Master Thesis

Encoder-Decoder Approaches for Detection and Diagnosis of Anomalies in Machine Control Applications

J.M. Posch

Master Thesis DKE-21-35

Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science of Artificial Intelligence
at the Department of Data Science and Knowledge Engineering
of Maastricht University

Thesis Committee:

Dr. K. Driessens
Dr. R. Möckel

Maastricht University
Faculty of Science and Engineering
Department of Data Science and Knowledge Engineering

July 28, 2021

Abstract

Faults and defects in machine control applications and industrial systems can have far-reaching and costly consequences in terms of downtime or damage to equipment. Encoder-decoder architectures using a reconstruction objective offer a framework in which only nominal system data is necessary to train models to detect and diagnose these faults as anomalies. However, encoder-decoder architectures can prove difficult to control and interpret, posing risks for their reliability in practical application. In order to alleviate these drawbacks, a novel architecture, termed Self-Attention Autoencoder, is proposed. The proposed architecture is evaluated and compared to existing encoder-decoder architectures for its anomaly detection performance on industrial data. The performance of the Self-Attention Autoencoder is further generalized through evaluation on different types of data and anomaly classes. Additionally, an anomaly diagnosis methodology using both reconstruction error and attention matrix is proposed in order to assist engineers by identifying potential causes. The Self-Attention Autoencoder is evaluated for its anomaly diagnosis performance under the proposed methodology and shown to outperform existing architectures.

Contents

Acknowledgements

Throughout the research, experiments and writing of this master thesis I am grateful to have received much support and assistance. I would first like to thank my supervisor Jacques Verriet, who continuously put me in contact with the right people at the right time, answered many of my questions, asked important critical questions himself and steadily provided valuable feedback on the form and content of my master thesis. I would like to thank my supervisor Kurt Driessens for many greatly extended meetings in which substantial decisions were made, for his availability in times of need and for his reliable input shaping the motivation and direction of my thesis. I would further like to thank my supervisor Rico Möckel for encouraging me to extend my research into less explored directions as well as suggesting the important idea of generating synthetic datasets. Furthermore, I would like to acknowledge my colleagues at the ESI and MCS department of TNO for their input and questions furthering my work. In particular, I want to thank Jeroen Broekhuisen, who, with great patience and time-investment, solved my data-recording issues and thus prevented an unknown number of stressful days. I would further like to thank Remy Fennet of Cordis, who always promptly answered my questions on information and licenses for the Cordis Suite. Finally, I wish to thank Laura Robinson for her constructive feedback late into the night as well as her amazing support making all the difference during the most trying times of the thesis process.

This research was carried out as part of the ITEA3 18030 MACHINAIDE project under the responsibility of ESI (TNO) with Cordis Automation as carrying partner. The MACHINAIDE project is supported by the Netherlands Organisation for Applied Scientific Research TNO and Netherlands Ministry of Economic Affairs.

Chapter 1

Introduction

1.1 Problem statement and motivation

Modern industrial systems aim to be highly optimized for flexibility and efficiency. In what has commonly been referred to as Industry 4.0, machine control applications monitor physical processes and make decisions based on manufacturing requirements, sensor-data and machine state [1]. Within this context, the reliability of systems is a key factor in maintaining and improving productivity. Since complex machine control applications and their associated machine setups may involve thousands of interacting functions, it is practically impossible to manually maintain an overview of these functions running in parallel. Due to the increasing complexity of machine control applications and the connected nature of machines in production lines, single faults or defects can have far-reaching and costly consequences in terms of downtime or damage to equipment [2]. Furthermore, the types of defects and faults that might be encountered during the operation of an industrial system are usually not fully known in advance. Moreover, the time-series data resulting from such systems is high-dimensional and multi-modal in nature. Hence, it is a significant challenge to determine whether a machine control application and machine setup is performing correctly. In order to minimize the impact of faults or prevent them outright, the field of anomaly detection and diagnosis has received considerable attention from both industry and academia [3, 4, 5, 6]. In the context of this field, faults and defects are considered anomalies. Anomaly detection refers to the task of identifying observations that deviate considerably from the normal expected patterns in a dataset. Anomaly diagnosis involves determining the root cause of these deviations. This thesis focuses on the realization of a Machine Learning methodology to detect anomalous behavior of machine control applications and industrial systems as well as diagnose possible root causes of anomalies.

The approaches developed and evaluated within this thesis are set in the context of the Machinaide project. The Machinaide project addresses data-driven techniques and knowledge-based services for optimization of machines using digital twins [7]. Digital twins are virtual representations of physical systems that allow for the flow and processing of data between the system and the representation [8]. An important application of digital twins is to monitor and analyze industrial systems in order to assist in the development of models to detect anomalous behavior. The models developed in this thesis are trained and evaluated on data from one such digital twin of an industrial machine setup along with its machine control application.

