Unmasking Overestimation: A Re-evaluation of Deep Anomaly Detection in Spacecraft Telemetry.

Air Transport Operations MSc thesis

Lars Herrmann

TU Delft

Unmasking Overestimation: A Re-evaluation of Deep Anomaly Detection in Spacecraft Telemetry.

Air Transport Operations MSc thesis

by

Lars Herrmann

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on the 4th of July 2023.

Student number:	4471857
Project duration:	March 2022 – June 2023
Thesis committee:	Dr. Bruno Lopes Dos Santos
	Dr. Alessandro Bombelli
	Dr. Angelo Cervone
	Dr. Wim J.C. Verhagen
	Marie Bieber

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Acknowledgements

As I stand at the threshold of this significant milestone, I pause to acknowledge those who have journeyed with me on this exciting and challenging endeavor.

My deepest appreciation goes to my supervisor Marie Bieber, who has been an extraordinary pillar of support throughout this journey. Marie, you guided me meticulously through the process, inspiring me to aim high, which led me to present at two conferences and submit to a renowned journal. I am grateful not only for your professional guidance but also for your friendship and understanding during the times when things did not go as planned. It's amusing to remember those early English meetings when our communication felt more distant - how far we've come since then!
I am also deeply grateful to my co-supervisors, Bruno Santos and Wim Verhagen. Your valuable feedback and thoughtful guidance at key milestones kept me on track. Your insight was instrumental in refining the scope of the thesis, for which I am particularly appreciative.
My sincere thanks go to the colleagues from ESA, who lent their expertise and support during this process. I would like to particularly acknowledge Fabrice Cosson, our main contact; Rene Seiler, our domain expert on reaction wheels; and Jean-Francois Flamand, for his enthusiastic assistance.

As I bid adieu to the student life that I have cherished since 2015, I reflect upon the numerous connections that have enriched this experience. The lively camaraderie of Marcushof, with 25 of us on a single floor, made the initial homesickness bearable. Special thanks to Joao, Bryony, Joko, Verena, and Marco for brightening those early days.
I am incredibly thankful for the bond I formed with Jelle. The countless sleepovers, despite living just around the corner, and the shared ups and downs made the journey not just bearable, but memorable.
As I moved into my apartment with the "Balpol boys," Kees and Emile, a new chapter unfolded, brimming with wild stories and friendships that I continue to cherish. I'm also grateful for the "Menno boys" - Stav, Louis, and Jordy, for their companionship.
I would like to extend my gratitude to the DUT19 team, especially the core members, for a year filled with learning and adventure as we constructed our prized vehicle.
My heartfelt appreciation goes to Kipras, my fellow food, sport, and nerd-stuff enthusiast. Our shared love for Brazilian jiu-jitsu, albeit belated on my part, has been a source of joy.
An integral part of this acknowledgement, and indeed, my journey, is my dear friend Kurt. Your open-hearted generosity knew no bounds - from offering me unrestricted access to your flat and keys during my summer visits to Berlin, to ensuring you visited me at least once a year in the Netherlands. Even when I faltered in keeping in touch, you were there, consistently reaching out. Your unwavering friendship is something I am immensely grateful for. Thank you, Kurt, for being a steady presence in my life.
I am incredibly grateful to my dear friend Emre for his companionship. Our shared moments over coffee and porridge and his invaluable emotional support during the highs and lows will always be cherished.

During my master's journey, the world flipped due to the pandemic, and simultaneously, I embraced the joy of fatherhood. My profound thanks to Rosanne, for bringing our beautiful daughter, Indy, into the world and for her unwavering support during the rocky initial phase of my thesis.
Lastly, but certainly not least, I owe a debt of gratitude to my mother, my greatest supporter and believer. You have been there with me since I declared my intention to move to Berlin and have stood by me in every step that followed. Your unending faith in me has been the strongest source of my inspiration. Your love and support made all the difference.

To each one who has touched my life during this journey - thank you for making it more rewarding and memorable.

Lars Herrmann
Delft, June 2023

Contents

List of Figures vii

List of Tables ix

List of Abbreviations xi

Introduction xiii

I Scientific Paper 1

II Literature Study
previously graded under AE4020 25

1 Introduction 27

2 Properties of Time Series Data 29
2.1 Temporality . . . 29
2.2 Dimensionality . . . 29
2.3 Nonstationarity . . . 29
2.4 Noise . . . 30

3 Anomalies 31

4 Deep Learning 33
4.1 Single Layer Perceptron . . . 33
4.2 Feed Forward Neural Network . . . 33
4.3 Convolutional Neural Network . . . 34
4.4 Long Short Term Memory networks . . . 35
4.5 Autoencoder . . . 36
4.6 Variational Autoencoder . . . 36
4.7 Generative Adversarial Networks . . . 37
4.8 Transformer . . . 37
4.8.1 Self-attention . . . 38
4.8.2 Transformer architecture . . . 39

5 Deep Anomaly Detection 41
5.1 Anomaly criteria . . . 42
5.2 State of the Art Deep Anomaly Detection . . . 42
5.2.1 Metrics/Critique of the State of the Art . . . 42

6 Metrics 47

7 Data-set 51
7.1 Real-Life Satellite Telemetry Data of ESA Satellites . . . 51
7.2 Validation Data-sets . . . 51
7.3 Comparison . . . 51

8 Research Approach 53
8.1 Knowledge Gap . . . 53
8.2 Scope . . . 53
8.3 Research Objective . . . 53
8.4 Research Question . . . 53

9 Conclusion 55

Bibliography 57

List of Figures

3.1 Anomaly types in time series data [9]. 32
3.2 Detailed classification of anomalies in time series according to Tang et al. [9]. 32

4.1 Schematic of a Single-Layer-Perceptron. 34
4.2 Schematic view of a Feedforward Neural Network [20]. 34
4.3 An example of a 2-D convoltion on 4x2 data with a 2x2 Kernel without padding [13]. 35
4.4 Long Short Term Memory Network [20]. 35
4.5 Schematic of an Autoencoder. 36
4.6 Visualisation of self-attention, excluding the softmax operation [4]. 38
4.7 Self-attention with linear transformation for query, key and value role [4]. 39
4.8 Transformer block with a self attention layer, layer normalization, a feed forward layer and another layer normalization [4]. 40

5.1 Minimax association learning as shown by Xu et al. [41]. 43

6.1 Adjusting of instance-based evaluation with Point Adjust. 48

List of Tables

5.1 Performance comparison of recent Deep Anomaly Detection algorithms. 42

List of Abbreviations

AE	Autoencoder
AUC	Area Under the Curve
CNN	Convolutional Neural Network
DAD	Deep Anomaly Detection
ESA	European Space Agency
GAN	Generative Adversarial Networks
GAT	Graph Attention Network
GMM	Gaussian Mixture Model
GNN	Graph Neural Networks
kNN	k-Nearest Neighbor
LSTM	Long Short-Term Memory
MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration
NPT	Nonparametric Dynamic Thresholding
OCSVM	One-Class Support Vector Machine
PA	Point Adjust
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SMAP	Soil Moisture Active Passive
SMD	Server Machine Dataset
SWAT	Secure Water Treatment
USAD	UnSupervised Anomaly Detection
VAE	Variational Autoencoder

Introduction

The genesis of this research project traces its roots back to a cooperative effort between the Technical University of Delft (TU Delft) and the European Space Agency (ESA) as part of the Open Space Innovation Platform (OSIP). Launched in 2019, OSIP emerged as ESA's strategic initiative to serve the burgeoning demands of the modern space sector. Being a prominent conduit for innovation, it welcomes novel ideas that challenge the conventional boundaries of the space industry. This collaborative project epitomizes such groundbreaking endeavors.

The project, focusing on anomaly detection in spacecraft telemetry data, emerged as a response to the need for a technological leap in the field. As the deluge of telemetry data generated by satellites and other complex space systems continues to surge, the requirement for efficient and accurate anomaly detection methods has become an imperative. Traditional techniques, which are heavily reliant on human interpretation and preset criteria, face various challenges. These include the need for expert analysis and ongoing modifications to keep up with the dynamic nature of space missions.

Our research project sought to address these issues head-on by critically evaluating the use of Deep Anomaly Detection (DAD) methods. It took two particular systems under its purview, namely star trackers and reaction wheels. Upon discovering a substantial amount of failure data for reaction wheels, it became the focus of our study. ESA experts were regularly consulted throughout the research to provide nuanced insights into the reaction wheel behavior, failures, and data from the sentinel missions.

What sets this project apart is its rigorous scrutiny of existing DAD methods, unveiling their limitations, and pioneering the need for alternative strategies. The study's deep dive into the telemetry data's intricacies, backed by cutting-edge DAD algorithms, sets a unique precedent for future research in the domain.

From a societal standpoint, the project's implications are profound. The ability to accurately identify anomalies in spacecraft telemetry data plays a pivotal role in assuring the smooth operation of space missions. By improving these anomaly detection methods, the project contributes directly to enhancing the reliability of space missions, ultimately benefiting global satellite communications, weather forecasting, navigation, and Earth observation systems.

Furthermore, by improving upon existing methods, this research promotes efficiency in resource usage, potentially leading to cost savings in space mission operations. Also, the findings of this project can be translated into other industries that rely heavily on anomaly detection, such as manufacturing, healthcare, and energy, thereby leading to broader societal benefits.

This project, through its rigorous study and innovative methodology, paints a new path for future research endeavors. It proves that pushing the envelope, asking the right questions, and challenging existing frameworks can lead to more robust and reliable anomaly detection methods in spacecraft telemetry data. And as we enter an era where space exploration and its commercial exploitation become increasingly critical, such strides in the domain of anomaly detection will hold the key to the door of unprecedented possibilities.

This thesis report is organized as follows : In Part I, the scientific paper is presented. Part II contains the relevant Literature Study that supports the research.

I

Scientific Paper

Unmasking Overestimation: A Re-evaluation of Deep Anomaly Detection in Spacecraft Telemetry

Lars Herrmann[1*], Marie Bieber[1], Wim J.C. Verhagen[2], Fabrice Cosson[3], Bruno F. Santos[1]

[1*]Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629HS, Zuid-Holland, The Netherlands.
[2]Aerospace Engineering and Aviation, RMIT University, Carlton, 3053, Victoria, Australia.
[3]European Space Research & Technology Centre, ESA, Keplerlaan 1, Noordwijk , 2200AG, Zuid-Holland, The Netherlands.

*Corresponding author(s). E-mail(s): L.Herrmann-1@student.tudelft.nl;
Contributing authors: M.T.Bieber@tudelft.nl;
wim.verhagen@rmit.edu.au; Fabrice.Cosson@esa.int;
B.F.Santos@tudelft.nl;

Abstract

As the volume of telemetry data generated by satellites and other complex systems continues to grow, there is a pressing need for more efficient and accurate anomaly detection methods. Current techniques often rely on human analysis and preset criteria, presenting several challenges including the necessity for expert interpretation and continual updates to match the dynamic mission environment. This paper critically examines the use of Deep Anomaly Detection (DAD) methods in addressing these challenges, evaluating their efficacy on real-world spacecraft telemetry data. It exposes multiple flaws in current DAD research, highlighting the tendency for performance results to be overestimated and suggesting that simpler methods can sometimes outperform more complex DAD algorithms. By comparing established metrics for anomaly detection with newly proposed ones, this paper aims to improve the evaluation of DAD algorithms. It underscores the importance of using less accuracy-inflating metrics and offers a comprehensive comparison of DAD methods on popular benchmark datasets and real-life satellite telemetry data. Among the DAD methods examined, the LSTM algorithm demonstrates considerable promise. However, the paper also reveals the potential limitations of this approach, particularly in complex systems that lack a single, clear predictive failure channel. The paper concludes with a series

of recommendations for future research, including the adoption of best practices, the need for high-quality, pre-split datasets, and the investigation of other prediction error methods. Through these insights, this paper contributes to the improved understanding and application of DAD methods, ultimately enhancing the reliability and effectiveness of anomaly detection in real-world scenarios.

Keywords: Deep Anomaly Detection (DAD) , Real-life Satellite Telemetry Data, Anomaly Detection Metrics, Time-series Anomaly Detection

1 Introduction

Satellites and other complex systems generate increasing amounts of telemetry data that can be analyzed by terrestrial systems. Monitoring this data is crucial for ensuring the success of spacecraft operations and missions. Anomaly detection is a key method to prevent spacecraft loss due to undetected flaws or slow responses to hazards. However, most current anomaly detection methods rely on human evaluation of aggregated data and out-of-limit checks with established criteria. These methods have significant disadvantages, as they require specialized expertise and effort to organize and analyze the data. Additionally, the dynamic mission environment requires continual updates to the criteria, and important information may be missed during the telemetry data compilation process.

In the coming years, these challenges are expected to intensify due to ongoing advancements in computer and storage capacities. As a result, the volume of telemetry data will significantly increase, placing greater demands on technical resources and data aggregation techniques. Deep learning for anomaly detection in high-dimensional time-series data has shown promising results with recent advancements in neural network architecture and increased processing power. Some deep learning algorithms perform better than traditional anomaly detection techniques on real-world time-series challenges [1], with reported F1 scores greater than 0.9 indicating highly accurate deep anomaly detection capabilities. However, the widely used Point Adjust (PA) method [2] in modern Deep Anomaly Detection (DAD) research has faced criticism in recent publications, mainly due to its tendency to overestimate accuracy[3–5]. Wu and Keogh also highlighted four common problems with existing datasets: small sample sizes, incorrect anomaly densities, incorrect labeling, and run-to-failure bias[6]. Given these limitations, it is crucial to re-evaluate current state-of-the-art deep anomaly detection methods and assess their accuracy using improved metrics and real-world data.

This paper aims to address the aforementioned shortcomings by applying and evaluating DAD methods on real-world spacecraft telemetry data. In doing so, it contributes to the state of the art in the following ways:

- We compare established metrics for anomaly detection with newly proposed metrics, aiming to improve the measurement and evaluation of anomaly detection algorithms. By considering a range of metrics, we provide a more robust framework for assessing algorithm performance.

- We investigate the performance of anomaly detection methods with different levels of complexity using metrics that are less likely to overestimate accuracy. This allows for a more comprehensive assessment of the methods' effectiveness.

- The comparison and evaluation of the anomaly detection methods are conducted on two popular benchmark datasets as well as on a real-life dataset of satellite telemetry data. This provides a comprehensive and realistic assessment of the methods' performance in different scenarios.

Through these contributions, we aim to enhance the understanding and evaluation of Deep Anomaly Detection methods, ultimately improving the reliability and effectiveness of anomaly detection in real-world applications. The remainder of this paper is organized as follows. Section 2 provides an introduction to the background and current state of the art in anomaly detection. Section 3 describes the metrics, anomaly detection algorithms, and thresholding methods employed in this study for the comparison and evaluation of the methods. Section 4 provides a comprehensive overview of the datasets utilized in the case studies, offering pertinent information about each dataset. Furthermore, it presents the detailed results obtained from the case studies and engages in an in-depth discussion of the findings. Finally, in Section 5, we summarize the key findings and limitations of the study and suggest potential directions for further research and improvement in the field of Deep Anomaly Detection.

