome of the attribute calculations. Furthermore, it provides a visual cue to uncover the underlying patterns that govern the entire population of generated samples.

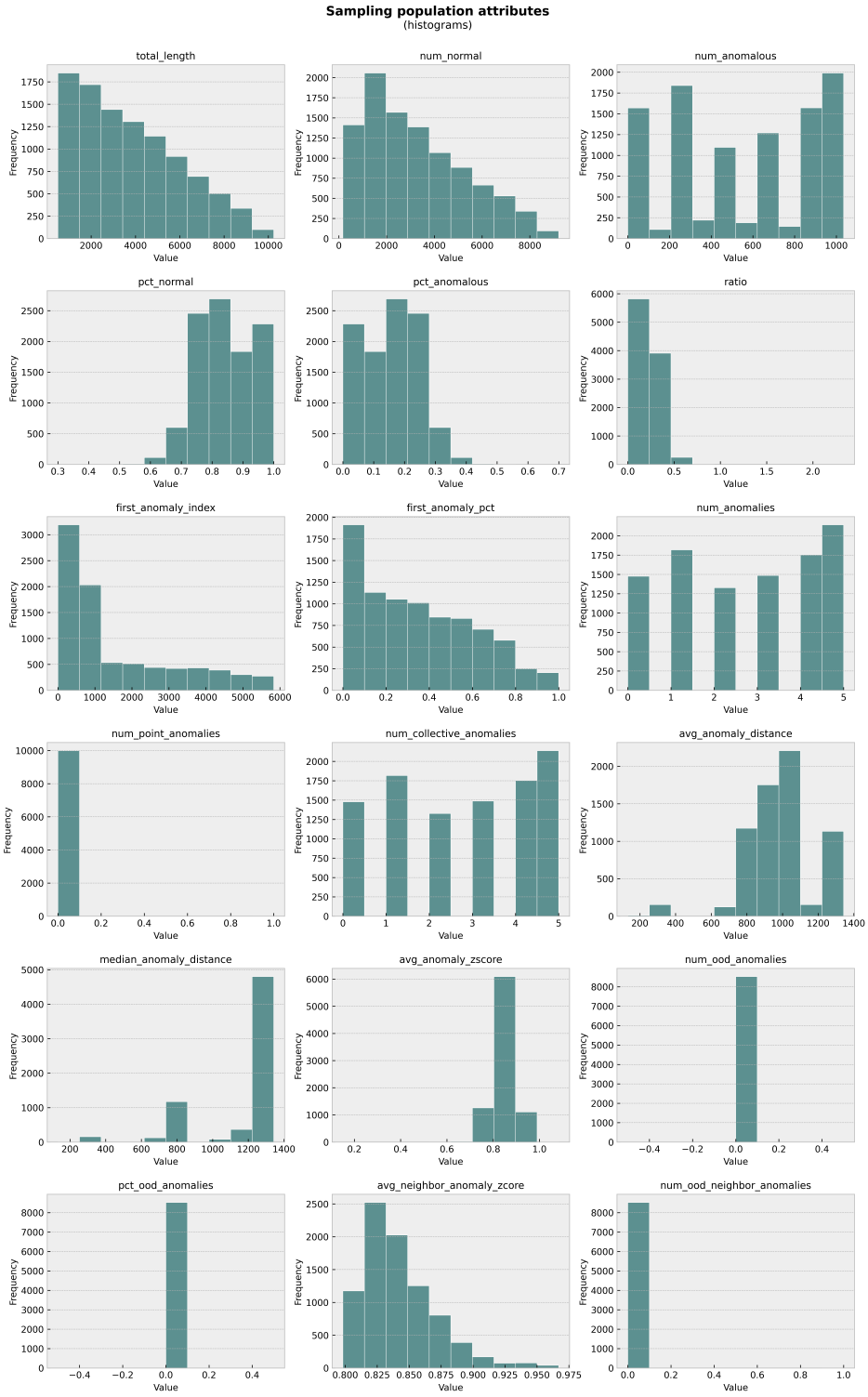

Figure A.2: Full output of the *plot_population_attributes()* function.

Figure A.2 shows an example output of the above procedure for 10.000 generated samples using random lengths. Beyond validation, these plotted attribute histograms also serve as a diagnostic tool to assess the coverage of desired scenarios within the generated samples. Researchers can gauge whether the distribution of attributes aligns with their experimental objectives, ensuring that the method has effectively produced a diverse spectrum of scenarios.

In essence, the combined usage of the previous two functions enhances the understanding of the attribute landscape across the generated sample population. This insightful analysis aids researchers in comprehending the characteristics of the experiment's foundation and in verifying the method's efficacy in capturing a broad array of relevant scenarios.

A.1.4 Filtering

To facilitate experimentation customization, the implementation includes a filtering mechanism that enables researchers to curate subsets of generated samples based on chosen attribute thresholds. This filtering process, *filter_samples()*, effectively assembles controlled test benches for algorithm evaluation. For instance, researchers can filter for samples where anomaly frequency falls within a designated range, creating a series of experiments that explore algorithm performance as anomalies become increasingly rarer.

This adaptability caters to various research contexts, enabling the exploration of attribute combinations relevant to specific anomaly detection scenarios. As we describe in Chapter 4, this method serves as the foundation for constructing our experimental setup.

A.2 Attributes computed by the Monte Carlo sampling *python* implementation

Table A.1: List of attributes currently available for the *python* implementation of the Monte Carlo sampling method for time series.

Attribute	**Description**
length	the total length of the time series
# normal points	the number of normal (0) class data points in the time series
# anomalous points	the number of anomalous (1) class data points in the time series
normal class pct.	the percentage of normal (0) class data points in the time series
anomalous class pct.	the percentage of anomalous (1) class data points in the time series
anomaly ratio	the ratio of anomalous class data points to normal class data points
first anomaly index	the index of the first occurrence of an anomaly (distance to the first anomaly)
first anomaly pct.	the first occurrence of an anomaly as a percentage of the total sample length
number of anomalies	the total number of anomaly clusters (single anomalous data points count as a cluster as well)
# point anomalies	the number of point anomalies (i.e., isolated anomalous data points)
# collective anomalies	the number of collective anomalies (i.e., total number of anomalies - number of point anomalies)
mean anomaly distance	the average distance (measured in number of data points) between the start of each anomaly
median anomaly distance	the median distance (measured in number of data points) between the start of each anomaly
avg anomaly z-score	the average distance (measured in number of standard deviations) the anomalous points are from the mean of the series
# *ood* anomalies	number of out-of-distribution anomalous points, as defined by the out-of-distribution criteria set by the user
pct *ood* anomalies	percentage of out-of-distribution anomalies from the total number of anomalous data points
avg neighbor anomaly z-score	the average distance (measured in number of standard deviations) the anomalous points are from the mean of the specified neighborhood
# *ood* neighbor anomalies	number of out-of-distribution anomalous points in the specified neighborhood, as defined by the "ood_criteria" set by the user
pct *ood* neighbor anomalies	percentage of out-of-distribution anomalies in the specified neighborhood from the total neighborhood length

A.3 Sensitivity Score Implementation

As an integral component of this work's public repository, we have included a dedicated *python* module that provides the means to compute the *x-score* for a specific attribute.