This thesis aims to address the challenges of anomaly detection and diagnosis in industrial systems through three major steps: First, existing machine learning approaches for anomaly detection are adapted to and evaluated on the digital twin data (see Sections 2.2, 2.4 and 4.1). Secondly, in order to improve on the drawbacks of existing models, a new architecture for anomaly detection and diagnosis is proposed based on the domain requirements of machine control applications and industrial systems (see Section 2.3). The proposed architecture is evaluated on the digital twin data (see Section 4.2.1). Furthermore, the anomaly detection capabilities of the proposed model are generalized to different types of data and anomalies through evaluation on synthetic datasets (see Sections 4.2.2 to 4.2.6). Thirdly, methods for the diagnosis of anomalies are proposed and investigated qualitatively on the digital twin data as well as evaluated quantitatively on the synthetic data (see Sections 2.5 and 4.3).

1.2 Related work

Existing approaches to anomaly detection and diagnosis can be classified based on the learning task as well as modeling approach. As mentioned in Section 1.1, anomalies that might be encountered in an industrial system are usually not fully known in advance. In other words, the learning task investigated in this thesis can be termed semi-supervised since labeled normal (non-anomalous) data can be collected but labeled anomalous data is not available. Modeling approaches to anomaly detection in semi-supervised settings comprise *Probabilistic* models, *One-Class Classification* and *Reconstruction* models as well as purely *Distance-based* methods [9].

Distance-based methods include nearest-neighbor methods [10][11] and partitioning tree-based methods [12]. Since these methods are generally applied in the original data space, their performance suffers in higher-dimensional data [13] making them unfit for use in the problem-domain investigated in this thesis. Probabilistic models predict anomalies through estimation of the normal (non-anomalous) data probability distribution [9]. They include modeling this distribution through kernel density estimators [14], multivariate Gaussian distributions or Gaussian mixture models [15]. A test point is considered anomalous when it shows a sufficiently low log-likelihood under the modeled distribution. Similar to distance-based methods, these models perform worse in higher-dimensional environments. This is because they require an exponential increase in training data to properly model the distribution of normal (non-anomalous) data[1] as the dimension of the data space increases [9]. One-class classification methods aim to directly learn a decision boundary for normal data. The decision boundary can either be learned in the immediate input space or in a feature space. Methods in input space include Minimum Volume Ellipsoids [16] and One-Class Neighbor Machines [17]. Methods in feature space include kernel and deep-feature based One-Class Support Vector Machines [18] and Support Vector Data Description [19]. Note that since only labeled normal data is available for training, only one side of the decision boundary can be determined. When the boundary of the data is complex and non-convex the amount of training data required to fit One-Class Classification models can become very large [20]. Given that data from industrial systems is of time-series type[2] and multi-modal and high-dimensional in nature, approaches other than One-Class Classification are investigated in the context of this thesis.

Reconstruction-based methods learn to encode and subsequently decode (or reconstruct) normal dat-

[1] Within the context of this thesis, *normal* refers to nominal (i.e. non-anomalous) data. It does not refer to normally distributed data.

[2] Note that the data being of time-series type effectively acts as a multiplicative factor in terms of dimensionality.

apoints. They aim to detect anomalies by failing to accurately reconstruct them under the learned model, interpreting the reconstruction error as an anomaly score [9]. Reconstruction methods include statistical approaches such as principal component analysis (PCA) [21] and different variants of neural network encoder-decoder models such as autoencoders. It has been shown that Undercomplete Autoencoders recover the same optimal subspace as that spanned by the PCA eigenvectors if only linear neurons[3] are used [22]. However, autoencoders containing neurons with non-linear activation functions have the advantage of being able to exploit non-linear feature correlations and are being employed for semi-supervised anomaly detection in settings ranging from image data [23] to industrial control systems [24]. Furthermore, encoder-decoder models can be constructed using existing, well-established neural network architectures in order to best adapt to the problem domain.

Recurrent neural networks (RNNs) [25] are architectures specifically designed for the domain of time series data. In encoder-decoder configurations, these architectures and variations designed to overcome backpropagation limitations using memory cells such as Long-Short Term Memory (LSTM) [26] can be used for the task of anomaly detection [27][28]. However, RNN architectures process data sequentially, making them slow, which is an issue for online anomaly detection on low-performance programmable logic controllers (PLCs) often used in industrial setups. An additional challenge associated with these architectures is the issue of low interpretability [28]. Causes of anomalies are difficult to trace due to the recurrent processing of the network. Recent efforts in the field of natural language processing have lead to the development of Transformer models [29]. Transformer models are designed to process sequential data through an encoder-decoder approach and access and weigh previous states via an attention mechanism. They can efficiently process the entire time-series at once and are able to focus on subsets of the information through their attention mechanism. Recently, Meng et al. [30] have shown that the Transformer architecture can be adapted to detect anomalous behavior in a multi-variate time-series format.

In this thesis we will examine the anomaly detection performance of encoder-decoder approaches including Undercomplete Autoencoders, LSTM autoencoders and Transformers in the context of machine control applications. It will be demonstrated that there are significant drawbacks to using the standard Transformer architecture in an anomaly detection setting. Consequently, a new anomaly detection architecture, termed Self-Attention Autoencoder, is proposed in order to eliminate these drawbacks while conserving the advantages of the Transformer. An extensive evaluation of the new architecture is performed for different generalized types of data and anomalies. The associated experiments give insight into the performance of encoder-decoder approaches over training, under different numbers of anomalous dimensions and their robustness to noise. Finally, the attention mechanism together with the reconstruction error will be re-purposed for use in anomaly diagnosis and evaluated for their interpretability.