2 Literature Review

2.1 State of the Art

Numerous studies have investigated various aspects of deep anomaly detection on satellite data, exploring diverse topics within this field. At the core of anomaly detection algorithms lies the data they analyze. Time series data, characterized by a sequence of time-dependent variables, represents a distinctive type of input data with its own unique properties and challenges. Grasping the characteristics of time series data can aid in effectively leveraging the contextual information it holds. The main characteristics of time-series data are temporality, dimensionality, non-stationarity, and noise [7]. Temporality arises because time-series is a set of measurements recorded with corresponding dates [8]. Usually, they are recorded at equal time intervals, but time series data with unequal spacing also exists. The series of data points have a specific order, with each subsequent point relying on prior values. The number of features included in each observation is referred to as dimensionality. Choi et al.

distinguish between univariate data with one variable and multivariate data with multiple variables [7]. Low-dimensional and high-dimensional data are also defined by Chalapathy and Chawla [9]. The dimensionality affects computational cost and determines the most appropriate method for analysis. A time series is considered stationary if its characteristics remain unchanged regardless of the time it's observed. Such series are simpler to analyze, however, most real-world data is affected by fluctuations. Seasonality, concept drift, and change points are examples of factors that make time series non-stationary. Noise is a random disturbance that interferes with a useful information system. It refers to unwanted changes in an information signal during its capture, storage, transmission, processing, or conversion [10]. Noise makes it challenging to distinguish between anomalous samples and the noise itself. Reducing noise in the data preparation stage is crucial as it significantly impacts the performance of detection algorithms.

Various efforts to characterize the nature of anomalous data have been documented in the literature. Hawkins definition of an anomaly is: "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" [11]. According to Barnett and Lewis' an anomaly is "an observation or subset of observations, which appears to be inconsistent with the remainder of that set of data" [12]. Time-series anomalies are classified into three types: Point, Contextual, and Collective anomalies [13]. Point anomalies are data points or sequences that drastically depart from typical values. Contextual anomalies are instances that appear strange in a particular context, but could be normal in other circumstances. Collective anomalies consist of multiple related data points that are unusual compared to the entire dataset, even though individually they might not seem suspicious. While anomalies in time series can be broadly categorized into these three types, they can also be further divided into more specific subsets that depend on the domain being analyzed. For example, Tang et al. defined six patterns to categorize vibration anomalies[14].

Choi et al. categorized classic anomaly detection approaches into time/frequency domain analysis, statistical models, distance-based models, auto-regressive models, and clustering models [7] . Basora et al. groups distance-based and clustering models together and expands the classification to include ensemble-based, domain-based, and subspace-based methods [15] . Telemetry data from spacecraft is typically analyzed in the time domain using simple limit checking with upper and lower limits for the observed values [16]. However, fixed thresholds can be limiting for dynamic systems. To address this, adaptive limit checking has been developed [17] .

Traditional methods for anomaly detection face limitations in scaling with increasing dimensionality and large data volumes. In contrast, deep learning methods, specifically DAD, have shown superior performance in such scenarios [9, 18, 19]. Recent research has focused on DAD to overcome these challenges, and it has been successfully applied to various tasks across different domains.

The advancement in deep learning architecture and increase in data and computational resources have resulted in deep learning models performing some tasks at a human-level, even surpassing it in certain cases. This has also fueled extensive research in the field of diagnostics. The most common deep learning architectures that are used in anomaly detection are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), specifically the Long Short-Term Memory (LSTM) Networks [20], Autoencoder (AE) and Variational Autoencoder (VAE) [21], Generative Adversarial Networks (GAN) [22], Graph Attention Network (GAT) [23] and Transformers [24].

DAD models aim to minimize an objective loss function during training, which depends on the model architecture and relates to abnormality decision criteria. These models output an anomaly score, which is a numeric value that indicates the probability of a sample being abnormal, and samples are labelled as anomalous when the score exceeds a certain threshold. While domain experts used to set the threshold empirically, it is now determined based on training results, either through performance evaluation on validation data for labelled data or by using Extreme Value Theory for non-labelled data [25]. DAD can be categorized into three types depending on the method used to calculate the anomaly score: Reconstruction error, Prediction error, and Dissimilarity [7]. However, the first two criteria are more commonly used than the third.

Autoencoders, variational autoencoders, generative adversarial networks, and transformers are examples of models that typically use reconstruction errors to obtain an anomaly score. They learn low-dimensional representations of the data and map them to the input space to calculate residuals by comparing the reconstructed values with the original data. Reconstruction-based methods assume that anomalies lose information when mapped to a lower dimensional space and cannot be effectively reconstructed. Therefore, high reconstruction errors suggest a high chance of being anomalous [26].Prediction error methods use a model to fit the given data and predict future values. The difference between the model output and the actual values is used to identify anomalies. Commonly used models for prediction error anomaly scores include LSTM, CNN, Graph Neural Networks (GNN), and Transformers. Dissimilarity based models measure distance or similarity between data instances. Objects that are distant from a cluster or distribution are considered anomalies. A table of current state of the art algorithms with their corresponding architecture, anomaly criterion and benchmark scores can be seen in Table **??**.

2.2 Limitations in the State of Art

While many papers claim to have unsupervised algorithms, several of them suffer from data leakage, especially many of those that achieve the best results. Data leakage refers to the utilization of information during the model training process that would not be available at prediction time. Algorithms that really provide unsupervised results often rely on extreme value theory [16, 23, 26, 27, 29] or use a discriminator network [27]. However, many top-performing algorithms create the illusion of being

unsupervised but actually utilize the test data to determine the threshold [18, 30, 31].

Classic cross-validation with a standard split can be considered data leakage when applied to most time series datasets. This is because it relies on the assumptions of independence and identical distribution, which may not hold in many real-world scenarios. The independence assumption implies that the values in the time series are not influenced by previous or future values. However, in many engineering systems, there are dynamics and interdependencies that violate this assumption. For example, in a mechanical system, the current state of the system may depend on its past states or external factors. This violates the independence assumption and can lead to biased and inaccurate performance estimates when using standard cross-validation. Similarly, the identical distribution assumption assumes that each observation in the time series is drawn from the same underlying probability distribution. However, in engineering systems, it is common for the distribution to change over time due to factors such as wear and tear, aging, or external influences. Therefore, the identical distribution assumption may not hold, and using standard cross-validation can introduce bias and inaccuracies in the evaluation of anomaly detection methods. There exist other methods, for example cross-validation on a rolling basis, that could be used. Additionally, some algorithms use the input of the assumed or known fraction of outliers to determine the threshold [19, 33]. This approach has two issues. Firstly, the true outlier fraction is often unknown in real-life datasets. Secondly, it is not appropriate to apply the outlier fraction to the training dataset, as the training data is typically assumed to consist of normal instances.

Recent deep anomaly detection research has reported high anomaly detection scores, leading to a perceived increase in accuracy. However, many studies use a method called PA [16] which artificially inflates metric scores and significantly improves real positive identification. Some authors have identified problems with PA and proposed new metrics[3–5]. They found out that many DAD algorithms do perform worse than a random signal or an untrained network when using point adjust to compare the results, demonstrating its inherent flaw. Additionally, the benchmark datasets used by the research community to compare Deep Anomaly Detection algorithms have been criticized. Wu and Keogh argue that these benchmarks are flawed and create an illusion of progress due to issues such as triviality, unrealistic anomaly density, mislabeled ground truth, and run-to-failure bias[6].

3 Methodology

3.1 Deep Anomaly Detection

This study seeks to provide a comprehensive representation of the DAD research field, focusing on the two main anomaly detection criteria: reconstruction error and prediction error. This exploration aims to discern whether increasing algorithmic complexity directly corresponds to more precise anomaly detection. To do so, algorithms of varying complexity were chosen, each demonstrating different strengths and attributes in their performance. The selection process encompassed both traditional

and innovative anomaly detection techniques. One algorithm uses prediction error as its anomaly detection criterion, while the other two operate based on reconstruction error. Notably, one of the algorithms chosen introduces a novel anomaly detection metric, the 'association discrepancy', showcasing the current trend of inventing new techniques to enhance detection capabilities. Furthermore, one of the chosen algorithms comes from the same domain as the telemetry data of satellites, which is considered to provide a unique perspective to the research. The selection process also considered performance capabilities, with one algorithm chosen specifically for its impressive results in numerous instances.

After careful consideration, the three algorithms selected were: The Anomaly Transformer, USAD, and LSTM. The Anomaly Transformer, the most complex among the three, features a transformer network, a learnable Gaussian kernel, two-phase learning, and introduces its own 'association discrepancy'. USAD, a moderately complex model, necessitates the use of two autoencoders trained in two phases, one of which involves adversarial training. LSTM, on the other hand, represents the simpler end of the spectrum, involving the training of a basic LSTM network to fit the data. These choices ensure a wide-ranging exploration of the current state of Deep Anomaly Detection research.

3.1.1 Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy [19]

Xu et al. have proposed an "Anomaly Transformer" that utilizes the global representation capability of transformers to handle long sequences effectively. The authors use the self-attention mechanism to calculate "association discrepancy" between "prior association" and "series association". "Series association" refers to the association of a specific time-point with the entire data of the chosen sliding window, whereas "prior association" refers to the association with the adjacent region represented by a learnable Gaussian kernel. The assumption of the dataset is that anomalies are rare, and most of the data is "normal." Therefore, a normal data point would have a high association with the whole data series, while an abnormal point would have a higher association with adjacent points containing more abnormal patterns due to continuity. The difference between the prior association and the series association is called association discrepancy. A low discrepancy indicates an anomaly, while a high discrepancy indicates a normal point. The algorithm employs minimax association learning, which is depicted in Figure 1. In the "minimize" phase, the prior association is adjusted to approximate the series association and adapt to the temporal patterns to decrease the association discrepancy. In the "maximize" phase, the series association is optimized to increase the association discrepancy and focus more on the non-adjacent horizon.
This paper has demonstrated excellent results on benchmark datasets and is one of the first to utilize transformers. The authors use a new association-based detection criterion, which they pair with reconstruction error to obtain an anomaly score. They use the outlier fraction to determine the threshold.

The authors of the study concatenated individual anomaly sequences from the test datasets into a single dataset file [34], which raises two issues. Firstly, when calculating the metric score by averaging the results of individual anomaly sequences, it can lead to inflated scores, as explained further in Subsection 3.2. This can give the perception of better performance. Secondly, by concatenating the anomaly sequences, the dataset becomes discontinuous. The anomaly sequences are not necessarily related to each other and are from different times. This means that at the timestep where one sequence transitions to another, there will be a sudden difference in values. Datasets suffer from run-to-failure bias, where anomalies often occur towards the end of a sequence [6]. The sudden change in input at the transition point could lead to flagging the entire time-window around that point as an anomaly. While the algorithm correctly detects the anomaly, it may only be detecting the beginning of a new sequence rather than an actual anomaly. Another critique point is that the sliding window length is relatively small, consisting of only 100 data points which is only about 1% of the whole sequence in some cases of the SMAP and MSL dataset. In order to detect an association discrepancy between the dataset and adjacent time-points, a longer window would be necessary. However, extending the window would significantly increase the computational complexity of the transformer, as the calculation of query and key alone already has a time complexity of $O(N^2)$.

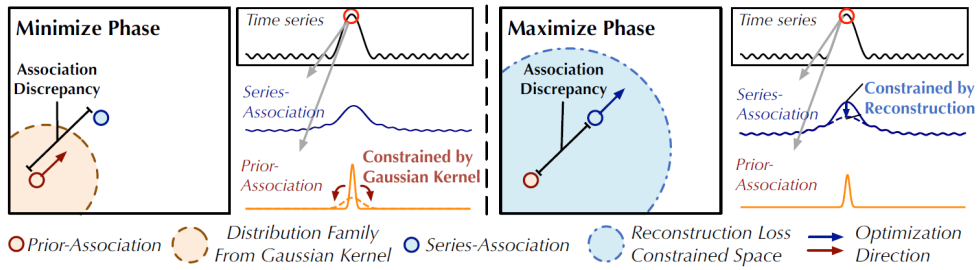


Fig. 1 Minimax association learning as shown by Xu et al. [19].

3.1.2 LSTM: Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding [16]

This algorithm utilizes LSTM and Nonparametric Dynamic Thresholding (NPT) to detect spacecraft anomalies. During training, the LSTMs are fitted to the normal operating data of the spacecraft to predict future telemetry data. However, the algorithm only predicts one channel (feature) of the data stream. Therefore it is is necessary to train multiple models when no single good predictor is available. For the real-life dataset that we introduce in this paper, the current channel was specifically selected as the target variable for prediction. This decision was based on the understanding that reaction wheel failures are frequently associated with friction, and it is anticipated that the current measurements would reflect this impact. Although a similar

impact was expected on temperature, the results were not as promising. The trained LSTMs are then used to generate anomaly scores by calculating the prediction error in the testing phase. In the testing phase, the NPT algorithm is used to set a threshold for the anomaly scores generated by the LSTMs. The threshold is dynamically adjusted based on the past anomaly scores, which allows for the detection of anomalies with varying degrees of severity. Additionally, error pruning techniques are used to ensure that anomalous sequences are not considered as a result of regular noise within a stream. This helps to filter out false positives and improves the accuracy of anomaly detection by focusing on significant deviations from normal patterns. Overall, this algorithm utilizes LSTMs to model the spacecraft's normal behavior and dynamic thresholding to detect deviations from the normal behavior, enabling the detection of spacecraft anomalies in real-time.

3.1.3 USAD: UnSupervised Anomaly Detection on Multivariate Time Series [30]

Audibert et al. have developed an algorithm that employs two adversarial autoencoder networks, inspired by GAN, to achieve high stability, robustness, and training speed without compromising accuracy. UnSupervised Anomaly Detection (USAD) consists of an encoder network and two decoder networks, which are combined into an architecture that includes two autoencoders, AE1 and AE2, sharing the same encoder network. The architecture can be seen in Figure 2. The training of USAD is carried out in two phases. In the first phase, the two autoencoders are trained to reconstruct the normal input windows. In the second phase, the two autoencoders are trained in an adversarial manner, where AE1 attempts to deceive AE2 while AE2 tries to distinguish between real and reconstructed data. The encoder-decoder network is trained on the normal data to learn the temporal patterns and correlations between the variables in the time series. The anomaly scoring mechanism then uses the reconstruction error of the network to generate an anomaly score for each data point. The adversarial training of the encoder-decoder architecture is shown to amplify the reconstruction error and improve stability compared to GAN methods. Additionally, the algorithm introduces a sensitivity threshold that can be adjusted without retraining the model to increase or decrease the detection sensitivity. They use grid search to determine the threshold that gives the best f1-score.

3.2 Metrics

The F1-score is a commonly used metric for evaluating time-series anomaly detection algorithms. However, recent studies in deep anomaly detection report high F1-scores, leading to a perceived increase in accuracy. It is important to note that most studies use a technique called Point Adjust before scoring the performance of an algorithm. This technique was first used by Xu et al. [2]. The principle of PA can be seen in Figure 3, where all instances in an anomalous sequence are considered true positives when at least one anomaly is detected within the sequence. This technique greatly amplifies the detection of true positives and artificially inflates the F1 score[3–5]. Hence they propose new metrics.

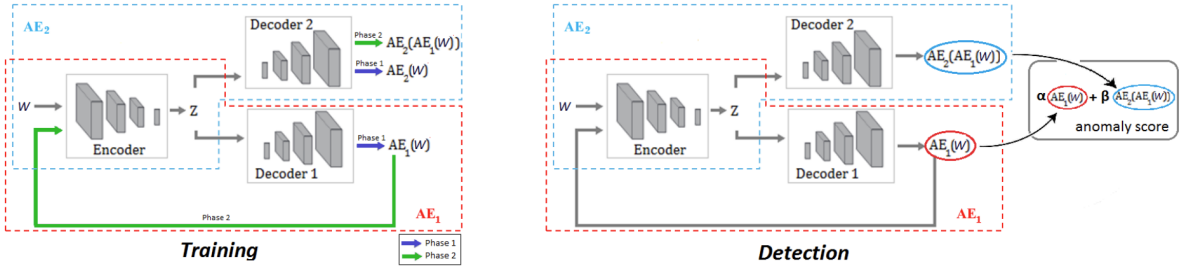


Fig. 2 Architecture of the USAD algorithm as proposed by Audibert et al. [30] illustrating the information flow at training (left) and detection stage (right).