Additionally, the module includes functionality to generate graphical representations similar to those previously presented, showcasing the performance of different models across the entire attribute spectrum. Figure A.3 shows the output visualization provided by our implementation.

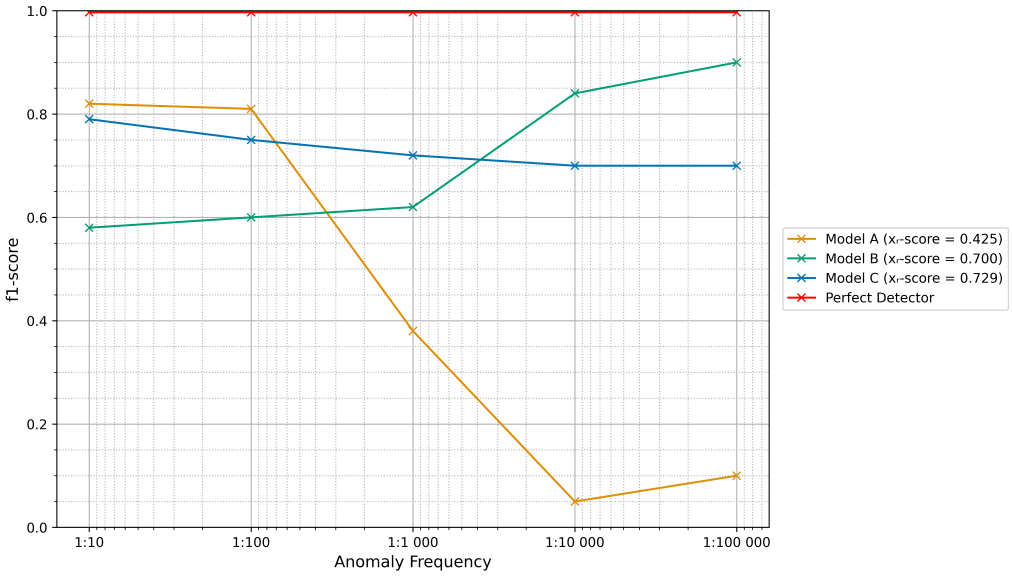

Figure A.3: Output visualization generated by our *python* implementation

This integration equips users with a comprehensive toolkit, enabling them to both quantify and visualize algorithm behavior systematically under a range of attribute conditions.

A.4 Pseudocode for the TadGAN algorithm

Algorithm 1: Overview of the TadGAN algorithm, with thresholding and pruning

With : m, batch size
$epoch$, number of iterations over the data
n_{critic}, number of iterations of the critic per epoch
n, time series length

// Training
foreach $epoch$ **do**
  for $k = 0, ..., n_{critic}$ **do**
    Generate m samples from real data;
    Generate m samples from random data following a standard normal distribution;
    Compute the gradient of the loss function for the critic network C_x;
    Update the critic C_x parameters using the previous gradient (*adam* optimizer);
    Compute the gradient of the loss function for the critic network C_z;
    Update the critic C_z parameters using the previous gradient (*adam* optimizer);
  end
  Generate m new samples from real data and random data;
  Compute the gradient of the Encoder-Decoder networks $(\mathcal{E}, \mathcal{G})$ using the GAN loss and the reconstruction loss;
  Update the Encoder-Decoder networks parameters using the previous gradient;
end

// Anomaly scoring
for $i = 0, ..., n$ **do**
  Obtain a reconstructed data point $\hat{x}_i = \mathcal{G}(\mathcal{E}(x_i))$ by encoding and decoding the original data point x_i;
  Calculate the reconstruction error (RE) between x_i and $\hat{x}_i$ using DTW;
  Calculate the mean and standard deviation of $RE(x)$ and the critic score $\mathcal{C}_x(\hat{x}_i)$ and calculate their respective z-scores $Z_{RE(x)}$ and $Z_{\mathcal{C}_x(\hat{x}_i)}$;
  Compute the final anomaly score as a product combination of the previous values;
end

// Anomaly identification (Thresholding)
Initialize an empty list to store anomalous sequences, a_{seq};
for $i = 0, ..., n - windowsize,$ **do**
  Define a sliding window starting at i and ending at $i + windowsize$;
  Calculate the mean and standard deviation of the window scores;
  Define a static threshold as 4 standard deviations from the mean;
  if $score > threshold$ **then**
    Add the identified anomalous time points within the window to a_{seq};
end

// Pruning
Sort anomalous sequences in a_{seq} by decreasing maximum anomaly score;
foreach *sequence* **do**
  Calculate the percent change p_d between the current and previous sequence scores;
  if $p_d < p_{threshold}$ **then**
    Reclassify subsequent sequences as normal;
    break;
end

A.5 Pseudocode for the TadGAN-DT algorithm

Algorithm 2: Overview of the TadGAN-DT algorithm error computation, dynamic thresholding and pruning methods

With : n, time series length
h, number of historical error values

// Error computation
for $i = 0, ..., n$ **do**
  Obtain a reconstructed data point $\hat{x}_i = \mathcal{G}(\mathcal{E}(x_i))$ by encoding and decoding the original data point x_i;
  Calculate the reconstruction error (RE) between x_i and $\hat{x}_i$ using point-wise difference;
  Compute the exponentially-weighted moving average using h previous error values;
  Add the resulting value to a vector of smoothed errors, e_s;
end

// Dynamic Thresholding and Scoring
Initialize a list T to store threshold candidates;
Calculate the mean (μ_{es}) and standard deviation (σ_{es}) of the smoothed errors;
for $z = 2, ..., 10$ **do**
  Calculate the threshold using the formula $t = \mu_{es} + z * \sigma_{es}$;
  Create a list e_{below} of errors below threshold t;
  Calculate their mean ($\mu_{e_{below}}$) and standard deviation ($\sigma_{e_{below}}$);
  Calculate the difference between these values and the the values of the smoothed errors e_s;
  Compute the objective function value for the threshold t;
  Store the threshold t and the output value in the T;
end
From T, find the threshold t_{max} that maximizes output value;
Compute a normalized score for each sequence of anomalous errors (a_{seq}) based on its distance from t_{max};

// Pruning
Sort anomalous sequences in a_{seq} by decreasing maximum anomaly score;
foreach *sequence* **do**
  Calculate the percent change p_d between the current and previous sequence scores;
  if $p_d < p_{threshold}$ **then**
    Reclassify subsequent sequences as normal;
    break;
end

A.6 Pseudocode for the XTadGAN algorithm

Algorithm 3: Overview of the XTadGAN algorithm contextual error computation, dynamic thresholding and rarity-based pruning methods

With : n, time series length
ν, expected anomaly frequency
p_0, base value for the percent-change threshold

`// Contextual Error computation`

for $i = 0, ..., n$ **do**
  Obtain a reconstructed data point $\hat{x}_i = \mathcal{G}(\mathcal{E}(x_i))$ by encoding and decoding the original data point x_i;
  Calculate the reconstruction error (RE) between x_i and $\hat{x}_i$ using point-wise difference;
  Compute the exponentially-weighted moving average using $1/\nu$ previous error values;
  Add the resulting value to a vector of smoothed errors, e_s;
end