1.3 Research questions

The aim of this thesis is to develop and investigate methods for the detection and diagnosis of anomalous behavior in machine-control applications and industrial system. In this context, it will answer the following research questions:

[3] I.e. no non-linear activation functions. Instead the output of a neuron is only calculated via a weighted sum over the inputs.

RQ 1 Can system behavior of machine control applications and machine setups be learned using encoder-decoder approaches in order to enable effective detection of anomalies?

a) What is the anomaly detection performance[4] of Undercomplete Autoencoders, LSTM Autoencoders and Transformer encoder-decoders on the digital twin data?

b) How do different anomaly score evaluation methods influence anomaly detection performance?

c) Are certain types of anomalies of the digital twin data more or less difficult to detect than others?

RQ 2 Can the proposed Self-Attention Autoencoder improve on the downsides of the Transformer while conserving its advantages?

a) What is the anomaly detection performance of the Self-Attention Autoencoder on the digital twin data?

b) What is the performance[5] of the Self-Attention Autoencoder on different types of synthetic data, including general mathematical and machine-control-domain-based signals?

c) What is the performance of the Self-Attention Autoencoder on different anomaly classes, namely point, group, contextual point and contextual group anomalies?

d) How does the performance of the Self-Attention Autoencoder change when varying the number of dimensions in which an anomaly signal is present?

e) Is the reconstruction error on training data a good indicator for anomaly detection performance on test data?

f) How does the performance of the Self-Attention Autoencoder change when introducing varying amounts of noise to the test and calibration data?

RQ 3 Can reconstruction error and the attention matrix be re-purposed for use in anomaly diagnosis with the Transformer model and novel Self-Attention Autoencoder?

a) What is the anomaly diagnosis performance[6] of the Transformer model when employing the proposed methodology on generalized synthetic data?

b) What is the anomaly diagnosis performance of the Self-Attention Autoencoder model when employing the proposed methodology on generalized synthetic data?

1.4 Contributions

The key contributions of this thesis can be summarized as follows:

[4] Anomaly detection performance on the digital twin data is measured in terms of sensitivity and specificity, AUC and F1-score (see Section 3.3).

[5] Architecture performance on synthetic data is measured in terms of test-calibration anomaly score ratio (see Section 3.3).

[6] Anomaly diagnosis performance is measured in terms of anomalous observer contribution ratio and anomaly attention ratio (see Section 3.3).

- A novel architecture termed Self-Attention Autoencoder is developed to improve on the downsides of the Transformer model within the context of anomaly detection and diagnosis.
- The Self-Attention Autoencoder is evaluated and compared to Undercomplete Autoencoders, LSTM Autoencoders and Transformers for its anomaly detection performance on digital twin data. The Self-Attention Autoencoder is shown to equal performance of the best evaluated existing architecture (Transformer).
- The performance of the Self-Attention Autoencoder is generalized through evaluation on different types of data and anomaly classes as well as investigated under varying number of anomalous dimensions and for its robustness to noise.
- An anomaly diagnosis methodology using both reconstruction error and attention matrix is proposed in order to assist engineers by identifying potential anomaly causes.
- The Transformer and Self-Attention Autoencoder are evaluated under the proposed methodology for their anomaly diagnosis performance, where the Self-Attention Autoencoder is shown to significantly outperform the Transformer.

Chapter 2

Methods for Anomaly Detection and Diagnosis

This chapter will outline various methods for anomaly detection and diagnosis; however, before launching into detailed explanations of methods and architectures, an overview of the different anomaly classes as defined in Ruff et al. [9] and applied to time-series data will allow for a more comprehensive conceptualization. *Point anomalies* are observations short in duration and outside the normal value-range of a signal. *Group anomalies* are collections of consecutive observations outside the normal value-range of a signal. *Contextual point anomalies* are observations short in duration, within the value-range of a signal but deviating from the regular pattern of the signal. *Contextual group anomalies* are collections of consecutive observations, within the value-range of a signal but deviating from the regular pattern of the signal. Examples of all described anomaly classes can be found in Appendix A.2.3 in Figures A.11 to A.14.