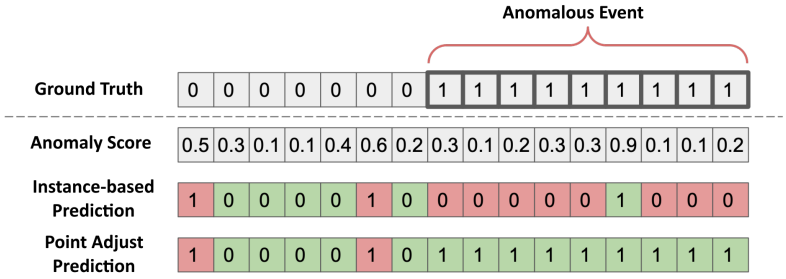


Fig. 3 This figure taken from Doshi et al. [4] demonstrates the comparison between the commonly used Point Adjust evaluation method and the traditional instance-based evaluation.

- F_{C1}-Score: According to Garg et al., a perfect anomaly detection algorithm should be able to detect at least one anomalous data point per anomaly event without any false positives[5]. Therefore, they proposed a new metric called Composite F-Score (F_{C1}). The F_{C1}-Score is calculated similar to the F_1-Score by taking the harmonic mean of precision and recall. However, the recall is calculated event-based instead of instance-based:
  $Recall_{event} = \frac{TP_e}{TP_e+FN_e}$

- $F1_{PA\%k}$: Kim et al. argue that PA overestimates detection accuracy while using F1-score without PA underestimates accuracy due to incomplete test set labeling. The authors suggest a new metric, called $F1_{PA\%k}$, which can address the problems of over- and underestimation. This metric is similar to PA, but it only considers an event as detected when the proportion of correctly identified instances in the event exceeds a threshold value k. If a user wants to remove the dependency on a specific threshold value k, it is recommended to measure the Area Under the Curve (AUC) of $F1_{PA\%k}$ by gradually increasing k from 0 to 100. This approach allows for a comprehensive evaluation of the model's performance across various threshold values, providing a more robust assessment of anomaly detection capability and is referred to as $F1_{PA\%k\text{-}AUC}$.

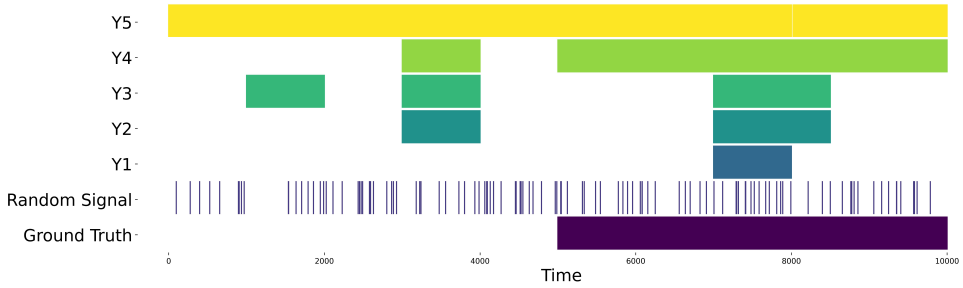


Fig. 4 Comparison of example detectors in Anomaly Detection using selected evaluation methods for anomaly detection. The corresponding metric scores can be seen in Table 2

Table 2 Comparison of Evaluation Methods for Anomaly Detection

Name	F1	F1 PA	Fc	$F1_{PA\%k=20}$	$F1_{PA\%k\text{-}AUC}$
Random	0.02	0.99	0.63	0.02	0.03
Y1	0.33	1	1	0.33	0.47
Y2	0.4	0.91	0.75	0.91	0.55
Y3	0.35	0.83	0.6	0.83	0.50
Y4	0.91	0.91	0.91	0.91	0.91
Y5	0.66	0.67	0.67	0.67	0.67

Although these new metrics are generally considered an improvement over the F1 metric with point adjust, they have their advantages and disadvantages. To evaluate which metrics perform well in different scenarios, several example signals were created and can be seen in Figure 4. The first simulated anomaly detector is a random signal with a 0.01 probability of flagging a timestep as an anomaly. Y1 to Y4 are constructed signals that correctly detect portions of the anomaly, and some of them also have false positive sequences during normal operation. Y5 is a special case where the alarm is raised the whole time except for one instance where a false negative occurs in the actual anomaly. The example signals can be seen in Figure 4 and their results for the four different metrics can be found in Table 2. As previously stated, the F1 score underestimates the capabilities of a detection algorithm. All scores are relatively low except for Y4, where most of the anomalous event was captured. When detecting an anomalous event, it is not essential whether the whole anomalous segment was detected or only a fraction, since an operator would investigate the system as soon as an alarm is raised. Therefore, one could argue that detector Y1 is better than Y4 as no false positives were raised. However, the F1 score does not capture this. Therefore, PA was created, as it was reasoned that an anomaly detector should have minimal false positives and detect an anomalous event. Looking at the F1 PA scores, it can be observed that the metric overestimates detection capabilities. All detectors achieve high scores, and the random detector almost reaches a perfect score. The F_{C1}-Score improves on this, as the deviation between a good detector (Y1) and a worse one (Y3) becomes more significant, and the random signal scores fewer points. The $F1_{PA\%k}$

metric rightly scores the random signal even lower. However, it is sensitive to the parameter k. In this case, a k of 20 was selected, and it can be seen that signal Y1 scored low because the fraction of detected anomalous instances in the anomaly event was less than 20%. The Signal Y5 can be regarded as a poor detector since it raises an alarm for almost all the time, yet none of the metrics appear to identify this. The $F1_{PA\%k\text{-}AUC}$ method is not sensitive to threshold selection however signals Y1, Y2 and Y3 perform worse than the bad signal Y5. In conclusion, the F_{C1}-Score is better than the other metrics while it still has problems with clearly marking the signal Y5 as a bad detector. Therefore the comparison of the results will be measured with the Composite F-Score in the remainder of this paper.

In order to obtain a final metric score for a dataset with multiple anomaly sequences, there are two options for combining the individual results: taking the average of the metric results for all anomaly sequences or summing up the classes of the confusion matrix and calculating the metric using this sum. However, the latter approach tends to give better results, contributing to the perceived increase in accuracy. The following example illustrates this: Figure 5 depicts a dataset comprising two dis-

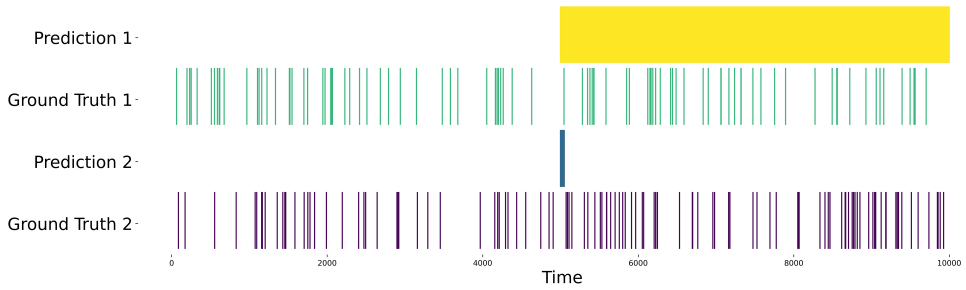


Fig. 5 Impact of Combining Metric Scores on Anomaly Detection Performance

tinct anomaly sequences. The first anomaly, accounting for approximately 50% of the anomaly sequence, is more readily detectable compared to the relatively brief second anomaly. A random signal, having a 0.01 probability, is applied as a detector, yielding a resultant $F1_{PA}$ score. Upon computation of the average $F1_{PA}$ score, the outcome is determined to be 0.497. The first anomaly garners a score of 0.995, while the second anomaly, due to a lack of true positives, scores 0. Conversely, when the total metric is computed via the summation of the confusion matrix elements, the score is substantially higher at 0.979. This elevation in score is attributed to the point adjust mechanism, which assigns the entirety of the first anomaly sequence as true positives. Given the longer duration of the first anomaly, it significantly contributes to the true positive and false negative classifications, thereby influencing the computed metric. This calculation methodology inherently undermines the significance of shorter and more challenging-to-detect anomalies. Consequently, the study opts for the utilization of the average score methodology.

3.3 Comparative evaluation approach

In this paper, the three selected deep anomaly detection methods are compared to several popular classic anomaly detection methods and three baseline methods. The four classic methods used for comparison are Gaussian Mixture Model (GMM), k-Nearest Neighbor (kNN), One-Class Support Vector Machine (OCSVM), and Principal Component Analysis (PCA). These methods have been widely used in anomaly detection research and serve as established benchmarks for comparison [35–38] and are implemented with PyOD [33]. Along with the selected algorithms, the importance of baseline testing in the evaluation of anomaly detection methods is highlighted, as underscored by the findings of Kim et al. [3]. They asserted that despite attaining high test scores, some detection methods might not exhibit improvement when compared to simple baseline methodologies. This underlines the significance of maintaining a fair and comprehensive evaluation process by including basic benchmark methods. Therefore, this study incorporates three baseline methods to ensure a robust comparative analysis of the performance of the selected algorithms.

- The Raw Input Method: This simple technique calculates the norm of the input vector, serving as a primary benchmark against more sophisticated methods.
- The Untrained Autoencoder: In this method, the weights of the Autoencoder are randomly initialized from a standard normal distribution.
- The Random Signal Method: Here, each instance has a probability of $p = 0.01$ of being flagged as an anomaly. This method, despite its inherent randomness, offers a basic statistical baseline against which the detection capabilities of other methods can be assessed.

The incorporation of these baseline methods in the analysis enables a comprehensive performance evaluation of the selected algorithms, setting them against both conventional techniques and simpler benchmark approaches. This facilitates a holistic understanding of their effectiveness in the Deep Anomaly Detection research field.

As mentioned in Subsection 2, most of the state-of-the-art algorithms utilize thresholding methods that suffer from data leakage, giving them an advantage. To ensure a fair comparison, the same thresholding approach was applied to all algorithms. A grid search was conducted, exploring all possible thresholds, to find the threshold that maximizes the F1-score. For the LSTM algorithm, this study reports the results for both thresholding methods: the original nonparametric method proposed by the authors and the best F1 method. This approach allows for a comprehensive comparison and evaluation of all algorithms using the same thresholding approach. It also provides an opportunity to assess the performance of the unsupervised NPT method and determine its effectiveness in anomaly detection.

Table 3 Key characteristics of the datasets used for this study

	ESA	**SMAP**	**MSL**
Total anomalies	7	69	36
Unique telemetry channels	10	55	27
Telemetry values evaluated	146,887	429,735	66,709

4 Experiments

4.1 Case Study Description

This section introduces the datasets used to evaluate various Deep Anomaly Detection algorithms. We use a dual-approach featuring real-world satellite telemetry data from the Sentinel-1 mission and recognized validation datasets such as the Mars Science Laboratory rover and Soil Moisture Active Passive satellite dataset provided by NASA. These diverse datasets facilitate a comprehensive assessment of the DAD methods, shedding light on their performance in both real-world and standardized scenarios. The subsequent subsections detail each dataset and its unique contribution to our study. An overview of the key characteristics of the datasets can be seen in Table 3.

4.1.1 Real-Life Satellite Telemetry Data of ESA Satellites

Satellites carry four reaction wheels, a component that presents a compelling use-case for applying machine learning in anomaly detection. The accumulated data from these identical reaction wheels—despite variations in satellite sizes—offers an opportunity for holistic prognostics. This data, relatively accessible, encapsulates a mechanical component's degradation over time. Key sensor readings, such as temperature, current, and speed, reflect changes in friction—usually the culprit behind reaction wheel anomalies or failures [39].

The Sentinel-1 mission, comprising two polar-orbiting satellites, Sentinel-1A and Sentinel-1B, launched in April 2014 and April 2015 respectively, was selected as the data source for this study. Its appeal lies in the existence of multiple sequences of anomalous reaction wheels, enabling label generation for testing algorithm efficacy. Seven anomalies, identified by ESA operational personnel, form the basis of our test sets. These are drawn from the data of seven days before and after each anomaly's onset, while the training sets comprise the preceding seven-day data.

The telemetry data, comprised of various features like current, temperature, speed, and torque, is captured at irregular intervals, but usually multiple times per minute. To address the irregularity, the data was averaged within one-minute intervals to provide a consistent timescale for analysis. There were two instances when no data was recorded—18 and 30 minutes long, respectively—wherein interpolation was used to fill the data gaps. Ultimately, the data was normalized based on the training set values.

4.1.2 Validation Datasets

To ascertain the efficacy of the proposed anomaly detection methodology, it's essential to benchmark its performance against datasets that are widely accepted within the research community. This step becomes particularly crucial considering the limited availability of labeled data and infrequency of anomalies in the use-case dataset. The Mars Science Laboratory (MSL) rover and Soil Moisture Active Passive (SMAP) satellite dataset, released by National Aeronautics and Space Administration (NASA), meets these requirements and serves as an ideal point of reference [16]. This dataset encompasses real-world spacecraft data, with input channels anonymized for security and privacy. It consists of a real-valued telemetry stream and binary commands, which are either sent or received by the corresponding subsystem. This data, labeled by domain experts, provides a robust foundation for evaluating the proposed anomaly detection approach.

4.2 Results

In this section, the results of the three DAD algorithms, the four classical algorithms, and the three benchmark methods are compared based on their performance on the three selected datasets. Additionally, a qualitative comparison is conducted on an example anomaly. Table 4 presents the results for the F_{C1}-Score. It can be observed that the LSTM algorithm achieves the best results or performs on par with other algorithms on all three datasets. Interestingly, when the LSTM algorithm utilizes non-parametric thresholding instead of the best F1-score, the F_{C1}-Score improves due to the filtering of false positives. This distinction may initially appear confusing. However, it is important to note that the best F1 method optimizes specifically for the F1-score and not the F_{C1}-Score and therefore scores higher in that metric as can be seen in Table 5. Additionally, it can be observed that the NPT method scores lower on the F1 metric. This is due to the fact that while it effectively reduces false positives, it does so at the expense of true positives.
Regarding the baseline methods, the results indicate that the raw signal method achieves results that are difficult for other methods to surpass, particularly on the European Space Agency (ESA) and SMAP dataset where the results are quite high. Comparing the other two deep learning methods to the raw signal baseline, it can be seen that the USAD algorithm outperforms the baseline on the MSL dataset. However, the Anomaly Transformer does not surpass the baseline results on any of the datasets.

Furthermore, it is observed that all of the classical methods outperform the Anomaly Transformer on all three datasets. This observation is intriguing as it suggests that the more complex methods, such as USAD and the Anomaly Transformer, do not perform well in comparison. Conversely, the simple deep learning algorithm, the LSTM, demonstrates excellent performance. This highlights the importance of considering the effectiveness of the algorithm design and complexity, rather than solely relying on the sophistication of the method.