`// Dynamic Thresholding and Scoring`

Initialize a list T to store threshold candidates;
Calculate the mean (μ_{es}) and standard deviation (σ_{es}) of the smoothed errors;
for $z = 2, ..., 4$ **do**
  Calculate the threshold using the formula $t = \mu_{es} + z * \sigma_{es}$;
  Create a list e_{below} of errors below threshold t;
  Calculate their mean ($\mu_{e_{below}}$) and standard deviation ($\sigma_{e_{below}}$);
  Calculate the difference between these values and the the values of the smoothed errors e_s;
  Compute the objective function value for the threshold t;
  Store the threshold t and the output value in the T;
end
From T, find the threshold t_{max} that maximizes output value;
Compute a normalized score for each sequence of candidate anomalous errors (a_{seq}) based on its distance from t_{max};

`// Rarity-based Pruning`

Create an auxiliary array of timestamps for each anomalous candidate;
Sort anomalous candidates in a_{seq} by decreasing maximum anomaly score;
foreach *sequence* **do**
  Calculate the time delta Δt between the current candidate sequence and the last sequence classified as an anomaly;
  Compute the rarity-based percent-change threshold $p_{thresh} = p_0 \times e^{(1-\nu \cdot \Delta t)}$;
  Calculate the percent-change p_d between the current and previous sequence scores;
  if $p_d < p_{thresh}$ **then**
    Reclassify subsequent sequences as normal;
    break;
end

Appendix B

Supplementary Figures, Tables, and Plots

B.1 Benchmark datasets

Table B.1: Overview of the used benchmark datasets and all 369 time series

Data Source	**NASA**		**Numenta**				**UCR**
	MSL	**SMAP**	**AdEx**	**AWS**	**Traffic**	**Tweets**	
Time Series	27	53	5	17	7	10	250
# point anom. *(length = 1)*	0	0	0	0	0	0	3
# collective anom. *(length > 1)*	36	67	11	30	14	33	247
Anomalous points	7766	54696	795	6312	1560	15651	49363
Total points	132046	562800	7965	67644	15662	158511	19353766
Anomaly %	5,88%	9,72%	9,98%	9,33%	2,31%	9,87%	0,26%

B.2 Complete implementation pipeline

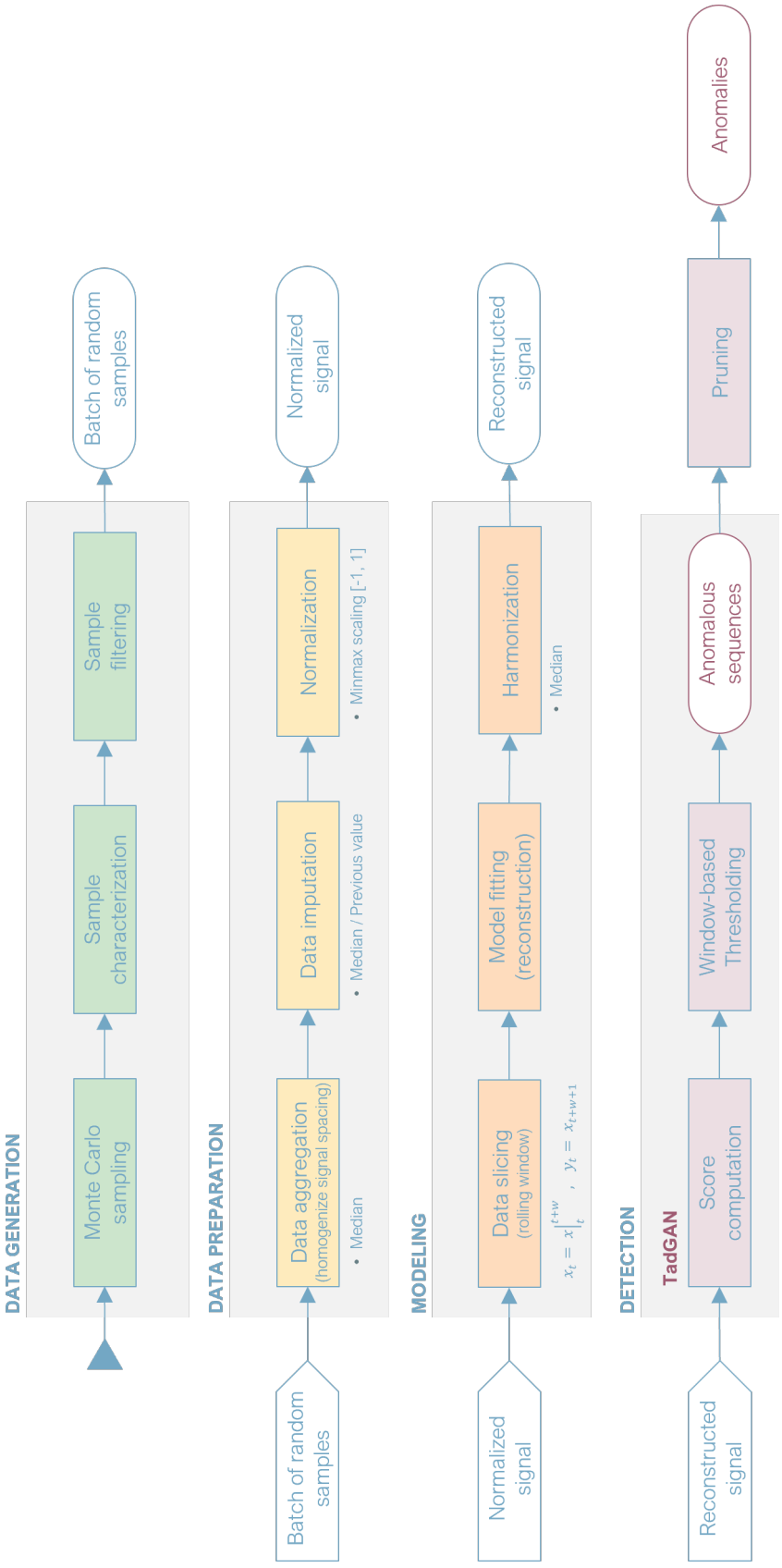

Figure B.1: A schematic representation of the entire implementation pipeline, from Monte Carlo sampling to anomaly detection. TadGAN is used as an illustrative model.