2.1 Anomaly detection via reconstruction

Reconstruction-based methods for anomaly detection involve encoding and subsequently decoding input data. During the training phase, the model is given normal (non-anomalous) data and is tasked with encoding it into a compressed or abstracted representation. The model then uses the representation in order to reconstruct the input. A reconstruction model φ_θ is trained by minimizing the objective function L_r with regards to parameters θ of the model. As shown in Equation 2.1, L_r measures the deviation[1] between the input x and reconstructed data $\varphi_\theta(x)$. This deviation is termed reconstruction error and is measured across all N samples of the training dataset.

$$\min_\theta \; L_r(x) = \min_\theta \sum_{n=1}^{N} ||x_n - \varphi_\theta(x_n)||^2 \tag{2.1}$$

The intention is for the model φ_θ to learn the structure and properties of the normal input data. Figure 2.1 shows a visualization of the testing phase of a reconstruction model. During the testing phase

[1] This thesis uses mean-squared error for measuring the deviation between the input and reconstructed data.

the model is given both normal and anomalous data. Since the model has been trained on only normal data, it is expected to reconstruct normal inputs well and anomalous inputs badly. Thus, the degree of anomaly - also termed anomaly score - of a given datapoint x is defined as its reconstruction error $L_r(x)$. Note that a single given anomaly score does not correspond to a classification of its associated datapoint. In order to classify a datapoint, its anomaly score needs to be evaluated based on a condition or threshold. Datapoints with anomaly scores meeting the condition or exceeding the threshold are then classified as anomalous. The different anomaly score threshold types evaluated in this thesis are presented in Section 2.4.

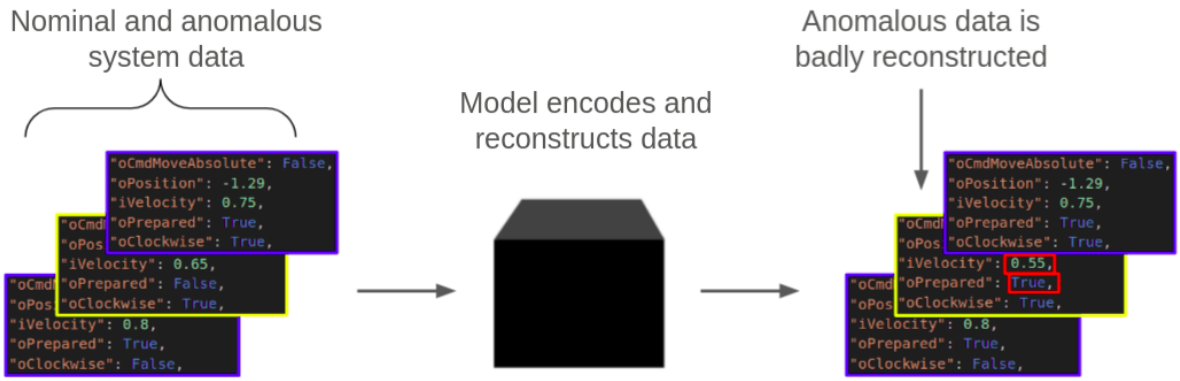


Figure 2.1: Visualization of the testing phase of a reconstruction model for anomaly detection. Normal (blue-outlined) and anomalous (yellow-outlined) system data is passed into the model, which encodes and reconstructs the input data. The normal data is reconstructed well. The anomalous data shows reconstruction errors (highlighted in red).

Note that, without any constraint on the model, the reconstruction objective can be trivially solved by using the identity mapping for φ_θ. This is undesirable as reconstruction errors for all data would equal zero and thus could not be used to distinguish normal and anomalous datapoints. In order to avoid this problem, the models in this thesis rely on a bottleneck, where input data is compressed into a lower-dimensional representation. This forces the models to develop an abstract representation of the input. Given the models are trained on normal data, they are expected to develop good compressed representations for reconstructing normal data, resulting in low anomaly scores, but have difficulties with representing (and thus successfully encoding and decoding) anomalous data, resulting in high anomaly scores. The actual reconstruction models used to encode and decode data, replacing the black box in Figure 2.1, will be presented in Section 2.2.

2.2 Encoder-decoder models

In order to perform reconstruction, the models employed in this thesis incorporate both an encoder and decoder component, which are connected in sequence. The encoder transforms the input into an abstract representation. The decoder then maps this representation to an output, which is compared with the input. Encoder and decoder components can represented by single neural network layers or by more complex networks.

2.2.1 Undercomplete Autoencoders

A fully connected Autoencoder whose encoded dimension is less than the input dimension[2] is called undercomplete [32]. Undercomplete Autoencoders have been investigated in the context of anomaly diagnosis for decades [33] and still remain in use today [34]. Since they consist of fully connected neural network layers and lack architectural adaptations for time-series data, they are not expected to show best performance in the industrial setting. They are incorporated in this thesis as a baseline architecture.

2.2.2 LSTM Autoencoders

In contrast to Undercomplete Autoencoders, LSTM-based [26] Autoencoders are specifically designed for time-series data. This thesis uses an architecture originally proposed by Malhotra et al. [27] for multivariate time-series anomaly detection. Note that, since LSTM Autoencoders process data in a recurrent fashion, they become computationally expensive for large time-windows. Additionally, their recurrent processing entails that causes of anomalies are difficult to trace back through the network.