Table 4 F_{C1}-Score for various methods: Bold indicates best result or in 2% from best result. ↑ is marked when score is higher than Baseline: Raw

Dataset	ESA	MSL	SMAP
LSTM	**1.000**(↑)	0.413 (↑)	0.579 (↑)
LSTM*	0.914 (↑)	**0.567**(↑)	**0.705**(↑)
USAD	0.772 (↓)	**0.574**(↑)	0.393 (↓)
Anomaly Transformer	0.336 (↓)	0.201 (↓)	0.263 (↓)
GMM	0.767 (↓)	0.473 (↑)	0.409 (↑)
KNN	0.778 (↓)	0.486 (↑)	0.382 (↓)
OCSVM	0.807 (↓)	0.549 (↑)	0.371 (↓)
PCA	0.820 (↑)	0.382 (↓)	0.355 (↓)
Baseline: Raw	0.820	0.387	0.404
Baseline: AE	0.558	0.346	0.344
Baseline: Random	0.468	0.191	0.183

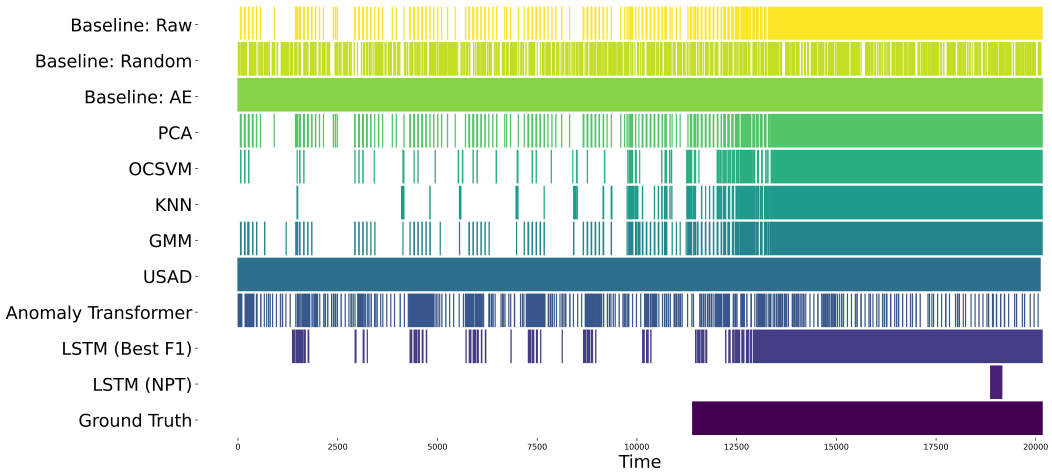


Fig. 6 Qualitative results of one anomaly of the ESA dataset.

In Section 3.2, it was established that the F_{C1}-Score is an improvement over other popular metrics but still has some limitations. Therefore, it is valuable to qualitatively examine the results of the predictions. Figure 6 illustrates an example anomaly from the ESA dataset along with the corresponding predictions made by the algorithms. This qualitative analysis provides further insight into the performance and behavior of the algorithms.

In the bottom of the figure, it can be observed that the anomaly starts at approximately 60% of the timeseries and continues until the end. The Baseline Autoencoder and the USAD algorithm raise an alarm for almost the entire sequence, thus failing to

Table 5 F1-Score for various methods: Bold indicates best result or in 2% from best result. ↑ is marked when score is higher than Baseline: Raw

Dataset	**ESA**	**MSL**	**SMAP**
LSTM*	**0.715**(↑)	**0.478**(↑)	**0.544**(↑)
USAD	0.617 (↓)	0.410 (↑)	0.286 (↓)
Anomaly Transformer	0.067 (↓)	0.089 (↓)	0.113 (↓)
GMM	0.695 (↑)	0.366 (↑)	0.308 (↑)
KNN	**0.713**(↑)	0.365 (↑)	0.260 (↓)
OCSVM	0.677 (↓)	0.391 (↑)	0.300 (↓)
PCA	0.664 (↑)	0.322 (↓)	0.270 (↓)
Baseline: Raw	0.664	0.324	0.313
Baseline: AE	0.494	0.292	0.262
Baseline: Random	0.082	0.056	0.013

Table 6 $F1_{PA}$-Score for various methods: Bold indicates best result or in 2% from best result. ↑ is marked when score is higher than Baseline: Random

Dataset	**ESA**	**MSL**	**SMAP**
LSTM	**1.000**(↑)	0.451 (↓)	0.612 (↓)
LSTM*	0.928 (↑)	0.643 (↓)	**0.787**(↑)
USAD	0.773 (↓)	**0.684**(↓)	0.517 (↓)
Anomaly Transformer	0.907 (↓)	0.568 (↓)	0.595 (↓)
GMM	0.835 (↓)	0.572 (↓)	0.519 (↓)
KNN	0.849 (↓)	0.630 (↓)	0.519 (↓)
OCSVM	0.882 (↓)	0.641 (↓)	0.468 (↓)
PCA	0.889 (↓)	0.471 (↓)	0.451 (↓)
Baseline: Raw	0.889	0.482	0.501
Baseline: AE	0.673	0.489	0.477
Baseline: Random	0.912	**0.686**	0.642

provide a useful signal. Surprisingly, both methods still achieve a F_{C1}-Score and F1-score of 0.61, highlighting the limitations of these metrics as discussed in Subsection 3.2.

The baseline random method, as expected, randomly raises alarms with a relatively consistent density throughout the sequence. On the other hand, the Anomaly Transformer raises flags throughout the entire sequence, with a seemingly higher density of alarms before the anomaly.

The raw signal, the four classical methods, and the LSTM algorithm using the best F1 threshold exhibit a more meaningful signal. They indicate fewer alarms before the anomaly while increasing the density after its onset. One could argue that the LSTM algorithm with the original unsupervised nonparametric thresholding, demonstrates the best signal. The anomaly is identified relatively late in the sequence; however, it is important to note that timeliness is not a focus of this research. It avoids false positives and successfully raises an alarm during the anomaly.
Furthermore, as evident in Table 4 and Table 5, it can be observed that the raw signal surpasses some of the anomaly detection methods in terms of performance. Moreover, Table 6 confirms that the point adjust metric is not reliable, as the random signal outperforms all methods except the LSTM algorithm.

5 Conclusions

This paper illuminates several flaws in current Deep Anomaly Detection research, revealing that performance results are often overestimated. Despite the escalating complexity of anomaly detection methods, simpler approaches and even baseline methods outperform some of the more intricate DAD algorithms. While the F_{C1}-Score represents an improvement over other metrics, the qualitative results indicate that it can still lead to inflated performance for poor detectors. Among the deep anomaly detection methods examined, the LSTM algorithm demonstrates highly promising results across all datasets and even achieves a perfect score on the real-life dataset. However, it's important to note that this dataset involves a relatively simple subsystem with a clear channel to predict failure, and the real-life dataset had relatively few anomalies due to the reliability of satellites in actual scenarios. The performance of the LSTM algorithm may not be as robust in complex systems lacking a single channel that reliably predicts failure, and a larger dataset is needed to bolster confidence in these results.

For the field to progress, several steps should be taken. Research should adopt best practices such as avoiding data leak, applying truly unsupervised algorithms throughout, abstaining from using measuring or averaging methods that inflate performance results, and applying proper data split methods.

For effective comparisons, the community needs high-quality, pre-split datasets and metrics that correlate with good performance. This research compared algorithms using the same thresholding, but future work should also compare different unsupervised thresholding methods.

For satellite operators, further research could explore whether other prediction error methods such as Convolutional Neural Networks and Transformers also perform well on telemetry data. Modifying the LSTM to predict more than one channel could enhance explainability and potentially improve performance. Moreover, it would be beneficial to determine which subsystems are best suited to these methods.

In conclusion, while our findings highlight the potential of deep learning in anomaly detection, they also underscore the need for a more critical and nuanced approach to evaluating its effectiveness in real-world applications. The ongoing advancement in deep learning and the ever-increasing volume of satellite telemetry data open up exciting avenues for further research and innovation in this field.

References

[1] Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: A review. ACM computing surveys (CSUR) **54**(2), 1–38 (2021)

[2] Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., *et al.*: Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In: Proceedings of the 2018 World Wide Web Conference, pp. 187–196 (2018)

[3] Kim, S., Choi, K., Choi, H.-S., Lee, B., Yoon, S.: Towards a rigorous evaluation of time-series anomaly detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 7194–7201 (2022)

[4] Doshi, K., Abudalou, S., Yilmaz, Y.: Tisat: time series anomaly transformer. arXiv preprint arXiv:2203.05167 (2022)

[5] Garg, A., Zhang, W., Samaran, J., Savitha, R., Foo, C.-S.: An evaluation of anomaly detection and diagnosis in multivariate time series. IEEE Transactions on Neural Networks and Learning Systems **33**(6), 2508–2517 (2021)

[6] Wu, R., Keogh, E.: Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. IEEE Transactions on Knowledge and Data Engineering (2021)

[7] Choi, K., Yi, J., Park, C., Yoon, S.: Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. IEEE Access **9**, 120043–120065 (2021)

[8] Hamilton, J.D.: Time Series Analysis. Princeton university press, Princeton (2020)

[9] Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407 (2019)

[10] Tuzlukov, V.: Signal Processing Noise. Electrical Engineering & Applied Signal Processing Series. CRC Press, Florida (2018). https://books.google.nl/books?id=x6hoBG_MAYIC

[11] Hawkins, D.M.: Identification of Outliers vol. 11. Springer, London (1980)

[12] Barnett, V., Lewis, T.: Outliers in statistical data. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics (1984)

[13] Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)

[14] Tang, Z., Chen, Z., Bao, Y., Li, H.: Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. Structural Control and Health Monitoring **26**(1), 2296 (2019)

[15] Basora, L., Olive, X., Dubot, T.: Recent advances in anomaly detection methods applied to aviation. Aerospace **6**(11), 117 (2019)

[16] Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 387–395 (2018)

[17] Yairi, T., Nakatsugawa, M., Hori, K., Nakasuka, S., Machida, K., Ishihama, N.: Adaptive limit checking for spacecraft telemetry data using regression tree learning. In: 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583), vol. 6, pp. 5130–5135 (2004). IEEE

[18] Shen, L., Li, Z., Kwok, J.: Timeseries anomaly detection using temporal hierarchical one-class network. Advances in Neural Information Processing Systems **33**, 13016–13026 (2020)

[19] Xu, J., Wu, H., Wang, J., Long, M.: Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642 (2021)

[20] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)

[21] Bank, D., Koenigstein, N., Giryes, R.: Autoencoders. arXiv preprint arXiv:2003.05991 (2020)

[22] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)

[23] Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., Zhang, Q.: Multivariate time-series anomaly detection via graph attention network. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 841–850 (2020). IEEE

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural

information processing systems **30** (2017)

[25] Broadwater, J.B., Chellappa, R.: Adaptive threshold estimation via extreme value theory. IEEE Transactions on signal processing **58**(2), 490–500 (2009)

[26] Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., Veeramachaneni, K.: Tadgan: Time series anomaly detection using generative adversarial networks. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 33–43 (2020). IEEE

[27] Zhou, B., Liu, S., Hooi, B., Cheng, X., Ye, J.: Beatgan: Anomalous rhythm detection using adversarially generated time series. In: IJCAI, vol. 2019, pp. 4433–4439 (2019)

[28] Li, D., Chen, D., Jin, B., Shi, L., Goh, J., Ng, S.-K.: Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In: Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV, pp. 703–716 (2019). Springer

[29] Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2828–2837 (2019)

[30] Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A.: Usad: Unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3395–3404 (2020)

[31] Chen, Z., Chen, D., Zhang, X., Yuan, Z., Cheng, X.: Learning graph structures with transformer for multivariate time-series anomaly detection in iot. IEEE Internet of Things Journal **9**(12), 9179–9189 (2021)

[32] Goh, J., Adepu, S., Junejo, K.N., Mathur, A.: A dataset to support research in the design of secure water treatment systems. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10242 LNCS**, 88–99 (2017) https://doi.org/10.1007/978-3-319-71368-7_8

[33] Zhao, Y., Nasrullah, Z., Li, Z.: Pyod: A python toolbox for scalable outlier detection. Journal of Machine Learning Research **20**(96), 1–7 (2019)

[34] Xu, J., Wu, H.: Code Repository for Anomaly-Transformer (2022). https://github.com/thuml/Anomaly-Transformer

[35] Reynolds, D.A., et al.: Gaussian mixture models. Encyclopedia of biometrics **741**(659-663) (2009)

[36] Mucherino, A., Papajorgji, P.J., Pardalos, P.M.: k-Nearest Neighbor Classification, pp. 83–106. Springer, New York, NY (2009). https://doi.org/10.1007/978-0-387-88615-2_4 . https://doi.org/10.1007/978-0-387-88615-2_4

[37] Schölkopf, B., Williamson, R.C., Smola, A., Shawe-Taylor, J., Platt, J.: Support vector method for novelty detection. Advances in neural information processing systems **12** (1999)

[38] Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., Chang, L.: A novel anomaly detection scheme based on principal component classifier. Technical report, Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering (2003)

[39] Bialke, W., Hansell, E.: A newly discovered branch of the fault tree explaining systemic reaction wheel failures and anomalies. In: Proceedings of the European Space Mechanisms and Tribology Symposium, pp. 20–22 (2017)

II

Literature Study
previously graded under AE4020

1

Introduction

Satellites are complex and high-value system that generate growing amounts of telemetry data that can be processed by ground systems. Monitoring this telemetry data is an important task to ensure the mission success of spacecraft operations. Anomaly detection is an important tool to prevent spacecraft loss due to undetected faults or late response to potential hazards. However, current anomaly detection methods consist predominantly of out-of-limit approaches with predefined thresholds and manual analysis of aggregated data. These methods have severe downsides as they need major expert knowledge and are very labour extensive to define and analyse the data. Furthermore they need to update these thresholds throughout the mission as the environment and conditions are dynamic. Furthermore operations engineers might miss critical information that gets lost in the aggregation of telemetry data. The outlook is that the above mentioned problems will get more severe in the future. Increased computing and storage capabilities will lead to more telemetry data that would require more engineering resources and higher levels of data aggregation.
Recent advances in neural network architecture and increasing computational power led to promising results in using deep learning for detecting anomalies in high dimensional time-series data. Various deep learning algorithms have been developed that seemingly outperformed classical anomaly detection methods on real world time-series problems [28]. For some benchmark-datasets F1 scores have been reported that exceed 0.9, giving the impression of very accurate deep anomaly detection (DAD) capabilities. However, most of the recent DAD research use a technique called Point Adjust (PA) which was first proposed by Xu et. al [40]. Multiple recent papers have been critical towards this method and raised concern that this overestimates the accuracy of anomaly detection methods [10, 11, 22]. In addition to that Wu and Keogh showed that many of the current datasets suffer of one or more of four flaws. The datasets are often trivial, have unrealistic anomaly densities, have wrong labels and suffer from run-to-failure bias [39].

2

Properties of Time Series Data

The foundation for anomaly detection algorithms is the data they draw conclusions from. A special case of input data is time series where there is a sequence of one or more time dependent variables. Understanding the properties of time series data can help successfully taking advantage of the contextual information hidden in them. The fundamental factors that will be further explained are: temporality, dimensionality, nonstationarity, and noise [9].

2.1. Temporality

A time series is a collection of observations indexed by the date of each observation [15]. Often they are captured at equal time intervals although also unequal spaced time series data exists. The sequence of data points therefore has an order, where each successive data point in the series depends on the past values. Using the product rule a joint distribution of sequence of data can be written as:

$$p\left(x^1, x^2, \ldots, x^T\right) = p\left(x^1\right) \prod_{t=2}^{T} p\left(x^t \mid x^1, x^2, \ldots, x^{t-1}\right) \tag{2.1}$$

Where x^t is a data point observed at time t and each conditional probability $p(\cdot|\cdot)$ indicates the temporal dependence between current state and previous ones [9].

2.2. Dimensionality

Dimensionality indicates the amount of features that are captured in a single observation. Choi et al. differentiate between Univariate, that means data with one variable, and Multivariate data which has two or more variables [9]. Chalapathy and Chawla distinguish between low-dimensional and high-dimensional data [6]. The dimensionality of data drives computational costs and influences the choice of analysis method.
Univariate data describes an ordered set of real-valued observations. Anomaly detection algorithms for univariate data therefore only consider temporal dependence. Multivariate data on the other hand describe an ordered set of multidimensional vectors. They too can capture the temporal dependence but additionally the correlation between different features should be considered. Multivariate data can be seen as a group of univariate data streams and has been modelled like this in the past. However analyzing multivariate time series data has become typical. The dimensionality of the input data also has implications for the number of layers used in deep anomaly detection as deeper networks produce better results on high dimensional data [6].