B.3 Rarity Sensitivity Analysis plots for Precision and Recall

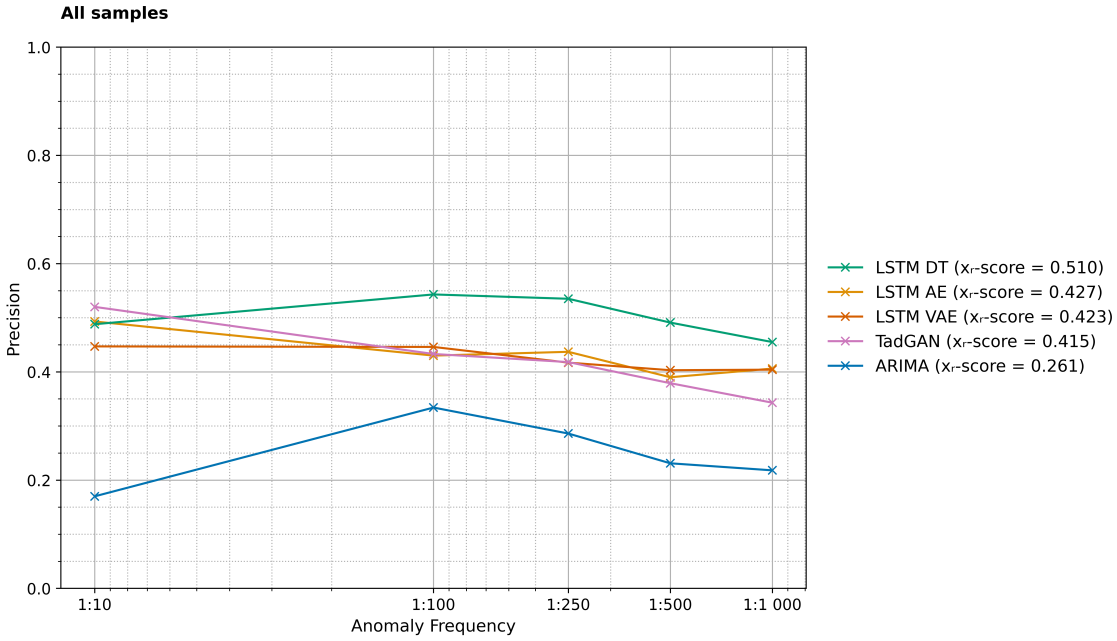

Figure B.2: Rarity Sensitivity Analysis: Precision plot (All samples)

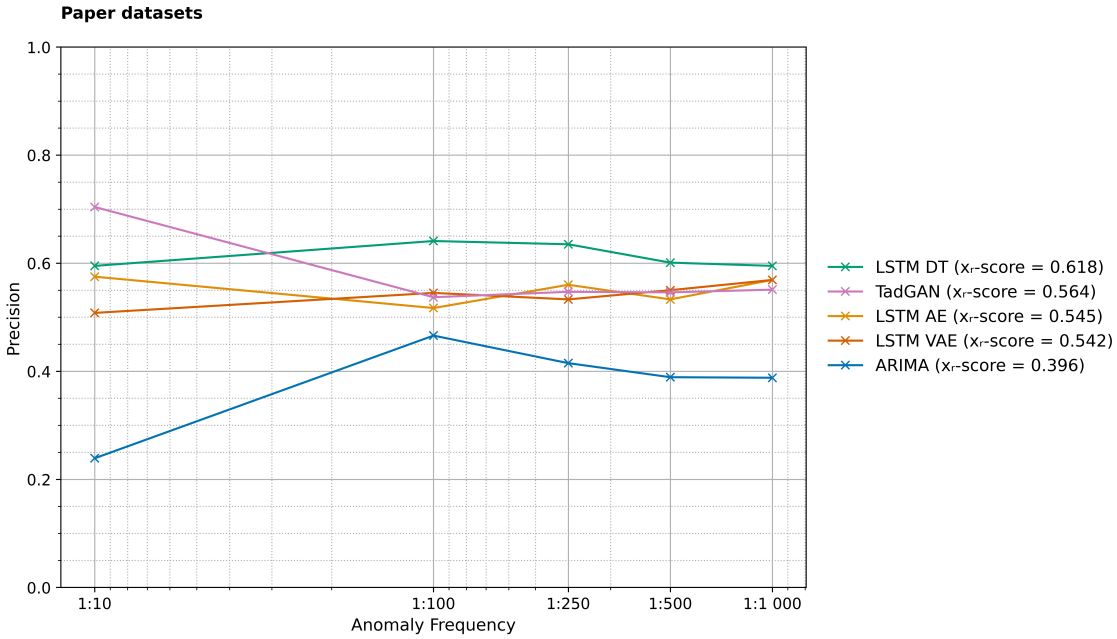

Figure B.3: Rarity Sensitivity Analysis: Precision plot (Paper datasets)

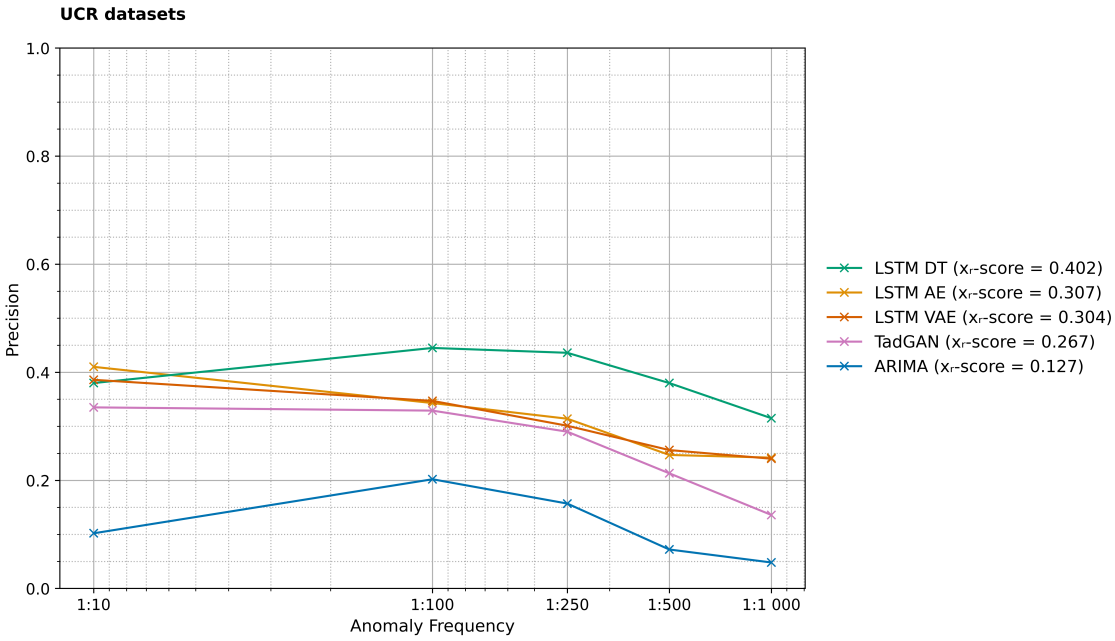

Figure B.4: Rarity Sensitivity Analysis: Precision plot (UCR datasets)