2.2.3 Transformers

Contrary to LSTMs, Transformer [29] models are able to process input sequences in parallel. This allows for a large decrease in computation time compared to LSTMs, especially for long input sequences. Transformers were designed and traditionally used for natural language processing, for example for translation. They make use of an attention mechanism, which allows them to focus to varying degrees on different elements of the input sequence as they process a given element. Figure 2.2 shows an illustrative example of an attention matrix for a sequence containing four elements. The i-th row of the attention matrix corresponds to the attention weights placed on every input element in order to process the i-th input element. E.g. in order to process the first element (row 0), the model places a large amount of focus on the 0th input element, a small amount of focus on the 1st input element and none on the 2nd and 3rd. Attention matrices can be helpful for understanding what the model identifies as relevant during its processing of the input.

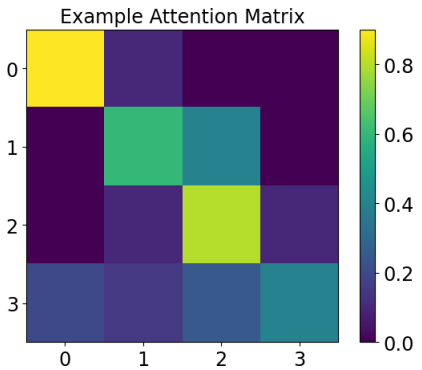


Figure 2.2: Example of an attention matrix for a sequence containing four elements.

[2]This, in essence, describes the presence of a bottleneck.

For anomaly detection in this thesis, Transformers will be used in their original form, as proposed by Vaswani et al. [29], with the following slight modifications: The Transformer model is fed with industrial system data as input to *both* the encoder and decoder[3]. Furthermore the input embeddings are removed since the industrial system data already consists of numerical values. Finally, the Softmax layer before the output is dropped as we intend to produce real values instead of probabilities. A visualization of these slight modifications can be seen in Figure 2.3.

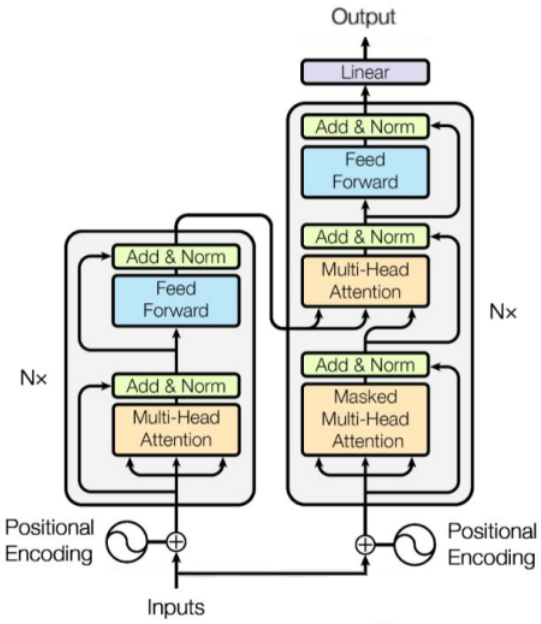


Figure 2.3: Visualization of the Transformer architecture as used in this thesis. Note that Multi-Head Attention layers contain multiple, separately trained, attention mechanisms. Adapted from Vaswani et al. [29].

Instead of comparing the processed output of the Transformer with an external target (as is done in translation), it is compared with the original input and the model is trained using the error signal resulting from that comparison. This, in effect, corresponds to training the model on a reconstruction objective. Using the Transformer model in the context of anomaly detection allows for faster computation through parallel processing. Furthermore, the attention mechanism has the potential to make the model's processing more interpretable and thus enable the development of methodologies to assist anomaly diagnosis.

However, as Section 2.5 will discuss in detail, the attention matrices of Transformer models will turn out not to be easily interpretable in the context of anomaly detection. Note that there are three separate layers of attention in the Transformer model. Furthermore, recall that the bottleneck of a reconstruction model is essential as a constraint to ensure that the model learns the structure of the data rather than defaults to an identity mapping. Unfortunately, ensuring the consistent presence of a bottleneck in Transformers is not as straightforward as desired. In Transformer models it is difficult to control the flow of information due to residual connections that act as information pipes past bottlenecks and

[3]This is in contrast to a Transformer in normal sequence-to-sequence configuration, in which the encoder is fed with the input and the decoder with an external target. For more information refer to Vaswani et al. [29].

dropout layers which act as potential information drains. Thus, any input information can theoretically flow through the entire model (through residual connections) or disappear at various points (through dropout layers). Issues with this complex flow of information became apparent when equipping a Transformer model with a bottleneck theoretically large enough to create an identity mapping. When the model was trained on the aforementioned reconstruction objective, it was observed that its parameters did *not* collapse into the identity mapping, as would be expected. The observation that the bottleneck cannot reliably be controlled in Transformer models, in combination with unintelligible attention matrices, provides motivation for the development of a new reconstruction architecture.