2.3. Nonstationarity

A stationary time series is one whose properties do not depend on the time at which the series is observed. Or, put in other words, a time series is stationary if its statistical properties of the observed series do not depend on the time of observation. For any $\tau \in \mathbb{Z}$, $t \in \mathbb{Z}^+$ and $F_{\mathbf{x}}$ being the joint distribution function, a discrete stochastic process for which the index variable takes a discrete set of values, is defined as strongly stationary if the following equation is satisfied:

$$F_{\mathbf{x}}\left(x^{1+\tau}, \ldots, x^{t+\tau}\right) = F_{\mathbf{x}}\left(x^1, \ldots, x^t\right) \tag{2.2}$$

While stationary processes are easier to analyze, most real world data is subject to fluctuating effects. Examples of effects that make time series nonstationary are seasonality, concept drift and change points.

- **Seasonality:** This represents a recurrent and periodic pattern over a limited time scale. Examples from the aerospace industry could be flight demand fluctuating throughout the year due to weather and holidays or the cycles of the power system of a satellite that is subject to eclipse.
- **Concept Drift:** In nonstationary real world problems the properties of the target value that is predicted by the model changes over time [36]. This change of the statistical data distribution over time can deteriorate the performance of a model trained on historical data.
- **Change Points:** The industrial application of anomaly detection often observes physical equipment. The operational conditions of this machinery might change due to different settings, operational modes or wear out of the equipment.

Nonstationary data introduce additional complexity in anomaly detection since data samples might falsely be labeled as anomaly. Therefore anomaly detection methods need to adapt to changes in data distribution for long term deployment.

2.4. Noise

Noise is an undesired random perturbation of a useful information system. More explicit it is a general term for unknown and unwanted alterations of a information signal during capture, storage, transmission, processing or conversion [37]. Noisy systems can make separation between anomalous samples and noise difficult. Thus, as noise seriously affects performance of detection algorithms, it is crucial to reduce noise during the data preparation stage.

3

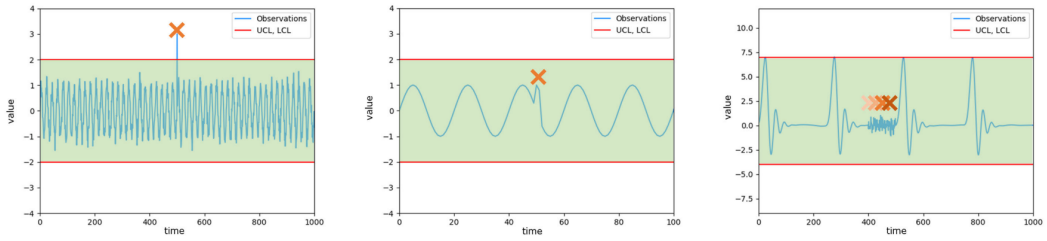
Anomalies

Numerous attempts to describe the essence of anomalous data were made in the literature. Hawkins definition of an anomaly is: “an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism” [17]. According to Barnett and Lewis’ an anomaly is “an observation or subset of observations, which appears to be inconsistent with the remainder of that set of data” [2]. Anomalies are also referred to as rare events, abnormalities, deviants, or outliers. The following two sections explain different categories of anomalies in univariate time series found in literature. However, it may not be valid to classify multivariate time series in the same way. Multivariate datasets require additional attention on the inter-correlation between the variables. With increased number of variables the anomaly pattern can become more complex. Multivariate time series can be decomposed into channels of the individual variables therefore constructing multiple univariate time series. There are examples in the literature where this is done and the alarms of the different channels are aggregated to one output [19]. However, in theorey is this not as accurate as observing the multivariate data since the correlation between variables is not considered [9].

In general, anomalies in time-series are categorized in three categories: Point anomalies, contextual anomalies and collective anomalies [7]. They are visualized in Figure 3.1 and are explained in detail in the following list:

- **Point Anomalies:** These are data points or sequences that suddenly diverge from the normative values. They are often caused by sensor errors or abnormal system operations. An example can be seen in Figure 3.1(a). They can relatively easy be spotted by simple technique’s like limit checking which uses a fixed upper control limit (UCL) and lower control limit (LCL) as boundaries. More advanced algorithms can use dynamic limits [16].
- **Contextual Anomalies:** This refers to data instances that are anomalous in a specific context but could be normal otherwise as can be seen in figure 3.1(b). Contextual Anomalies are also referred to as conditional anomalies and they posses two kind of attributes:
  - Contextual Attributes are used to determine the context of the data instance. For example for a satellite the operating mode is an important context for analyzing failures in reaction wheels (RW). For example, RWs are expected to be operated more during attitude acquisition mode as opposed to the nominal mode.
  - Behavioural attributes describe the non-contextual characteristics of an instance. Staying with the example of the reaction wheels: The measured current for any mode is a behavioral attribute.
- **Collective Anomalies:** This is defined as a collection of related data points that is anomalous with respect to the entire dataset. Individually the data instances might not raise suspicion but their occurrence as a collection is anomalous. An example can be seen in figure 3.1(c).

While the three categories of anomalies can be generalized to every time series dataset one can define more detailed subsets of anomalies. These refined categories depend on the domain the analysis is applied to. One example can be the six anomaly patterns Tang et al. defined to classify vibration anomalies [35]. The seven classes, including one class for normal data can be seen in Table 3.2.



(a) point anomaly (b) contextual anomaly (c) collective anomaly

Figure 3.1: Anomaly types in time series data [9].

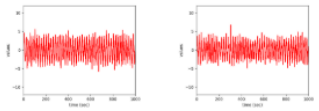
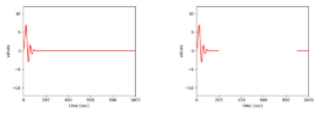
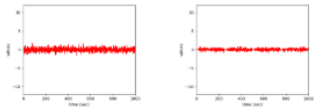
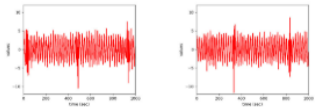
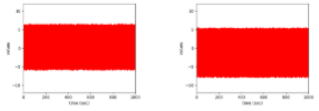
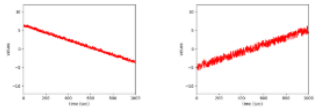
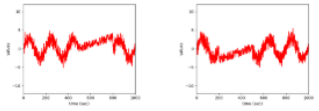
Anomaly patterns	Description	Examples
Normal (assumption)	The amplitude and frequency are stable over time steps, and the time response is symmetrical	
Missing	Most/all of the data are missing, and the time/frequency response becomes 0	
Minor	Compared to normal sensor data, the vibration amplitude is very small	
Outlier	One or more outliers appear in the time response	
Square	The time response oscillates within a limiting range like a square wave	
Trend	The data has an obvious non-stationary and monotonous trend	
Drift	The vibration response is nonstationary, with random drift	

Figure 3.2: Detailed classification of anomalies in time series according to Tang et al. [9].

4

Deep Learning

Deep learning (DL) has been successfully used in multiple areas of modern life such as speech recognition, natural language processing, computer vision and bio-informatics. Deep learning is a class of machine learning algorithms and allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [24]. The first developments can be attributed to F. Rosenblatt who modelled a perceptron, imitating a single neuron already in 1958 [29]. In recent years the amount of available data and computational power increased which led to DL models reaching human performance level for some use-cases, in some even to surpass it. These changes also sparked extensive research in the field of prognostics. In the following some popular DL architectures will be explained. In section 4.1 a single perceptron is explained which is the smallest building block for many popular neural networks. Multiple connected perceptrons are called a feed forward neural network which is explained in Figure 4.2. The Convolutional Neural Network and the Long Short Term Memory network are explained in section 4.3 and section 4.4 and are followed by the Autoencoder in section 4.5 and the Variational Autoencoder in section 4.6. Finally this chapter concludes with the Generative Adversarial Networks in section 4.7 and the Transformer in section 4.8.

4.1. Single Layer Perceptron

Structure and name of the Neural Network are inspired by the brain, mimicking the biological process of learning in humans. Neural Networks, also known as Artificial Neural Networks consist of an input layer, one or more hidden layers and an output layer. These ANN are further distinguished by grouping them into shallow neural networks, typically consisting of one hidden layer and deep neural networks which have multiple hidden layers. Layers are made up of several nodes. One single node including its connections is called perceptron and mimics the human neuron. In Figure 4.1 it can be seen how the perceptron processes input and arrives at an output. First the inputs are multiplied with their respective weights, representing the specific strength of this connection. Then the sum of these multiplications is added to the bias of the node. Then the activation function determines the level of activation for this node. In this example the activation function is binary which is usually not used because the gradient of the step function is zero and therefore back-propagation can not be used. Examples of typical activation functions employed are the Hyperbolic Tangent (tanh), Rectified Linear Unit (ReLU) and logistic function (Sigmoid).

4.2. Feed Forward Neural Network

Connecting multiple perceptrons by using the output of a node as an input to the next one is called a Multi-Layer-Perceptron (MLP) or Feed Forward Neural Network (FFNN). The goal of a FFNN is to approximate a function f^* and to define a mapping $y = f(x;\theta)$. Training the network has the purpose to learn the value of θ that approximates the function best. The model is called feedfoward because information is directed purely forward, with no feedback connections which loop the model output back. Feedforward neural networks are called networks because they consist of many functions that are chained together. For instance one could have three functions $f^{(1)}$, $f^{(2)}$ and $f^{(3)}$ which build a chain in the form of $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$. In this case $f^{(1)}$ would be the first layer, $f^{(2)}$ the second layer and so on. One speaks of the depth of the model when considering the number of layers used. The number of perceptrons per layer determine the width per layer.

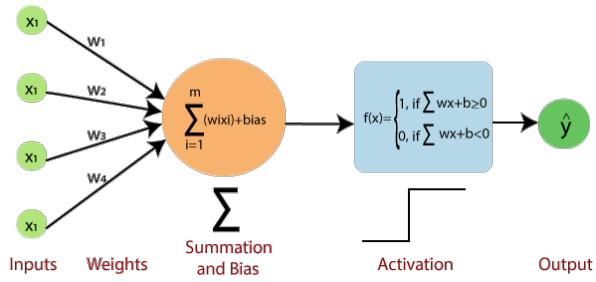


Figure 4.1: Schematic of a Single-Layer-Perceptron.

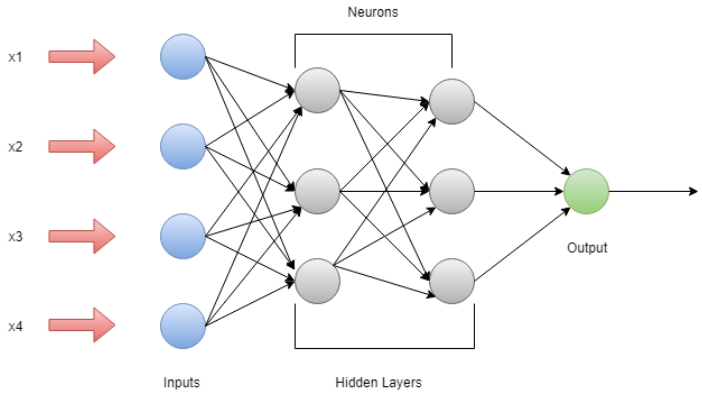


Figure 4.2: Schematic view of a Feedforward Neural Network [20].

An example of a FFNN with a depth of two and a width of three can be seen in Figure 4.2. There are multiple design decision to consider when building a feedforward neural network. One has to choose the architecture, that means the width and the depth of the model as well as how the nodes are connected. Furthermore the optimizer has to be chosen, the activation function that is used in the hidden layers and finally the cost function. Optimizers are algorithms that change the parameters of the network such as weights and biases. A well known algorithm is stochastic gradient descent (SGD) which tries to minimize the loss function by taking small steps in the direction opposite to the gradient. More sophisticated algorithms like Adaptive Moment Estimation (ADAM) are more common now because of their ability to converge faster towards the minima.

4.3. Convolutional Neural Network

Convolutional Neural Networks are used to process grid like data. They are very famous for their use on 2-D data in images however they can also be used in 1-D, such as regular time series data. They obtain their name from the mathematical operation called convolution. Neural networks are therefore called Convolutional Neural Networks if they use convolution instead of general matrix multiplication in at least one of the layers. A convolution is defined as:

$$s(t) = \int x(a)\, w(t-a)\, da. \tag{4.1}$$

Here x is the input and w is the kernel. The output is often referred to as the feature map. The convolution is typically denoted with an asterix:

$$s(t) = (x * w)(t) \tag{4.2}$$

Usually we do not have continous data, for example images have pixels and time sequence data is discrete. Then we use discrete convolution which is defined as:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)\, w(t-a) \tag{4.3}$$

The process of convolution is depicted in Figure 4.3. It can be seen that a [2,2] Kernel is sliding over [4,3] data which results in a [3,2] output because no padding is used. Padding is procedure of extending data at the

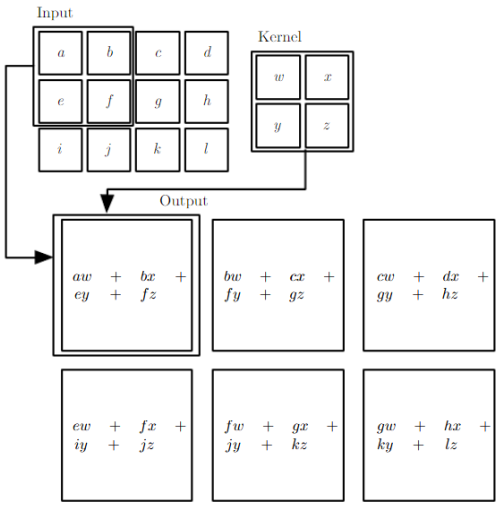


Figure 4.3: An example of a 2-D convoltion on 4x2 data with a 2x2 Kernel without padding [13].

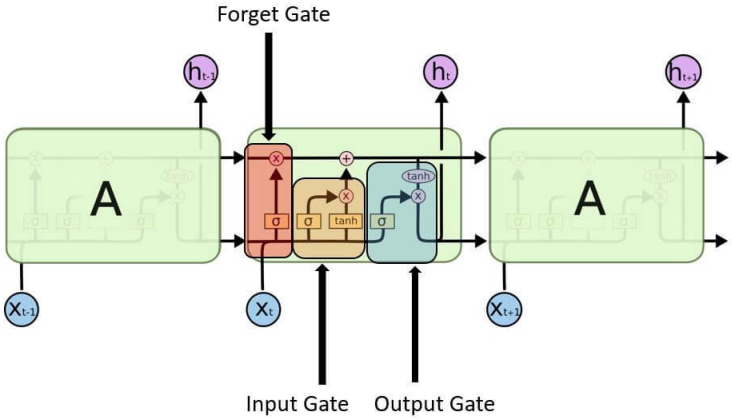


Figure 4.4: Long Short Term Memory Network [20].

edges with artificial data in order to prevent the shrinkage from input to output space because of convolution.

4.4. Long Short Term Memory networks

Feed Forward Neural Networks have difficulties processing sequential input like for example time series data. An answer to this was the Recurrent Neural Network (RNN) which was first brought up in 1986 [30]. A RNN is able to recursively handle the input, using a hidden state which can look into the past. However, do to adding a multitude of layers in series, this architecture has the problem of vanishing gradients. To overcome this problem the Long Short Term Memory (LSTM) network was proposed in 1997 [18]. One LSTM cell consist of three gates, the forget gate, the input gate and the output gate as can be seen in Figure 4.4.