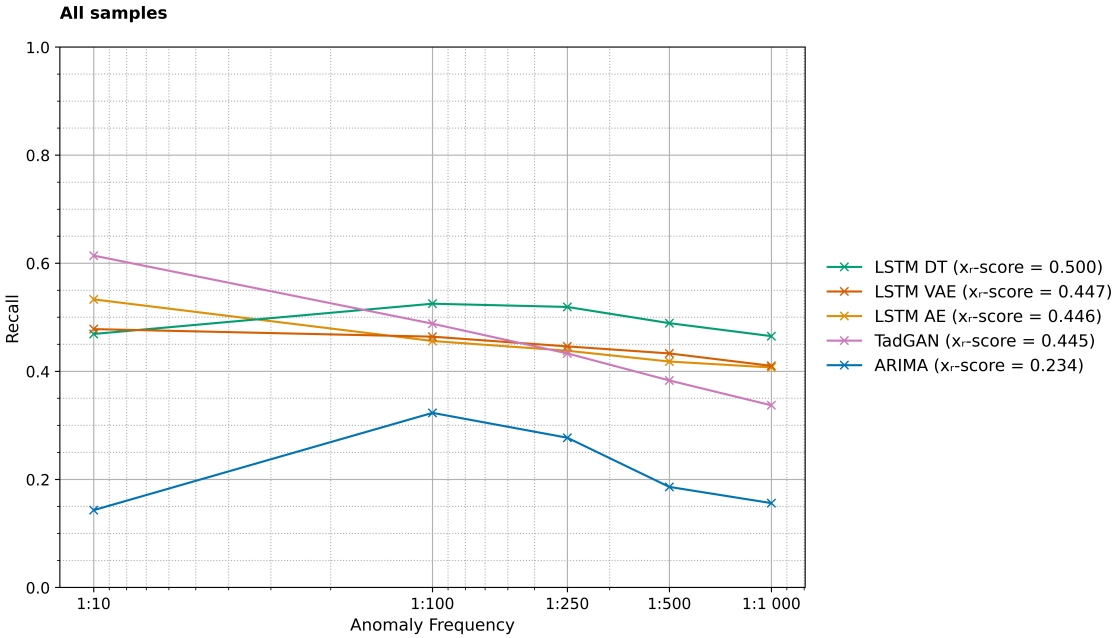

Figure B.5: Rarity Sensitivity Analysis: Recall plot (All samples)

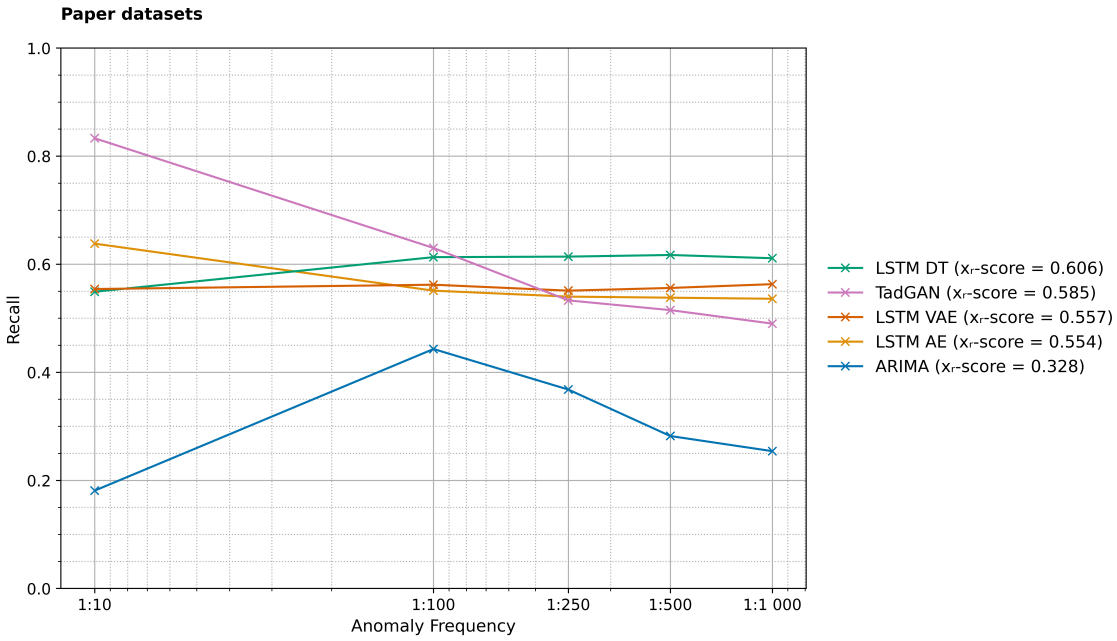

Figure B.6: Rarity Sensitivity Analysis: Recall plot (Paper datasets)

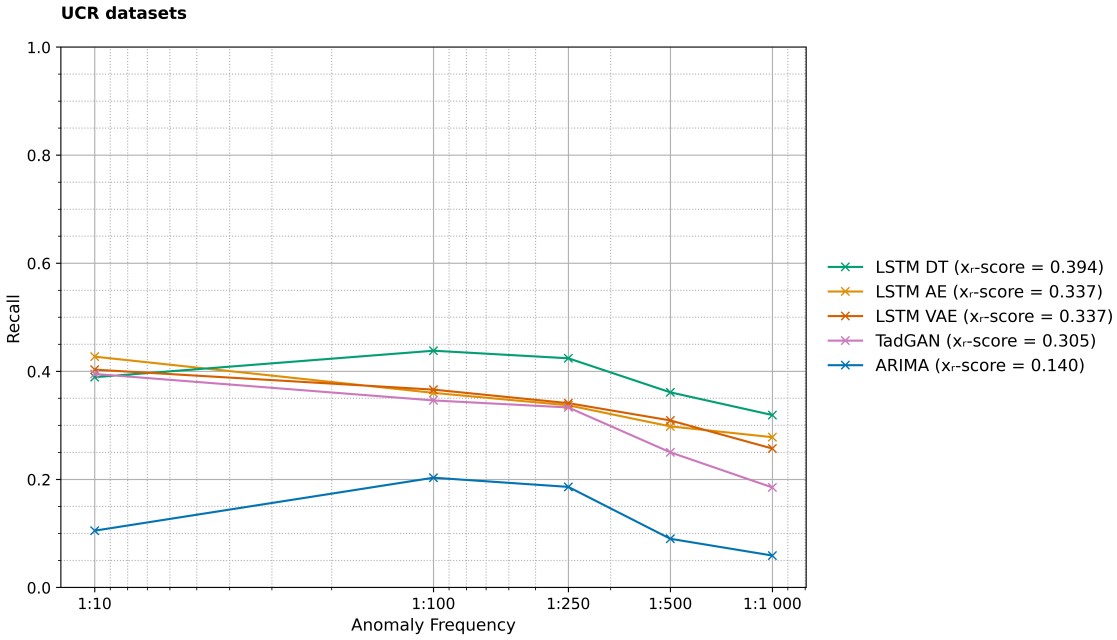

Figure B.7: Rarity Sensitivity Analysis: Recall plot (UCR datasets)