2.3 Self-Attention Autoencoder

With the previously noted issues of the Transformer in mind, a new architecture is proposed. The objective in designing this architecture is to obtain a controllable bottleneck as well as interpretable attention matrices while preserving the advantage of the Transformer, namely fast parallel processing of sequences. In a first step, all information pipes and drains, i.e. all residual connections and dropout layers, are removed from the Transformer. Next, with the intention of simplifying the model to its essentials, the decoder component of the model is completely removed. This means that input is now only fed into a single component and only a singular layer of attention remains. Furthermore, instead of Multi-Head Attention, Scaled-Dot-Product Attention[4] was used, which only produces a single attention matrix. The bottleneck and decoder of the new model are realized by converting the feedforward layer of the Transformer's encoder into an Undercomplete Autoencoder. A visualization of the proposed architecture, termed Self-Attention Autoencoder, is shown in Figure 2.4. Further note that while models with similar names were proposed in [35] and [36] these represent substantially different architectures to the model proposed in this thesis.

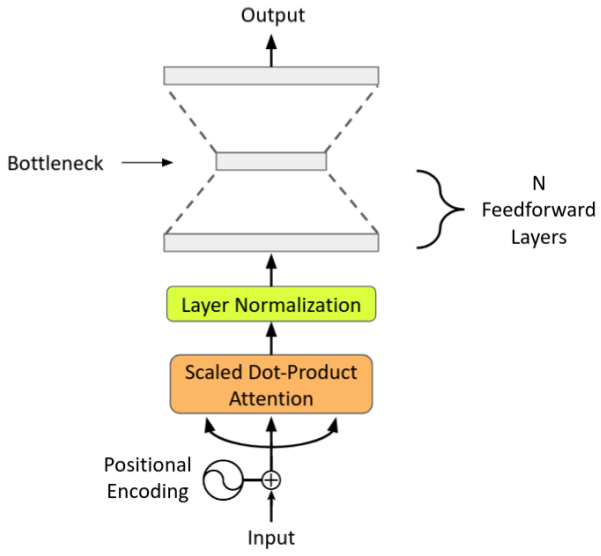


Figure 2.4: Visualization of the Self-Attention Autoencoder architecture proposed in this thesis.

[4]Both Multi-Head Attention and Scaled-Dot-Product Attention make use of the same underlying matrix operations. Multi-Head Attention refers to multiple, separately trained attention mechanisms, whose outputs are subsequently concatenated. Scaled-Dot-Product Attention refers to a single attention mechanism. For more information refer to Vaswani et al. [29].

2.3.1 Mathematical model

In order to give a concrete overview of the proposed model, this section presents a mathematical description of the architecture. The positional encoding shown in Equation 2.2 aims to ensure that information about the position of the input sequence elements is present, despite the model containing no recurrence or convolution [29]. With d_{input} corresponding to the dimensionality of the input data, a positional embedding $PE_{(t,2i)}$ for timestep t of signal i of a given input time-window is obtained.[5]

$$\begin{aligned} PE_{(t,2i)} &= \sin\left(\frac{t}{10000^{2i/d_{\text{input}}}}\right) \\ PE_{(t,2i+1)} &= \cos\left(\frac{t}{10000^{2i/d_{\text{input}}}}\right) \end{aligned} \tag{2.2}$$

After an input matrix[6] is passed into the model and summed with its positional embeddings, it is passed into the Scaled-Dot-Product Attention layer shown in Equation 2.3, where W_Q, W_K and W_V are weight matrices for Query, Key and Value[7].

$$Attention(X) = softmax(\frac{(XW_Q)(XW_K)^T}{\sqrt{d_{\text{input}}}}XW_V) \tag{2.3}$$

The output of the Scaled-Dot-Product Attention layer is passed into layer normalization [37], where $E[\,]$ is the expected value, $Var[\,]$ is the variance, γ and β are learnable parameters and ϵ is a value added to the denominator for numerical stability:

$$LayerNorm(X) = \frac{x - E[X]}{\sqrt{Var[X] + \epsilon}} \cdot \gamma + \beta \tag{2.4}$$

Finally, the output of layer normalization is passed into a fully connected feed forward neural network in the shape of an Undercomplete Autoencoder.

2.4 Anomaly score evaluation methods

The anomaly score (or reconstruction error) resulting from the application of the models described in Section 2.2 and 2.3 serves as a quantitative measure of the degree of anomaly. However, as described in Section 2.1, a single given anomaly score does not correspond to a classification of the corresponding datapoint. In order to obtain a classification, the anomaly score must be contextualized by comparing it to an indicative distribution of anomaly scores [9]. Within the context of this thesis, this distribution is a calibration dataset made up of normal datapoints that the models were never exposed to. Based on the anomaly scores computed on the calibration dataset, a threshold is established. Any testset datapoint whose anomaly score is above the threshold is classified as anomalous. The

[5] As in Vaswani et al. [29] we choose 10000 in the denominator. This allows for a maximum of $2\pi \cdot 10000$ unique positional encodings.

[6] Note that the input is a matrix since it consists of t timesteps of d_{input} signals.

[7] For further information refer to [29].

three types of threshold anomaly classifiers investigated in this thesis are presented in the following sections.