- **Forget Gate:** This part decides which values in the cell state should be discarded. The previous hidden state h_{t-1} and the current input x_t are used in a sigmoid function which takes values between 0 (forget the value) and 1 (keep the value). The results are then multiplied with the previous cell state.
- **Input Gate:** This part decides which values from the input should be added. Again, a sigmoid function determines which value to let through while a tanh function establishes the associated weights in the range from -1 to 1.

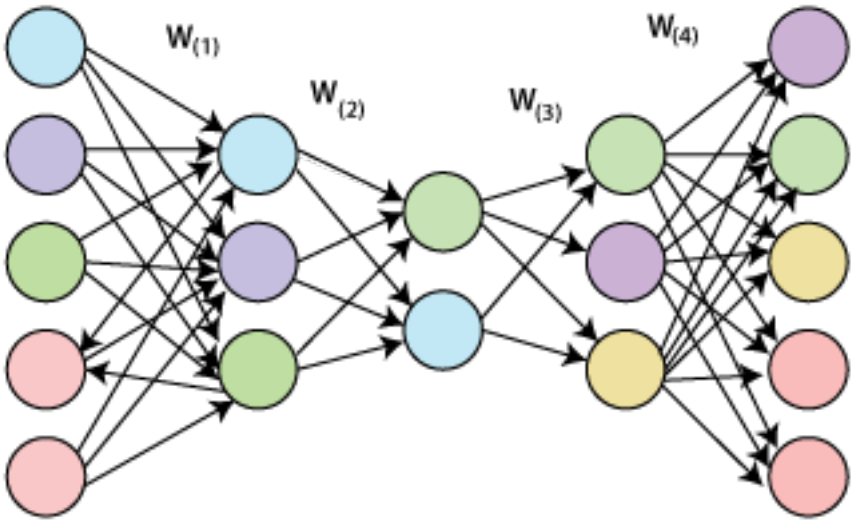


Figure 4.5: Schematic of an Autoencoder.

- **Output Gate:** This part decides which values will be in the output. A sigmoid function decides which values of the hidden state and input should be let through while a tanh function decides on the respective weights that are multiplied with the output of the sigmoid.

4.5. Autoencoder

Autoencoders are a type of artificial neural network that is trained to match its output to the input. It does so by learning an encoding in the latent layer h which represents the input internally. The network consists of two parts, the encoder $h = f(x)$ and a decoder $r = g(h)$ that produces the reconstruction r. Autoencoders do therefore not need labeled data and they can use the classic back-propagation algorithm. The latent layer typically is of lower dimension than the input space which is also called undercomplete. This forces the network to learn the most important features of the input by minimizing the loss between input and output.

These networks are used for data reconstruction, dimensionality reduction and feature learning. An example of an autoencoder structure with three hidden layers can be seen in Figure 4.5.

4.6. Variational Autoencoder

A variational autoencoder (VAE) is a neural network which architecture looks similar to an autoencoder, namely consists of an encoder and decoder part. It was first introduced by Kingma and Welling in 2014 [23]. While the autoencoder is a deterministic model, the variational autoencoder uses a stochastic generative model and can therefore also be used for generative processes. It is also called a directed probabilistic graphical model (DPGM). Variational inference (VI) is used to approximate the posterior which gives VAE the name. The posterior is a part of Bayes law and is expressed as $p(zx)$. The goal is to infer good values for the latent variables z given the samples x. Just as with the autoencoder, the encoder network transforms the input into a smaller latent space also called the bottleneck. However now the latent space is stochastic and the parameters of a Gaussian distribution are approximated. The latent representation is then sampled from that distribution and can be decoded from the second part of the network. The loss function of a VAE consists of a reconstruction term and a regularization term. The reconstruction term tends to make the network as accurate as possible. The regularization term is called the Kullback-Leibler divergence between the prior and posterior distribution which is derived using a technique called variational interference. It organizes the latent space as close to the standard Gaussian distribution. The parameters of the models are optimized bu using stochastic gradient decent (SGD). SGD is performed on the so evidence lower bound (ELBO) which is also called variational lower bound. Stochastic gradients which are necessary for SGD are obtained with a reparameterization trick.

4.7. Generative Adversarial Networks

Generative Adversarial Networks or GANs use a generative modeling approach that is differentiable. It is based on a game theory approach where the generator network competes against an adversary network called the discriminator network. The generator network produces samples $x = g(z;\Theta^{(g)})$ given the weights and biases of the generator $\Theta^{(g)}$ and using a random sample from a distribution z which could be normal or uniform. Its counterpart, the discriminator, seeks to differentiate between real samples from the training data and the samples that the generator produced. The trained parameters of the discriminator network are called $\Theta^{(d)}$. The probability value that the discriminator determines is defined by $d(x;\Theta^{(d)})$, giving the probability that the input x is real. GANs are often defined as minimax game where the discriminator D wants to maximize the objective V and the generator G wants to minimize it. A default choice for the objective is:

$$V\left(\boldsymbol{\theta}^{(G)},\boldsymbol{\theta}^{(D)}\right) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))] \tag{4.4}$$

This motivates the discriminator to learn to correctly distinguish between real and fake samples. For real samples the true labels $D(x)$ should be one and and for the fake case $G(z)$ zero. Therefore the objective for the discriminator becomes:

$$\max_D V(D) = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]}_{\text{Recognize real samples better}} + \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]}_{\text{Recognize fake samples better}} \tag{4.5}$$

The generator on the other hand tries to learn to trick the discriminator into classifying fake samples as real.

$$\min_G V(G) = \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]}_{\text{Fool generator into believing the samples are real}} \tag{4.6}$$

This can be written as a minimax game where the generator wants to minimize V while the discriminator wants to maximize it:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))] \tag{4.7}$$

GANs have a number of common failure modes such as the vanishing gradient problem, mode collapse and failure to converge. These failure modes are areas of active research and there are modifications of the algorithm that can help to overcome these limitations. If the discriminator is too good then the training can fail due to vanishing gradients. The original paper [14] modified the minimax loss to fix this and other sources use the Wasserstein loss which is designed to prevent the vanishing gradients. Mode collapse is when the generator fails to produce a wide variety of outputs that is as diverse as the distribution of real world data. Instead the generator produces a set of outputs that are very similar. The already mentioned Wasserstein loss can help with this problem. Furthermore Metz et al. [26] developed a Unrolled Generative Adversarial Network that solves mode collapse. Finally the last common problem is failure to converge. As the generator gets more proficient with training, the discriminators performance gets worse because it gets harder to tell real and fake data apart. Imagining a perfect generator, the discriminator would have 50% accuracy, effectively flipping a coin to predict the label. Thus, as the discriminators performance gets worse over time, the feedback it provides deteriorates as well. If the training continues at this point it might happen that the generators performance drops as well as it uses the random feedback from the discriminator. Various forms of regularization have been used to improve GAN convergence, for example adding noise to the discriminator inputs or penalizing discriminator weights.

4.8. Transformer

Transformers are designed to handle sequential input data. Popular applications are natural language processing and computer vision. Transformers use the mechanism of self-attention which will be explained here first. Self-attention is a technique that mimics cognitive attention and decides which parts of the input is more or respectively less important for the task at hand. Prior to self-attention, sequence modelling was mostly performed with RNN and CNN type of architectures. It was discovered that their performance could be improved by adding self-attention. Later was discovered that self-attention itself was already able to learn the task at hand [38].

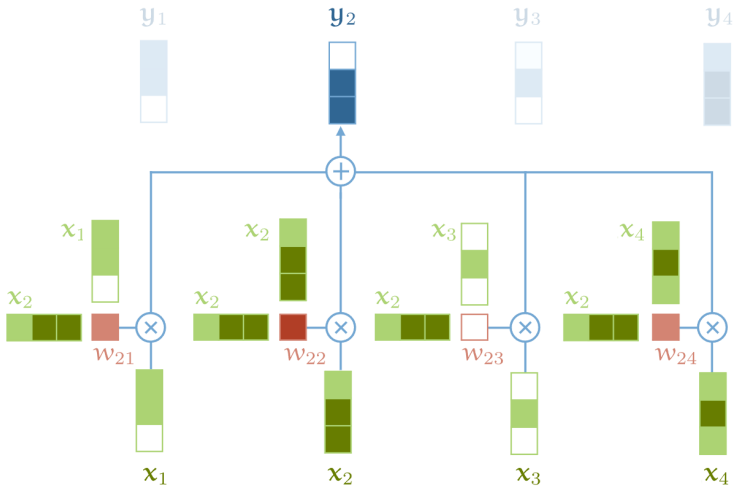


Figure 4.6: Visualisation of self-attention, excluding the softmax operation [4].

4.8.1. Self-attention

Self-attention is an operation that takes a sequence of input vectors $\boldsymbol{x}_n$ and maps them to a sequence of output vectors $\boldsymbol{y}_n$. This is be done by taking a weighted averaged of the input vectors and is defined as:

$$\boldsymbol{y}_i = \sum_j \boldsymbol{w}_{ij} \boldsymbol{x}_j \tag{4.8}$$

The weight $\boldsymbol{w}_{ij}$ is not a parameter, unlike in the neural networks. Before obtaining it an intermediate step is necessary to calculate $\boldsymbol{w}'_{ij}$. It is a function of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The most basic form of this is simply a dot product:

$$\boldsymbol{w}'_{ij} = \boldsymbol{x}_i^\top \boldsymbol{x}_j \tag{4.9}$$

The results of the dot product can be between positive and negative infinity. Applying a softmax they are mapped to between zero and one and sum up to one for the whole sequence:

$$\boldsymbol{w}_{ij} = \frac{\exp \boldsymbol{w}'_{ij}}{\sum_j \exp \boldsymbol{w}'_{ij}} \tag{4.10}$$

A visualisation of this basic operation of self-attention can be found in Figure 4.6. This basic operation does not yet have parameters that can be learned. The learning process is defined by upstream mechanisms, for example an embedding layer. However the self-attention that is used in transformers makes use of three additional tricks: Transformation of the input vector depending on its role, scaling of the dot product and multi-head attention.

Linear transformation of input: The first trick is the use of queries, keys and values. In the the self-attention operation every input vector $\boldsymbol{x}_i$ is used for three different things:

1. **Query:** It is used with all other input vectors to form the weights of its own output $\boldsymbol{y}_i$, see example in Figure 4.6 for $\boldsymbol{x}_2$ and $\boldsymbol{y}_2$

2. **Key:** It is used with all other input vectors to form weights of their respective output.

3. **Value:** It is used as part of the softmax operation as part of the weighted sum once all weights have been established.

In the basic self-attention, explained above, every input vector $\boldsymbol{x}_i$ had to fill these three roles. Adding independent linear transformation parameters, $\boldsymbol{W}_q, \boldsymbol{W}_k$ and $\boldsymbol{W}_v$ for the three different roles makes it easier to

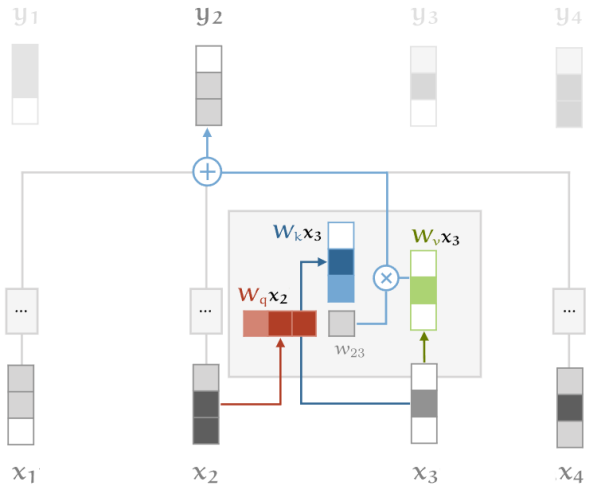


Figure 4.7: Self-attention with linear transformation for query, key and value role [4].

learn a better representation to suit them. The query, key and value vectors are then defined as follows:

$$\begin{aligned} \mathbf{q}_i = \boldsymbol{W}_{\mathrm{q}}\boldsymbol{x}_{\mathrm{i}} \quad \mathbf{k}_{\mathrm{i}} = \mathbf{W}_{\mathrm{k}}\boldsymbol{x}_{\mathrm{i}} \quad \boldsymbol{v}_{\mathrm{i}} = \mathbf{W}_{v}\boldsymbol{x}_{\mathrm{i}} \\ \boldsymbol{w}'_{\mathrm{ij}} = \mathbf{q}_{\mathrm{i}}^{\top}\mathbf{k}_{\mathrm{j}} \\ \boldsymbol{w}_{\mathrm{ij}} = \mathrm{softmax}\left(\boldsymbol{w}'_{\mathrm{ij}}\right) \\ \mathbf{y}_{\mathrm{i}} = \sum_{j} \boldsymbol{w}_{\mathrm{ij}}\boldsymbol{v}_{\mathrm{j}} \end{aligned} \tag{4.11}$$

The query, key and value technique applied to the basic architecture explained before can be seen in Figure 4.7.

Scaling the dot product: The second technique is the scaling of the dot product. The exponential part in Equation 4.10 scales with a higher input. This is due to the nature of the exponential function which grows faster the bigger the input. The dot product in Equation 4.9 growths larger with the embedding dimension k, which is the dimension of the input and output vectors. If the input values grow too large the softmax is getting dominated by single vectors. This minimizes the gradient which can result in slow or no learning at all. To scale the inputs to the inputs we simply divide by the squareroot of the dimension length:

$$w'_{ij} = \frac{\mathbf{q}_i^{\top}\mathbf{k}_{\mathrm{j}}}{\sqrt{\mathrm{k}}} \tag{4.12}$$

Multi-head attention: The final technique is the use of multiple attention heads in parallel. Each attention head produces its own set of output vectors $\boldsymbol{y}_n$ which according to the authors "allows the model to jointly attend to information from different representation subspaces at different positions" [38]. However by applying h parallel attention layers the computational cost also increases by h. The authors suggest cutting the input dimensions in h chunks and feeding only one of these chunks to each head, roughly keeping the computational cost the same.

4.8.2. Transformer architecture

Now that the self-attention operation is explained the transformer architecture that uses this layer can be further illustrated. While there is no rule for what does or does not qualifies as a transformer architecture an approach is used more commonly to use self-attention in larger networks.

As can be seen in Figure 4.8 this block consists of the self-attention layer, a normalization layer, a Multi-Layer-Perceptron layer and a final normalization layer. Residual connections, also called skip connection, are applied around the self-attention layer and the MLP. The order of these layers sometimes changes for different implementations.

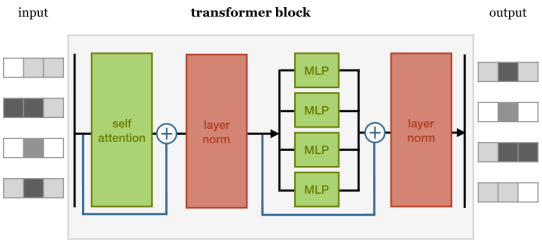


Figure 4.8: Transformer block with a self attention layer, layer normalization, a feed forward layer and another layer normalization [4].