B.4 Complete list of results for the baseline rarity-sensitivity analysis

Table B.2: F1-score, Precision and Recall for all baseline models

Algorithm	Anomaly Frequency	Paper samples			UCR samples			All samples		
		F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec
ARIMA	1:10	0.206	0.239	0.181	0.104	0.102	0.105	0.155	0.170	0.143
	1:100	0.454	0.466	0.443	0.202	0.202	0.203	0.328	0.334	0.323
	1:250	0.390	0.415	0.368	0.170	0.157	0.186	0.280	0.286	0.277
	1:500	0.327	0.389	0.282	0.080	0.072	0.090	0.204	0.231	0.186
	1:1000	0.307	0.388	0.254	0.053	0.048	0.059	0.180	0.218	0.156
LSTM AE	1:10	0.605	0.575	0.638	0.419	0.410	0.427	0.512	0.493	0.533
	1:100	0.534	0.517	0.551	0.352	0.343	0.360	0.443	0.430	0.456
	1:250	0.550	0.560	0.540	0.325	0.314	0.337	0.438	0.437	0.438
	1:500	0.535	0.533	0.538	0.270	0.247	0.298	0.403	0.390	0.418
	1:1000	0.552	0.569	0.536	0.259	0.242	0.278	0.406	0.406	0.407
LSTM VAE	1:10	0.530	0.508	0.554	0.394	0.386	0.403	0.462	0.447	0.478
	1:100	0.553	0.545	0.562	0.356	0.347	0.366	0.455	0.446	0.464
	1:250	0.542	0.533	0.551	0.320	0.301	0.341	0.431	0.417	0.446
	1:500	0.553	0.550	0.556	0.280	0.256	0.309	0.417	0.403	0.433
	1:1000	0.566	0.569	0.563	0.248	0.240	0.257	0.407	0.404	0.410
LSTM DT	1:10	0.571	0.595	0.549	0.384	0.380	0.389	0.478	0.488	0.469
	1:100	0.627	0.641	0.613	0.441	0.445	0.438	0.534	0.543	0.525
	1:250	0.624	0.635	0.614	0.430	0.436	0.425	0.527	0.535	0.519
	1:500	0.609	0.601	0.617	0.370	0.380	0.361	0.490	0.491	0.489
	1:1000	0.603	0.595	0.611	0.317	0.315	0.319	0.460	0.455	0.465
TadGAN	1:10	0.763	0.704	0.833	0.363	0.335	0.395	0.563	0.520	0.614
	1:100	0.580	0.537	0.630	0.338	0.329	0.346	0.459	0.433	0.488
	1:250	0.540	0.547	0.533	0.310	0.290	0.333	0.425	0.418	0.433
	1:500	0.530	0.546	0.515	0.230	0.213	0.251	0.380	0.379	0.383
	1:1000	0.519	0.551	0.490	0.157	0.136	0.185	0.338	0.343	0.337

B.5 Complete list of results for the TadGAN σ variation study

Table B.3: F1-score, Precision and Recall for TadGAN for each value of σ (all samples)

Standard Deviations	**Anomaly Frequency**	**Macro-average F1-score**	**Macro-average Precision**	**Macro-average Recall**
$\sigma = 1$	1:10	0,464	0,340	0,728
	1:100	0,399	0,280	0,694
	1:250	0,303	0,192	0,725
	1:500	0,234	0,137	0,798
	1:1000	0,220	0,128	0,771
$\sigma = 2$	1:10	0,484	0,397	0,619
	1:100	0,429	0,352	0,547
	1:250	0,423	0,346	0,545
	1:500	0,424	0,320	0,629
	1:1000	0,405	0,301	0,615
$\sigma = 6$	1:10	0,555	0,591	0,523
	1:100	0,458	0,484	0,434
	1:250	0,400	0,442	0,365
	1:500	0,290	0,352	0,247
	1:1000	0,285	0,355	0,237

B.6 TadGAN, TadGAN-DT and XTadGAN architectures

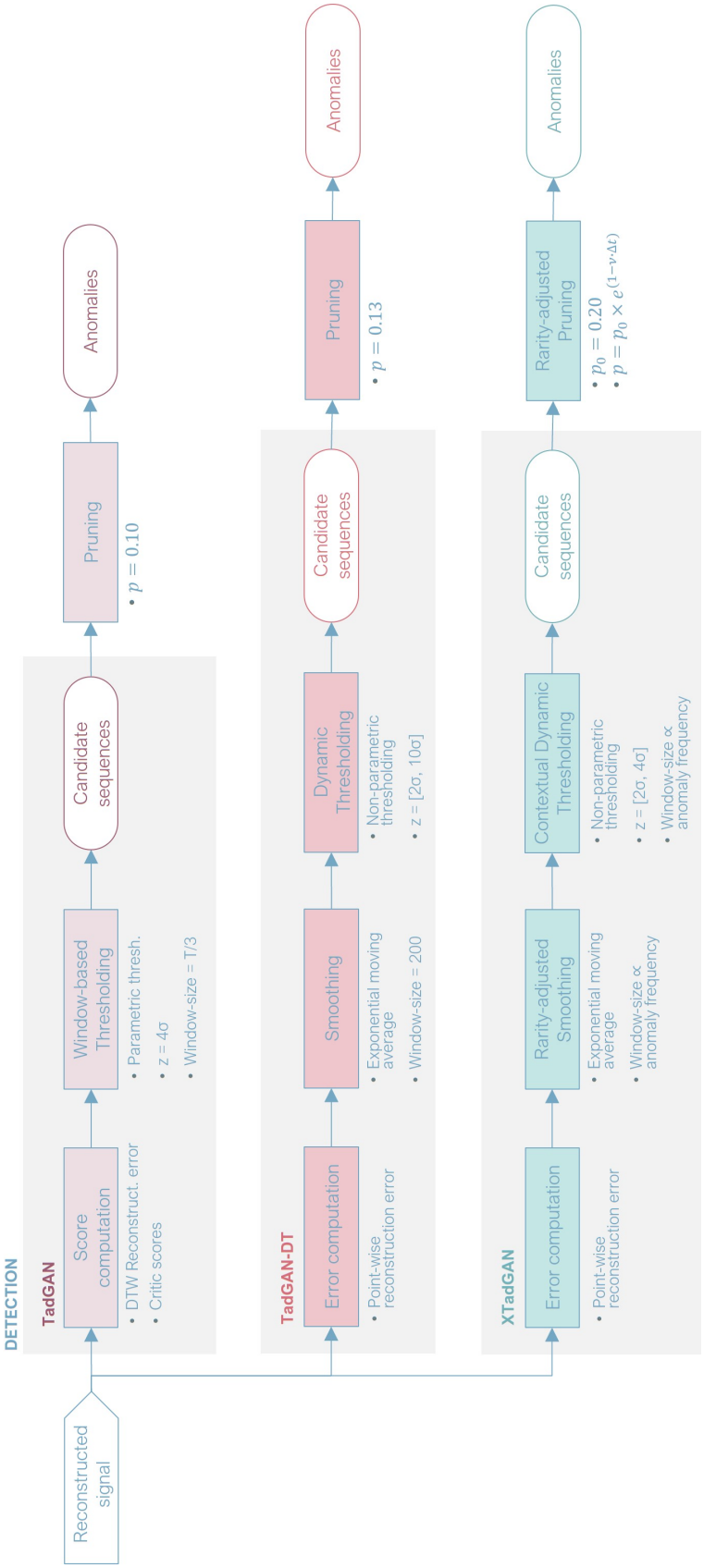

Figure B.8: Schematic diagram comparing the original TadGAN detection pipeline with the proposed novel TadGAN-DT and XTadGAN architectures

B.7 Complete list of results for the TadGAN-DT and XTadGAN architectures

Table B.4: F1-score, Precision and Recall for TadGAN-DT and XTadGAN (all samples)

Algorithm	**Anomaly Frequency**	**Macro-average F1-score**	**Macro-average Precision**	**Macro-average Recall**
TadGAN-DT	1:10	0.431	0.369	0.518
	1:100	0.400	0.329	0.511
	1:250	0.442	0.378	0.533
	1:500	0.516	0.445	0.614
	1:1000	0.521	0.455	0.609
XTadGAN	1:10	0.379	0.471	0.317
	1:100	0.390	0.446	0.346
	1:250	0.481	0.481	0.481
	1:500	0.552	0.542	0.562
	1:1000	0.587	0.600	0.574

Appendix C

Auxiliary study: adversarial training convergence for TadGAN

As discussed in section 4.1.5.1, the TadGAN algorithm showed stable performance above 10 epochs, with neglectable differences. This suggests efficient network convergence. In response to this observation, we conducted an additional analysis on the progression of generator loss and critic scores, which we present briefly here, though it is not the primary focus of this research.