2.4.1 Mean threshold

The mean threshold anomaly classifier c_μ, shown in Equation 2.5, is informed solely by the first statistical moment (or mean) of the calibration anomaly score distribution. A given test anomaly score a_{test} is considered anomalous if it is greater than the mean of the calibration anomaly score distribution μ_{cal} multiplied by the treshold factor α.[8]

$$c_\mu(a_{\text{test}}) = \begin{cases} 1 & a_{\text{test}} > \alpha \cdot \mu_{\text{cal}} \\ 0 & a_{\text{test}} \leq \alpha \cdot \mu_{\text{cal}} \end{cases} \tag{2.5}$$

It should be noted that the threshold factor α is a parameter that can be adapted to obtain different classification results. Section 3.4 will describe the range of values of α for the experiments of this thesis.

2.4.2 Standard deviation threshold

The second type of threshold anomaly classifier c_{sd}, shown in Equation 2.6, is informed both by the first and second statistical moments (mean and variance) of the calibration anomaly score distribution. A given test anomaly score a_{test} is considered anomalous if it deviates[9] by more than α standard deviations σ_{cal} from the mean of the calibration anomaly score distribution μ_{cal}.

$$c_{\text{sd}}(a_{\text{test}}) = \begin{cases} 1 & a_{\text{test}} > \mu_{\text{cal}} + \alpha \cdot \sigma_{\text{cal}} \\ 0 & a_{\text{test}} \leq \mu_{\text{cal}} + \alpha \cdot \sigma_{\text{cal}} \end{cases} \tag{2.6}$$

Given a sufficiently fine range of different threshold values α this threshold type is comparable to c_μ. However, c_{sd} could perform better than c_μ if the standard deviation of the calibration anomaly scores is especially large or small compared to their mean. In contrast to c_μ, c_{sd} would then be varied over a larger or conversely finer range of threshold values.

2.4.3 Series threshold - Western Electric rules

Instead of classifying single anomaly scores, the final anomaly classifier c_{WE}, shown in Table 2.1, establishes a set of decision rules that take the immediate context or local neighborhood of a series of consecutive anomaly scores into account [38].

The Western Electric rules c_{WE} are motivated by the consideration that consecutive series of datapoints jointly deviating from the mean are statistically unlikely under the assumption of a normal data distribution and thus likely to be anomalous. Note that, unlike the previous anomaly classifiers, c_{WE} does not have a variable threshold factor. For the experiments of this thesis the classification performance of every combination of the set of Western Electric rules will be evaluated.

[8] Note that the class label for an anomalous datapoint is 1 and the class label for a normal datapoint is 0.

[9] Only positive deviations from the mean are considered.

Rule	Definition	
Rule 1	A single data point for which holds:	$a_{\text{test}} > \mu_{\text{cal}} + 3 \cdot \sigma_{\text{cal}}$
Rule 2	Two out of three consecutive points for which hold:	$a_{\text{test}} > \mu_{\text{cal}} + 2 \cdot \sigma_{\text{cal}}$
Rule 3	Four out of five consecutive points for which hold:	$a_{\text{test}} > \mu_{\text{cal}} + 1 \cdot \sigma_{\text{cal}}$
Rule 4	Nine consecutive points for which hold:	$a_{\text{test}} > \mu_{\text{cal}}$

Table 2.1: The Western Electric rules c_{WE} for detecting anomalous datapoints. Note that standard Western Electric rules include both positive and negative deviations from the mean; however, in the context of anomaly detection, only high anomaly scores (positive deviations) are considered.

2.5 Anomaly diagnosis - methods for interpretability

With the methods described in the previous section, a datapoint can be classified as normal or anomalous depending on its anomaly score. Beyond binary classification though, further methods to interpret a model's output are desirable since, within the context of industrial systems, interpretability amounts to saved costs. If the cause of an anomaly can be quickly found, machines can be restarted or repaired earlier and consequently downtime is reduced. While human domain expertise remains crucial for resolving anomalies, the methods presented in this section intend to assist engineers by giving them the tools needed to identify and point in the direction of potential causes.

2.5.1 Reconstruction comparisons

The main method for interpretability on which this thesis expands is in principle a straightforward approach to anomaly diagnosis using reconstruction models, as it consists of comparing the input signal to the reconstructed signal obtained from the model. Figure 2.5 shows the input signal from a time-window classified as normal, as well as the reconstructed signal obtained from passing that input signal through the Self-Attention Autoencoder model. Note that while the reconstruction does show slight deviations, it mostly matches the input signal.

In contrast, Figure 2.6 shows the input signal from a time-window classified as anomalous and the reconstructed signal obtained from passing that input signal through the Self-Attention Autoencoder model. Note that the reconstructed signal deviates significantly from the input signal between timesteps 68 and 85. This indicates that between these timesteps there was a deviation from what the model, as trained on normal data, expected. Since the model is trained to reconstruct normal data well and is expected to reconstruct anomalous data badly, a deviation from the model's expectation indicates a probable cause of the anomaly, providing engineers with a specific direction for further investigation.