5

Deep Anomaly Detection

With the increase of computational power and data generated, recent deep learning based research made impressive progress in the area of anomaly detection. According to Choi et al. classic anomaly detection approaches can be divided into time/frequency domain analysis, statistical models, distance-based models, auto-regressive models and clustering models [9].
Basora et al. considers distance-based and clustering models as one group and additionally introduces ensemble-based methods, domain-based methods and subspace-based methods [3].
Time series data can be analyzed in the time domain with simple limit checking with upper and lower limits for the observed values. To overcome the limitations of these fixed thresholds for dynamic systems adaptive limit checking has been developed [42]. Frequency domain analysis is popular in vibrational data such as bearing or structural health data. The Fourier theorem can be used to transfer data from the time domain into the frequency domain. Most popular algorithm in practice is the fast Fourier transform [27].
Statistical models detect anomalies based on the estimation of probability density of the data and the assumption that anomalies will fall in low-probability regions. They can be further divided into parametric models, where parameters like mean, variance etc. are calculated or non-parametric models that are based on kernel or histogram methods.
Distance based models can be separated in nearest neighbor-based methods and clustering based methods. They rely on a distance of similarity function between data instances. One of the most popular nearest neighbor techniques is the k-Nearest Neighbors method. This method determines an anomaly score by calculating the distance to the k-Nearest Neighbors. The assumption is that anomalies are outliers that have a high distance to other instances. Clustering based Methods group data into clusters close to the centroid of each cluster based on their similarities. The assumption is that anomalies are far from these pre-defined clusters or have low probability to belong to any of them.
Autoregressive models are first fitted on training data with the assumption that the output depends only on its own previous values. Then the residuals between real and predicted value are calculated to find anomalies.
Domain-based methods define a boundary or domain based on the training data to separate normal data from anomalous instances. A popular example are support vector machines.
Subspace-based models project data into lower dimensional space where anomalous data can be easier found. The most popular technique principal component analysis (PCA) [21]. PCA is a dimensionality reduction technique where the eigenvectors of the data, which are called principal components, explain the maximum variance in the data. The original data is transformed into a lower dimensional space by using the first few principal components and then transferred back to the original dimensions. A reconstruction error is calculated between the original data and the transferred data. The assumption is that for anomalies the reconstruction error is high.
Classical methods do not scale well with increasing dimensionality and massive amounts of data. Furthermore Deep Learning outperforms traditional methods as the data volume increases [6]. Many traditional algorithms suffer under the curse of dimensionality which describes that as dimensionality increases the space increases so fast that the available data becomes sparse. Recent research focuses on Deep Anomaly Detection (DAD) to overcome these challenges. It was shown that DAD surpasses traditional methods [32, 41] and was applied to a diverse field of tasks.

2*Name	2*Year	2*Network Type	2*Anomaly Criterion	F1-Score Benchmark results			
				SMD	MSL	SMAP	SWAT
Anomaly Transformer [41]	2021	Transformer	Reconstruction Error, Association Discrepancy	92,33	93,59	96,69	94,07
BeatGAN [44]	2019	Generative Adversarial Network Autoencoder	Reconstruction Error	78,1	87,53	69,61	73,92
TadGAN [12]	2020	Generative Adversarial Network	Reconstruction Error Critique Score		62,3	70,4	
MAD-GAN [25]	2019	Long Short Term Memory Generative Adversarial Network	Reconstruction Error		87,47	81,31	0,77
MTAD-GAT [43]	2020	Graph Attention Network Attention	Prediction Error Reconstruction Error		90,84	90,13	
OmniAnomaly [34]	2019	Recurrent Neural Network Variational Autoencoder	Reconstruction Error	88,57	89,89	84,34	
THOC [32]	2020	Recurrent Neural Network One-class Network	Dissimilarity		93,67	95,18	88,09
USAD [1]	2020	Autoencoder Adverse Training	Reconstruction Error	93,82	91,09	81,86	84,6
GTA [8]	2021	Transformer Graph Neural Network	Prediction Error		91,11	90,04	91
LSTM [19]	2018	Long Short Term Memory	Prediction Error		0,69	0,71	

Table 5.1: Performance comparison of recent Deep Anomaly Detection algorithms.

5.1. Anomaly criteria

Deep Anomaly Detection models typically try to minimize an objective loss function when training. The objective is dependent on the model architecture and relates to one or more decision criteria for abnormality. The models commonly output a numeric value called anomaly score that increases with the likelihood that the sample is abnormal. Data samples are then labeled as anomalous when the score exceeds a certain threshold. Thresholds used to be set empirically by domain experts but are now determined according to training results. If labeled data is available this can be done by evaluating performance on validation data or in case of non-labeled data Extreme Value Theory has been used [5]. Some models employ an adaptive threshold that adjusts the threshold to the changes in time. DAD can be categorized into three types based on the way that they calculate the anomaly score: Reconstruction error, Prediction error and Dissimilarity.
In general, autoencoders, variational autoencoders, generative adversarial networks, and transformers calculate reconstruction errors in order to obtain an anomaly score. They do this by learning to capture the low dimensional representations of the data and then map it back to the input space. Then the residuals can be calculated by comparing the reconstructed values with the original data.Reconstruction-based methods assume that anomalies lose information when they are mapped to a lower dimensional space and thereby cannot be effectively reconstructed; thus, high reconstruction errors suggest a high chance of being anomalous [12].
Prediction error methods are learning to fit a model to the given data and use this to predict future values. Like the reconstruction method, the difference between the model output and the actual values are used to identify anomalies. LSTM, CNN, Graph Neural Networks and Transformers are commonly used with a prediction error anomaly score.
Dissimilarity based models measure distance or similarity between data instances. Objects that are distant from a cluster or distribution are considered anomalies. An example for DAD using dissimilarity error is the temporal hierarchical one-class network (THOC) [32]. THOC measures the similarity between features and clusters using cosine similarity.

5.2. State of the Art Deep Anomaly Detection

This section introduces some state of the art algorithms for anomaly detection categorized in the type of architecture the algorithm uses. An overview and comparison of the algorithms can be seen in Table 5.1.

5.2.1. Metrics/Critique of the State of the Art

Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy [41]

Xu et al. propose an "Anomaly Transformer" that leverages the ability of transformers to handle really long sequences due to its global representation. The authors use the self-attention mechanism to calculate "as-

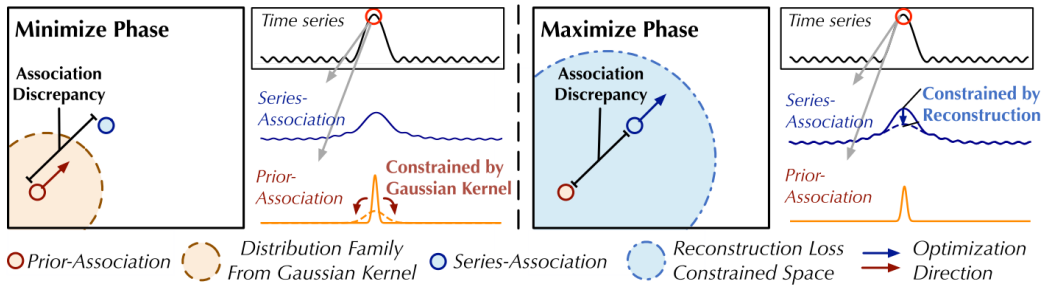


Figure 5.1: Minimax association learning as shown by Xu et al. [41].

sociation discrepancy" between "prior association" and "series association". Series association is the association of a given time-point with the whole dataset. The assumption of the data set is that anomalies are sparse and most of the data is "normal". Therefore a normal data point would have a high association with the whole data series. If the point would be abnormal, the association would be concentrated at the adjacent points since they contain more abnormal patters due to continuity. Prior association is the association with the adjacent region which is represented by a gaussian kernel that has a learnable scaling parameter. The difference between the prior association and the series association is called association discrepancy. A low discrepancy therefore corresponds to an anomaly and a high one with a normal point. The algorithm employs a minimax association learning that is visualized in Figure 5.1. In the minimize phase the prior association is changed to approximate the series association and adapt to the temporal patterns in order to decrease the association discrepancy. In the maximize phase the series association is optimized to increase the association discrepancy and therefore paying more attention to the non-adjacent horizon. The anomaly score considers a reconstruction error and the association discrepancy which can be seen in Equation 5.1.
The paper has excellent results on benchmark datasets and is one of the firsts to employ transformers. The key contribution is a new association based detection criterion which they pair with reconstruction error.

$$\text{AnomalyScore}\,(\mathcal{X}) = \text{Softmax}(-\text{AssDis}(\mathcal{P},\mathcal{S};\mathcal{X})) \odot \left[\left\| \mathcal{X}_{i,:} - \widehat{\mathcal{X}}_{i,:} \right\|_2^2 \right]_{i=1,\cdots,N} \tag{5.1}$$

GTA: Learning Graph Structures with Transformer for Multivariate Time Series Anomaly Detection in IoT

The authors propose a framework that automatically learns a graph structure, uses convolution to extract high level temporal context and models temporal dependency using a transformer based architecture. They significantly reduce the complexity of the graph learning from $\mathcal{O}(M^2)$ to $\mathcal{O}(1)$ for M candidate nodes by using Gumbel-Softmax sampling. The connection between nodes is regularized with a loss term to omit redundant connection. The novel graph convolution models the anomaly influence flowing process due to aggregation of the nodes with its neighborhood. Dilated convolutions capture the long-term temporal dependencies while the graph convolution describes topological relationships between the features. Then follows a multi-branch attention mechanism that uses global-learned attention to increase computational efficiency. The algorithm outputs a single step prediction that is labeled as anomaly if the difference from the actual value exceeds a certain threshold. The threshold is tuned with hyper parameter tuning and therefore needs labeled data.

BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series

Zhou et al. propose an algorithm that can detect anomalous "beats" when monitoring multivariate time series. BeatGan uses the loss function and the adversarial principle of the GAN method. However it also uses the structure of autoencoders to generate the adversial samples and it uses 1-d convolutions to slide along the temporal dimension of the data. Due to the adversarial regularization it obtains more robustness. The algorithm outperforms other state of the art methods on electrocardiogram data. The authors claim that BeatGan has interpretable output. They reason that it is interpretable because one can see which timesteps are labeled anomalous. However, it is found that all algorithms that were considered for this report have at least this information. Usually interpretable output has information beyond this, for example they can show which features caused the anomaly.
Furthermore the model was designed to detect anomalies on rhythmic data which might be hard to transfer

to other domains, eg. satellite data. Experiments were also performed on another dataset of motion capture data, however the data structure is similar to electrocardiogram data. Finally, the model only uses the reconstruction error as anomaly measure and uses the discriminator only for training the reconstruction.

MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks

Li et al. proposed a GAN framework for anomaly detection called MAD-GAN. It uses LSTM networks for the discriminator and generator networks. The time series data is segmented in sub-sequences with a sliding window before discriminating the data. They use a combined reconstruction and discriminator error. The vanilla GAN method only maps from latent variable into data space via the generator, but not the other way. Therefore they take an initial guess on the latent space and use gradient decent for the reconstruction loss. The recall values the algorithm achieves are good but the precision values are low. Furthermore they mention problems with computational time and accuracy seems to be decreasing with larger time window. One reason could be that for increasing time window the data space increases but the latent space is constant. Finally they also report about stability issues as the performance varies widely over different epochs.

TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks

Alexander Geiger et al. present a reconstruction and critique based GAN anomaly detection method for time series to time series mapping. It seems to be that TadGAN was largely inspired by MAD-GAN and improves on their shortcomings. They use a cycle-consistent GAN [45] because it can model the mapping of $\mathcal{X} -> \mathcal{Z}$ in both directions to encode and decode the data. Therefore TadGAN obtains a reconstruction score and a critique score which both are levered to obtaining an accurate result. They use bidirectional LSTM networks for the generators and convolutions as critics. Additionally to the in reconstruction error commonly used point wise difference, they use area difference and dynamic time warping (DTW). Both handle sequences as opposed to time points and are therefore able to identify regions of small difference for a longer period of time while dynamic time warping is additionally able to handle time shift issues. They conclude that using DTW and critique score performs best. If a known anomalous window overlaps any predicted windows it is scored as a true positive. This might further increase the problems regarding the point-adjust method that will be explained in chapter 6.

OmniAnomaly: Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network

OmniAnomaly by Su et al. is a Bayesian network using recurrent neural networks with stochastic variables. Anomaly detection objects (spacecraft, servers.. etc.) are subject to stochasticity and temporal dependence. The authors argue that therefore a stochastic modeling approach with temporal modeling would be ideal. Gated recurrent units (GRU) are used to model the long term dependencies in a sequence. GRU are, just like LSTM, improved RNN that address the vanishing gradient problem via gating mechanisms. They use a VAE that applies planar normalizing flows (planar NF), a technique that transforms the latent gaussian space into non-gaussian using invertable mapping. The anomaly score is obtained by calculating a reconstruction error. The error is calculated for each time point but previous latent variables in the timewindow are connected. More precisely they are connected by concatenation in the encoder, by Linear Gaussian State Space Model in the decoder and via the memory connections of the GRU. During training the threshold is selected following the principle of Extreme Value Theory [33]. Furthermore they can interpret anomalies by estimating the contribution of individual features.

MTAD-GAT: Multivariate Time-series Anomaly Detection via Graph Attention Network

Zhao et al. proposed an algorithm based on graph attention networks. They apply 1d-convolution to extract high level information from the raw time series. This is followed by two graph attention networks that are in parallel and each of their output is concatenated with the output of the convolution. One of these networks learns the correlation between features while the other one focuses on the temporal dependencies in the time series. The concatenated data is input to a Gated Recurrent Unit (GRU) which is followed by a forecasting based model and a reconstruction based model in parallel that are both used for combined anomaly scoring. The forecasting model is a FFNN which predicts the next timestep and the reconstruction based model is a VAE and reconstructs sequence to sequence. According to the authors it makes sense to combine them because they both have different advantages, however they do not mention which benefits they should have. Furthermore they do not mention computational time needed but considering that the architecture consists of 6 networks they might be on the larger side.

THOC: Timeseries Anomaly Detection using Temporal Hierarchical One-Class Network

Shen et al. propose a temporal one-class classification model in order to detect anomalies. By using dilated recurrent neural networks with skip connections they capture temporal dynamics in multiple scales. They obtain multiple hyperspheres with a hierarchical clustering process. The optimization objective involves multiple layers each with multiple hypersphere centers that represent normal behaviour. The loss is defined to minimize the dissimilarity between the extracted features and the centers. Furthermore a center orthogonality loss for diverse hypersphere representation in each layer and a self-supervision task loss which is based on timesequence prediction are added to the objective. While the focus in the paper lies on the one class classification it becomes apparent from the ablation study that the timesequence prediction contributes a lot to the accuracy as well.

USAD: UnSupervised Anomaly Detection on Multivariate Time Series

Audibert et al. present an algorithm that uses two adversarial autoencoder networks, inspired by GAN. They aim for achieving a high stability, robustness and training speed while maintaining or even surpassing the accuracy of other state of the art algorithms. According to the authors the adversarial training of the encoder-decoder architecture amplifies the reconstruction error while gaining stability compared to GAN methods. Furthermore they introduce a sensitivity threshold that can be changed without needing to retrain the model in order to increase or decrease the detection sensitivity. Furthermore they achieve a greatly reduced training time per epoch compared to OmniAnomaly. The timewindow of the input is small, even considering the downsampling, which might make modeling long term temporal dependencies difficult.

6

Metrics

To evaluate the performance of algorithms on datasets metrics are needed. Using the same metrics on benchmark datasets enables the research community to compare which methods are successful. Anomaly detection is usually tackled as a classification problem, labeling timesteps as positive (anomalous) or negative (normal). If it is predicted to be positive and the actual value is positive it is counted as true positive (TP). However if the ground truth is negative it counts as false positive (FP). In the same way it is done for negative predictions, a true negative (TN) responding to an actual negative label and false negative (FN) to the ground truth being positive. Summing all values for the four classes precision and recall can be calculated. Precision represents the fraction of how many of the predicted anomalies have been anomalous in reality and can be seen in Equation 6.1. Recall indicates how many positives have been predicted out of all positive predictions that could have been made and can be seen in Equation 6.2. As can be seen these two metrics are both important for assess the performance of an algorithm. However optimizing for precision minimizes false positives and maximizing recall minimizes false negatives. The F-score is a way to combine these two metrics in one objective. The F_β-score, which can be seen in Equation 6.3, has an adjustable parameter β which can be used to put emphasis on either precision or recall. A lower value puts more weight on precision while a higher value favours recall. A special case is the F_1-score which equals the importance of recall and precision. It can be seen in Equation 6.4.

$$Precision = \frac{TP}{TP + FP} \tag{6.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

$$F_\beta = (1 + \beta^2) \frac{Precision * Recall}{(\beta^2 * Precision) + Recall} \tag{6.3}$$

$$F_{\beta=1} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.4}$$

The F-score is used in most research studies of time-series anomaly detection. In the recent years high F_1-scores are reported in deep anomaly detection, giving the impression that an increase in accuracy has been made. However most studies apply a technique called Point Adjustment (PA) before scoring the performance of an algorithm. While this technique was first used by Hundman et al., it was attributed the name Point Adjust by later researchers [19]. The principle can be seen in Figure 6.1. As can be seen only one instance of the timesteps is rightfully labeled as anomalous. However, using Point Adjust, all instances in the anomalous sequence are considered as true positives when at least for one instance inside this sequence an anomaly was detected. The reason for this is that in practice operators are more interested in finding anomalous events, rather than individual data points. In spite of this, it greatly amplifies the detection of true positives and artificially inflates F1 score. Multiple authors reported to have found flaws when using point adjust.