C.1 Analysis of Generator Loss and Critic Scores Progression in TadGAN

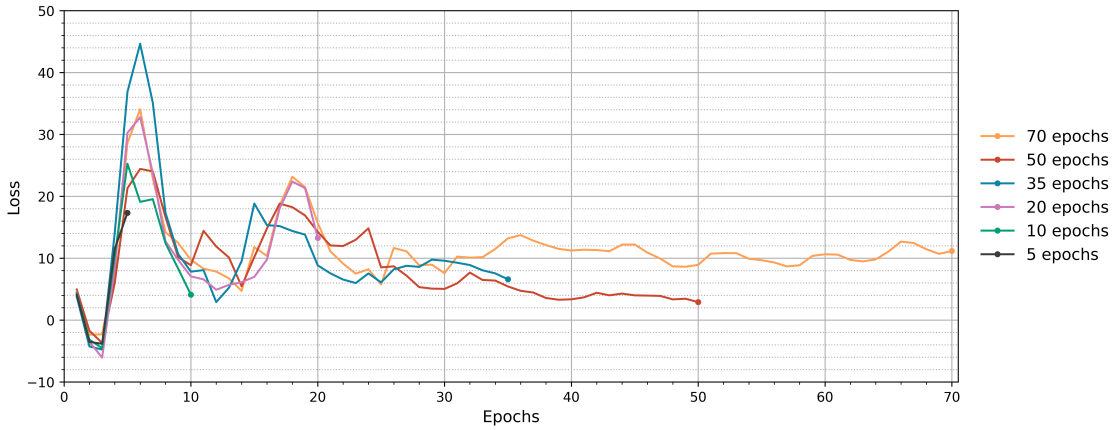

Figure C.1: Evolution of the Encoder-Decoder network loss by number of epochs. Median values for the 119 datasets.

In Figure C.1, we show the loss evolution for the Encoder-Decoder network trained for varying numbers of epochs. As expected in GAN training, there is initially high variability in the loss value, which eventually stabilizes as training progresses.

Additionally, the loss fluctuations exhibit a noticeable pattern, peaking around 4-6 and 16-20 epochs, with a trough between 8-14 epochs and again after 22 epochs. This pattern explains our reported results: even though the network stabilizes around 25 epochs, the consistent loss pattern ensures that by epoch number 10 the loss is low and aligns with the loss value after stabilization.

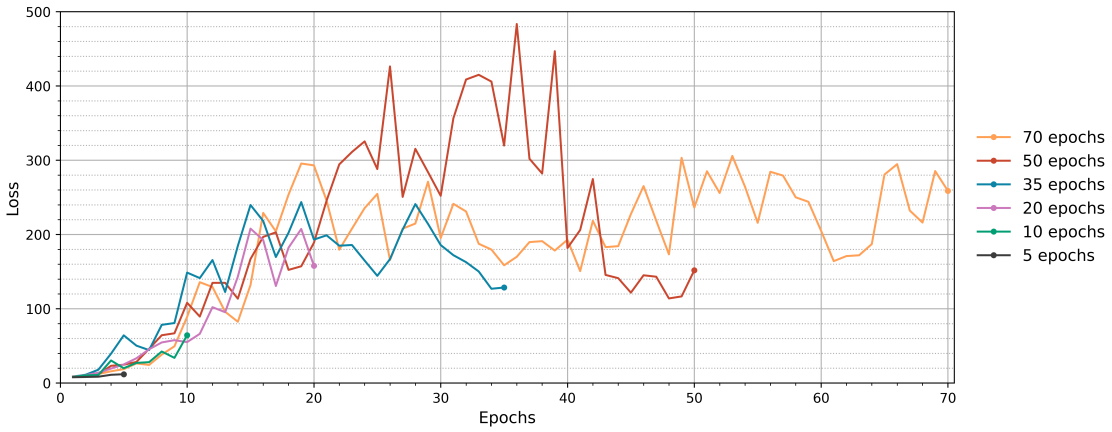

Figure C.2: Evolution of the Critic network C_x loss by number of epochs. Median values for the 119 datasets.

Figure C.2 presents the loss evolution for the Critic C_x network. Overall, this figure shows the expected pattern in a successful GAN training. With the exception of the model trained for 50 epochs, the critic loss increases as training progresses.

In broad terms, detailed in the main body of this work, the critic loss seeks to maximize the difference between scores assigned to real and random samples. Consequently, these results testify that the critic is undergoing successful training and effectively learning how to discriminate between real and fake samples.

References

Charu C. Aggarwal. 2017. *Outlier Analysis*. Springer International Publishing. https://doi.org/10.1007/978-3-319-47578-3_1

Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. 2017. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* 262 (2017), 134–147. https://doi.org/10.1016/j.neucom.2017.04.070 Online Real-Time Learning Strategies for Data Streams.

Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. 2019. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 622–637. https://doi.org/10.1007/978-3-030-20893-6_39

Sarah Alnegheimish, Dongyu Liu, Carles Sala, Laure Berti-Equille, and Kalyan Veeramachaneni. 2022. Sintel: A Machine Learning Framework to Extract Insights from Signals. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22)*. Association for Computing Machinery, 1855–1865. https://doi.org/10.1145/3514221.3517910

Jinwon An and Sungzoon Cho. 2015. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. In *Proceedings of the 2015 IEEE Conference on Advanced Video and Signal-Based Surveillance*. 392–397. https://doi.org/10.1109/AVSS.2015.7327993

Fabrizio Angiulli and Clara Pizzuti. 2002. Fast Outlier Detection in High Dimensional Spaces. In *Principles of Data Mining and Knowledge Discovery*, Tapio Elomaa, Heikki Mannila, and Hannu Toivonen (Eds.). Springer Berlin Heidelberg, 15–27. https://doi.org/10.1007/3-540-45681-3_2

Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. 2017. Time for a Change: a Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis. *Journal of Machine Learning Research* 18, 77 (2017), 1–36. http://jmlr.org/papers/v18/16-305.html

Donald J. Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining* (Seattle, WA) *(AAAIWS'94)*. AAAI Press, 359–370.