Garg et al. found out that when using point adjustment one can not distinguish between a good detector and a random one [11]. Therefore they proposed a new metric called Composite F-Score (F_C). They reason that an ideal detection algorithm would identify at least one anomalous data point per anomaly event while

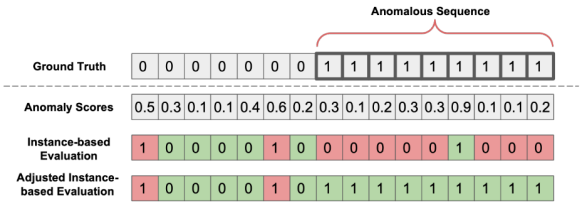


Figure 6.1: Adjusting of instance-based evaluation with Point Adjust.

having no false positives. Therefore they establish an event-wise definition for recall which can be seen in Equation 6.5. The time-wise calculation of Precision is identical to Equation 6.1.

$$Recall_{event} = \frac{TP_e}{TP_e + FN_e} \tag{6.5}$$

The F_C-score is than calculated similar to the F_1-score in Equation 6.4. The authors do not consider early detection but note that this would be desirable.

Kim et al. [22] showed that when using F_1-score with point adjust, a randomly generated anomaly score even exceeds the existing models. Furthermore they state that, in order to evaluate performance, baseline accuracy for the different datasets are needed. They propose simple models that are randomly initialized and untrained. The accuracy of existing models show no improvements against their newly proposed baseline even when PA is not used. Finally they propose an alternative metric that is called $PA\%K$. The authors argue that PA overestimates detection accuracy while using F1-score without PA underestimates accuracy due to incomplete test set labeling. The proposed alternative can mitigate the over- and underestimation effects. The metric works similar to Point Adjust but events are only "detected" when the ratio of correctly identified instances of an event exceeds the threshold K. K can be selected manually or it can be used to measure the area under curve while increasing K from 0 to 100.
Doshi et al. also criticized the current popular evaluation metrics and introduced the Sequence Precision Delay (SPD) as a new measure [10]. In order to obtain the SPD first two other metrics need to be calculated, the Sequence Alarm Precision (SAP) and the Average Detection Delay (ADD). With S being the number of anomalous events, τ_i being the starting time of an anomalous event and $T_i > \tau_i$ as the alarm time the ADD can be empirically calculated as follows as can be seen in Equation 6.6.

$$\text{ADD} = \frac{1}{S}\sum_{i=1}^{S}(T_i - \tau_i) \tag{6.6}$$

If no alarm was raised after the maximum tolerable delay δ_{max} the delay is kept constant. Therefore, the author argues, minimizing the detection delay is analogous to maximizing the true positive rate, except it assign a more specific cost to the detection delay.
The Sequence Alarm Precision is maximizing the number of detected anomalies in regards to the total amount of alarms. In this way it is similar to the popular precision metric. However the SAP focuses on sequences or also called events. With 1 being the indicator function and $\hat{S} = |T_j|$ is the total number of alarms the the way to compute the SAP can be seen in Equation 6.7.

$$SAP = \frac{1}{\hat{S}}\sum_{j=1}^{\hat{S}} 1_{\{=\}} T_j \in \cup[\tau_i, \tau_i + \delta_{\max}] \tag{6.7}$$

Finally the Sequence Precision Delay can be calculated similar to the the AUC metric by quantifying the area under the SAP vs normalised ADD. Add is normalized by dividing it through the maximum delay in order to map it to $[0,1]$. With α denoting the normalized ADD, SPD can be written as shown in Equation 6.8.
The metric works similar to the area under curve plot (AUC) and additionally to the precision it includes timeliness of the alarm.

$$SPD = \int_0^1 SAP(\alpha)\, d\alpha \tag{6.8}$$

The above mentioned metrics are performed with a fixed threshold. To provide a graphical expression of the performance of the model with varying threshold values the precision recall curve (PRC) can be used. PRC

is more informative on imbalanced datasets compared to the receiver operating characteristics plots that is sometimes used [31]. Anomaly detection is most of the time performed on imbalanced datasets as anomalies are rare events.

7

Data-set

7.1. Real-Life Satellite Telemetry Data of ESA Satellites

As use-case for machine-learning to detect anomalies on satellites the reaction wheels were identified as promising components. The considered satellites have 4 reaction wheels on board, which makes the amount of data big enough to be considered for prognostics. Furthermore they might vary in size between satellited but other than that they are similar, making it interesting to see if conclusions can be made from one satellite to another. The data is relatively easy to access and it is a mechanical component and therefore usually degrading. Most important sensor readings are the variables: temperature, current and speed. Reaction wheel failures and anomalies typically are rooted in a sudden or slow change in friction.
As Data source the missions Sentinel-1 and XMM Newton were identified. Both have several sequences with anomalous reaction wheels that were reported. These can be used to create labels for testing the algorithms. XMM Newton is in space since 1999 and Sentinel-1 since 2014.

7.2. Validation Data-sets

Since the data of the use-case is not labeled and anomalies are scarce it is necessary to validate with data-sets that are used in the research community. The Mars Science Laboratory (MSL) rover, Curiosity, and Soil Moisture Actice Passive (SMAP) satellite Data-set is a real-world spacecraft data-set released by NASA [19]. The data was expert-labeled and the input channels, which refer to one sensor or command channel, are anonymized.

7.3. Comparison

A preliminary data-analysis was performed on the ESA data and showed that the anomaly density is tiny compared to the validation dataset. This means that there are very few data segments that are anomalous compared to the huge time-window of operation, e.g. over 20 years for XMM Newton.

8

Research Approach

8.1. Knowledge Gap

Considering the literature that was reviewed in this paper there are several gaps in respect to the research topic: Machine learning anomaly detection applied to real-life data. First of all most of the reviewed papers assess the accuracy of anomaly detection methods using inaccurate evaluation metrics as described in chapter 6. The algorithms are also evaluated with benchmark data-sets that are inherently different than real-life data and are flawed as explained in chapter 7. Furthermore explainability of anomaly detection algorithms is often not considered. To understand decisions and model results it would be essential to have interpretable models. The final point is that most of the models are deterministic. For good decision-making in operational environments it would be a great improvement to quantify uncertainty.

8.2. Scope

In order to give the thesis form and a feasible time-frame choices have to be made on the aspects of the knowledge gap that are further researched. The research will investigate machine learning methods for anomaly detection using real-life satellite telemetry data from ESA. Since the results of recent studies are questionable as explained in the section above, a broader comparison study is performed to identify promising machine learning methods. The thesis will be conducted by using alternative metrics that were suggested in recent papers. Furthermore is the focus on using real-life reaction-wheel telemetry data from satellites and compare it to results of the benchmark data-sets.

8.3. Research Objective

Now that the scope is defined it is needed to formulate the research objective. The research objective puts the thesis in perspective to the current body of knowledge and entails the goals to be reached at the end of the research. The research objective is:

To find an accurate machine learning anomaly detection method for satellite systems by comparing classical machine learning methods to deep learning methods on real-life data.

8.4. Research Question

After defining the research objectives it is important to determine the research questions. Answering them will provide the key information that are necessary to achieve the research objective. In the following the main research question is stated:

Which machine learning anomaly detection algorithm is the most accurate.

This is the main research question. On the one side is is practice oriented by trying to solve the problem of ESA of using machine learning for prognostics. On the other side it also confirms with the theoretical side of the field since there is no valid comparison of algorithms on satellite systems due to the flaws in metric and benchmarks. Additionally several sub-questions have been specified to answer the detailed aspects of the main research question.

- **Which metrics should be chosen to measure performance for the comparison study?**
  This is an important sub-question that needs to be answered at the beginning of the thesis. As described earlier the current evaluation metric method commonly used is flawed. Selecting one or two metrics from the suggested ones in recent literature is necessary to compare the accuracy of the selected algorithms.

- **Is there a difference in performance between validation data-sets and the real-life ESA dataset?**
  Current benchmark datasets for anomaly detection do suffer from run-to-failure bias, mislabeled ground-truth, unrealistic anomaly density and trivial anomalies as described in 7. The current research largely concentrates on using these benchmarks instead of using real-life data it is of great value to explore how deep anomaly detection methods perform on real-world data.

- **Do deep learning algorithms perform better than classical machine learning methods?**
  As described in chapter 6 does the current impression of improvements in anomaly detection accuracy rest on the usage of flawed metrics. It is therefore of significance to re-investigate if deep learning methods increase the accuracy of anomaly detection and therefore justify the added complexity and computational requirements.

- **Do complex methods perform better than simple methods?**
  Deep anomaly detection algorithms differ largely in their architecture, anomaly criteria and their complexity. It is interesting to analyse if certain characteristics are more accurate than other if applied to real-world data.

9

Conclusion

Throughout this literature study, the current status, challenges, and opportunities of anomaly detection in satellite telemetry data were explored. It is clear that anomaly detection plays a pivotal role in ensuring the successful operation of satellite systems, yet the predominant manual and threshold-based methods currently employed face increasing challenges due to the dynamic nature of space environments and the rapidly growing volumes of telemetry data.

Despite the emergence of machine learning and deep learning techniques as promising tools for anomaly detection, the analysis of existing literature identifies several gaps in this field. Critically, the usage of inaccurate evaluation metrics and flawed benchmark datasets pose significant limitations to the effective assessment and comparison of machine learning methods. Additionally, the deterministic nature of most current models and lack of focus on model interpretability were also recognized as areas needing attention.

Given the scope of this study, the subsequent research will aim to bridge these gaps by conducting a comprehensive comparison of classical machine learning and deep learning methods for anomaly detection using real-life satellite telemetry data from ESA. The objectives are grounded in the necessity for accurate and reliable machine learning techniques for prognostics in satellite systems.

To achieve this, a series of research questions aimed at exploring key aspects of this challenge were established. This includes the identification of more suitable evaluation metrics, the exploration of the performance discrepancy between validation datasets and real-life data, and an assessment of the comparative performance between deep learning and classical machine learning methods. Furthermore, we aim to investigate whether complex methods offer any advantages over simpler ones when applied to real-world data.

In conclusion, this literature study provides the foundation for the forthcoming research. It is anticipate that the efforts will help in establishing a more effective and accurate anomaly detection system for satellite telemetry data, one that can cope with the increasing data volumes while offering more reliable prognostic capabilities. By addressing the identified gaps, it is expected to not only contribute to the practical aspects of satellite system operations but also enrich the theoretical framework in the domain of machine learning-based anomaly detection.

Bibliography

[1] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 3395–3404, 2020.

[2] Vic Barnett and Toby Lewis. Outliers in statistical data. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, 1984.

[3] Luis Basora, Xavier Olive, and Thomas Dubot. Recent advances in anomaly detection methods applied to aviation. Aerospace, 6(11):117, 2019.

[4] Peter Bloem. Transformers from scratch, 2019. URL https://peterbloem.nl/blog/transformers.

[5] Joshua B Broadwater and Rama Chellappa. Adaptive threshold estimation via extreme value theory. IEEE Transactions on signal processing, 58(2):490–500, 2009.

[6] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407, 2019.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3):1–58, 2009.

[8] Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, and Xiuzhen Cheng. Learning graph structures with transformer for multivariate time-series anomaly detection in iot. IEEE Internet of Things Journal, 9(12):9179–9189, 2021.

[9] Kukjin Choi, Jihun Yi, Changhwa Park, and Sungroh Yoon. Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. IEEE Access, 9:120043–120065, 2021.

[10] Keval Doshi, Shatha Abudalou, and Yasin Yilmaz. Tisat: time series anomaly transformer. arXiv preprint arXiv:2203.05167, 2022.

[11] Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. An evaluation of anomaly detection and diagnosis in multivariate time series. IEEE Transactions on Neural Networks and Learning Systems, 33(6):2508–2517, 2021.

[12] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Tadgan: Time series anomaly detection using generative adversarial networks. In 2020 IEEE International Conference on Big Data (Big Data), pages 33–43. IEEE, 2020.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. Communications of the ACM, 63(11): 139–144, 2020.

[15] James Douglas Hamilton. Time series analysis. Princeton university press, Princeton, 2020.

[16] Aboul Ella Hassanien, Ashraf Darwish, and Sara Abdelghafar. Machine learning in telemetry data mining of space mission: basics, challenging and future directions. Artificial Intelligence Review, 53:3201–3230, 2020.

[17] Douglas M Hawkins. Identification of outliers, volume 11. Springer, London, 1980.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[19] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pages 387–395, 2018.

[20] Sonoo Jaiswal. Long short-term memory (lstm) rnn in tensorflow, 2019. URL https://www.javatpoint.com/long-short-term-memory-rnn-in-tensorflow.

[21] Ian T Jolliffe. Principal component analysis for special types of data. Springer, 2002.

[22] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 7194–7201, 2022.

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436–444, 2015.

[25] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV, pages 703–716. Springer, 2019.

[26] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163, 2016.

[27] Henri J Nussbaumer and Henri J Nussbaumer. The fast Fourier transform. Springer, 1981.

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

[29] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6):386, 1958.

[30] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. nature, 323(6088):533–536, 1986.

[31] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PloS one, 10(3):e0118432, 2015.

[32] Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. Advances in Neural Information Processing Systems, 33:13016–13026, 2020.

[33] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly detection in streams with extreme value theory. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1067–1075, 2017.

[34] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pages 2828–2837, 2019.

[35] Zhiyi Tang, Zhicheng Chen, Yuequan Bao, and Hui Li. Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. Structural Control and Health Monitoring, 26(1):e2296, 2019.

[36] Srikanth Thudumu, Philip Branch, Jiong Jin, and Jugdutt Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. Journal of Big Data, 7:1–30, 2020.

[37] V. Tuzlukov. Signal Processing Noise. Electrical Engineering & Applied Signal Processing Series. CRC Press, Florida, 2018. ISBN 9781420041118. URL https://books.google.nl/books?id=x6hoBG_MAYIC.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[39] Renjie Wu and Eamonn Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. IEEE Transactions on Knowledge and Data Engineering, 2021.

[40] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In Proceedings of the 2018 world wide web conference, pages 187–196, 2018.

[41] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642, 2021.

[42] Takehisa Yairi, Minoru Nakatsugawa, Koichi Hori, Shinichi Nakasuka, Kazuo Machida, and Naoki Ishihama. Adaptive limit checking for spacecraft telemetry data using regression tree learning. In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583), volume 6, pages 5130–5135. IEEE, 2004.

[43] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In 2020 IEEE International Conference on Data Mining (ICDM), pages 841–850. IEEE, 2020.

[44] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. Beatgan: Anomalous rhythm detection using adversarially generated time series. In IJCAI, volume 2019, pages 4433–4439, 2019.

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232, 2017.