Christopher M Bishop. 1994. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing* 141, 4 (1994), 217–222. https://doi.org/10.1049/ip-vis:19941330

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. 2021. A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Comput. Surv.* 54, 3, Article 56 (apr 2021), 33 pages. https://doi.org/10.1145/3444690

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. *SIGMOD Rec.* 29, 2 (2000), 93–104. https://doi.org/10.1145/335191.335388

A Carreño, I Inza, and J A Lozano. 2020. Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. *Artificial Intelligence Review* 53 (2020), 3575–3594. https://doi.org/10.1007/s10462-019-09771-y

Vitor Cerqueira, Luis Torgo, and Carlos Soares. 2019. Machine learning vs statistical methods for time series forecasting: Size matters. (2019). https://doi.org/10.48550/ARXIV.1909.13316

Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3, Article 15 (jul 2009), 58 pages. https://doi.org/10.1145/1541880.1541882

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* 35, 1 (2018), 53–65. https://doi.org/10.1109/MSP.2017.2765202

Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. 2018. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Andrew F. Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. 2013. Systematic Construction of Anomaly Detection Benchmarks from Real Data. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description* (Chicago, Illinois) *(ODD '13)*. Association for Computing Machinery, New York, NY, USA, 16–21. https://doi.org/10.1145/2500853.2500858

Ralph Foorthuis. 2021. On the nature and types of anomalies: a review of deviations in data. *International Journal of Data Science and Analytics* 12, 4 (aug 2021), 297–331. https://doi.org/10.1007/s41060-021-00265-1

Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2020. TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks. *arXiv preprint arXiv:1905.10994* (2020), 33–43. https://doi.org/10.1109/BigData50022.2020.9378139

M. Goldstein and S. Uchida. 2016. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLoS One* 11, 4 (2016), e0152173. https://doi.org/10.1371/journal.pone.0152173

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. https://doi.org/10.48550/ARXIV.1406.2661

Mononito Goswami, Cristian Challu, Laurent Callot, Lenon Minorics, and Andrey Kan. 2022. Unsupervised Model Selection for Time-series Anomaly Detection. (2022). https://doi.org/10.48550/ARXIV.2210.01078

Frank E. Grubbs. 1969. Procedures for Detecting Outlying Observations in Samples. *Technometrics* 11, 1 (1969), 1–21. https://doi.org/10.1080/00401706.1969.10490657

D.M. Hawkins. 1980. *Identification of Outliers*. Chapman and Hall.

Zengyou He, Xiaofei Xu, and Shengchun Deng. 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters* 24, 9 (2003), 1641–1650. https://doi.org/10.1016/S0167-8655(03)00003-5

Vincent Hodge and J. Austin. 2004. A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review* 22 (2004), 85–126. https://doi.org/10.1023/B:AIRE.0000045502.10941.a9

Alexis Huet, Jose Manuel Navarro, and Dario Rossi. 2022. Local Evaluation of Time Series Anomaly Detection Algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM. https://doi.org/10.1145/3534678.3539339

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. https://doi.org/10.1145/3219819.3219845

Won-Seok Hwang, Jeong-Han Yun, Jonguk Kim, and Hyoung Chun Kim. 2019. Time-Series Aware Precision and Recall for Anomaly Detection: Considering Variety of Detection Result and Addressing Ambiguous Labeling. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 2241–2244. https://doi.org/10.1145/3357384.3358118

Rob J. Hyndman, Earo Wang, and Nikolay Laptev. 2015. Large-Scale Unusual Time Series Detection. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 1616–1619. https://doi.org/10.1109/ICDMW.2015.104

M.V. Joshi, V. Kumar, and R.C. Agarwal. 2001. Evaluating boosting algorithms to classify rare classes: comparison and improvements. In *Proceedings 2001 IEEE International Conference on Data Mining*. 257–264. https://doi.org/10.1109/ICDM.2001.989527

Viacheslav Kozitsin, Iurii Katser, and Dmitry Lakontsev. 2021. Online forecasting and anomaly detection based on the ARIMA model. *Applied Sciences* 11, 7 (2021), 3194. https://doi.org/10.3390/app11073194

Alexander Lavin and Subutai Ahmad. 2015a. Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE. https://doi.org/10.1109/icmla.2015.141

Alexander Lavin and Subutai Ahmad. 2015b. Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark. IEEE. https://doi.org/10.1109/icmla.2015.141

Dan Li, Dacheng Chen, Lei Shi, Baihong Jin, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. https://doi.org/10.48550/ARXIV.1901.04997

Bo Liu, Jianqiang Li, Cheng Chen, Wei Tan, Qiang Chen, and MengChu Zhou. 2015. Efficient Motif Discovery for Large-Scale Time Series in Healthcare. *IEEE Transactions on Industrial Informatics* 11, 3 (2015), 583–590. https://doi.org/10.1109/TII.2015.2411226

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. (2016). https://doi.org/10.48550/ARXIV.1607.00148

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series.. In *ESANN*, Vol. 2015. 89.

Eduardo H. M. Pena, Marcos V. O. de Assis, and Mario Lemes Proença. 2013. Anomaly Detection Using Forecasting Methods ARIMA and HWDS. In *2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*. 63–66. https://doi.org/10.1109/SCCC.2013.18

João Pereira and Margarida Silveira. 2019. Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 1–7. https://doi.org/10.1109/BIGCOMP.2019.8679157

Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. *Signal Processing* 99 (2014), 215–249. https://doi.org/10.1016/j.sigpro.2013.12.026

Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. 2019. Deep semi-supervised anomaly detection. (2019). https://doi.org/10.48550/ARXIV.1906.02694

Kamran Shaukat, Talha Mahboob Alam, Suhuai Luo, Shakir Shabbir, Ibrahim A. Hameed, Jiaming Li, Syed Konain Abbas, and Umair Javed. 2021. A Review of Time-Series Anomaly Detection Techniques: A Step to Future Perspectives. In *Advances in Information and Communication*, Kohei Arai (Ed.). Springer International Publishing, 865–877. https://doi.org/10.1007/978-3-030-73100-7_60

Sameer Singh and Markos Markou. 2003. Novelty detection: a review—part 1: statistical approaches. *Signal Processing* 83, 12 (2003), 2481–2497. https://doi.org/10.1016/j.sigpro.2003.07.018

Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. 2007. Conditional Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering* 19, 5 (2007), 631–645. https://doi.org/10.1109/TKDE.2007.1009

Ingo Steinwart, Don Hush, and Clint Scovel. 2005. A Classification Framework for Anomaly Detection. *Journal of Machine Learning Research* 6, 2 (2005).

Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. 2018. Precision and Recall for Time Series. https://doi.org/10.48550/ARXIV.1803.03639

J. Martínez Torres, P.J. Garcia Nieto, L. Alejano, and A.N. Reyes. 2011. Detection of outliers in gas emissions from urban areas using functional data analysis. *Journal of Hazardous Materials* 186, 1 (2011), 144–149. https://doi.org/10.1016/j.jhazmat.2010.10.091

J.W. Tukey. 1977. *Exploratory Data Analysis*. Addison-Wesley.

David H. Wolpert. 2002. The Supervised Learning No-Free-Lunch Theorems. https://api.semanticscholar.org/CorpusID:13844424

Lawrence Wong, Dongyu Liu, Laure Berti-Equille, Sarah Alnegheimish, and Kalyan Veeramachaneni. 2022. AER: Auto-Encoder with Regression for Time Series Anomaly Detection. In *2022 IEEE International Conference on Big Data (IEEE BigData)*. IEEE, 1152–1161. https://doi.org/10.1109/BigData55660.2022.10020857

Hu-Sheng Wu. 2016. A survey of research on anomaly detection for time series. In *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 426–431. https://doi.org/10.1109/ICCWAMTIP.2016.8079887

Renjie Wu and Eamonn Keogh. 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering* (2021). https://doi.org/10.1109/TKDE.2021.3112126

Asrul H. Yaacob, Ian K.T. Tan, Su Fong Chien, and Hon Khi Tan. 2010. ARIMA Based Network Anomaly Detection. In *2010 Second International Conference on Communication Software and Networks*. 205–209. https://doi.org/10.1109/ICCSN.2010.55

Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. 2019. BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4433–4439. https://doi.org/10.24963/ijcai.2019/616

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. https://doi.org/10.48550/ARXIV.1703.10593