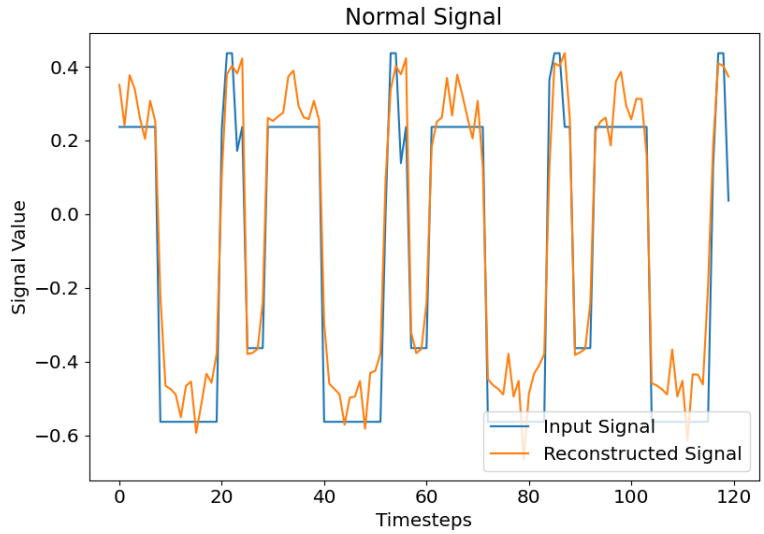


Figure 2.5: Example normal input signal (blue) and corresponding reconstructed signal (orange) output from an encoder-decoder model. The reconstructed signal shows slight errors but mostly matches the input signal.

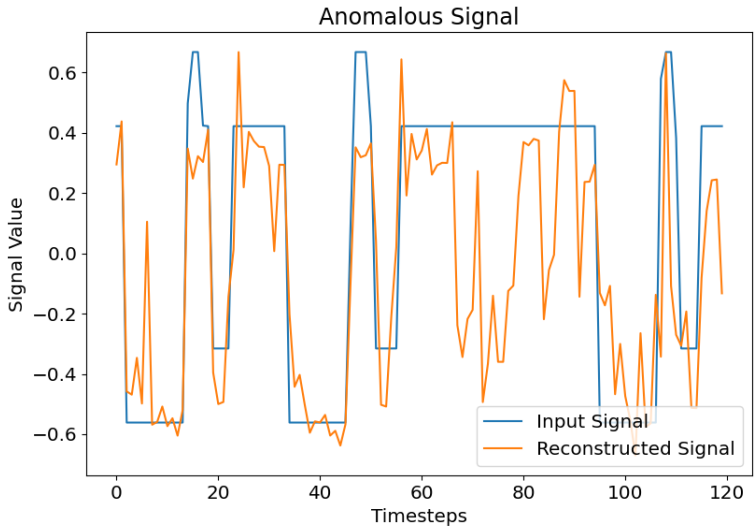


Figure 2.6: Example anomalous input signal (blue) and corresponding reconstructed signal (orange) output from an encoder-decoder model. Note the anomalous input signal from timestep 57 to 95. Further note the large deviation of the reconstructed signal from the input signal.

2.5.2 Observer contribution to reconstruction error

Reconstruction comparisons provide explicit insights into potential anomaly causes. However, the data resulting from real-life industrial systems consists of hundreds to thousands of signals, each of

which could be studied. Thus, the question becomes: which signal(s) should engineers investigate? To answer this, the second tool for anomaly diagnosis is introduced. Recall that the total anomaly score for a given time-window is the sum of the reconstruction errors of all the signals. Figure 2.7 displays the contribution of each observer (or signal) to the total anomaly score. One can determine which signals contribute most to the anomaly score - i.e. which signals the model reconstructs worst. These are the signals on which engineers should focus.

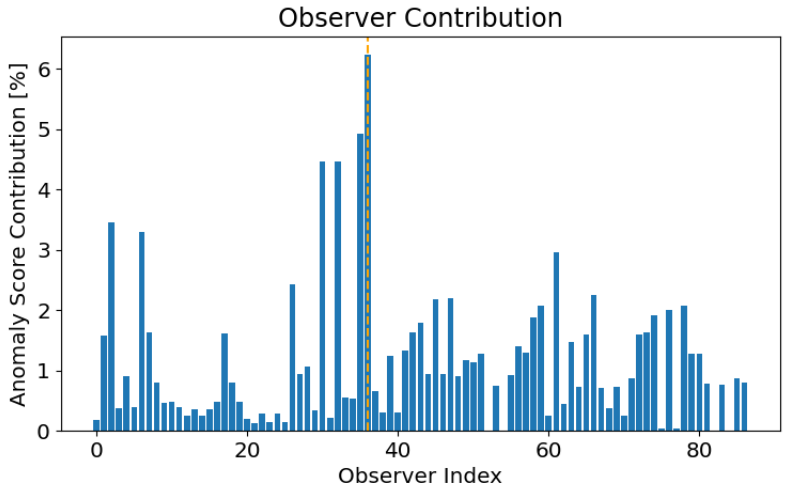


Figure 2.7: Relative contribution of each observer to the total anomaly score for a single time-window. Observer 36 (indicated with an orange dotted line) contributed the most to the anomaly score for this time-window.

2.5.3 Attention matrix interpretation

Signal contributions to the anomaly score provide a method for determining which signals to investigate. However, locating specific timesteps in which deviations occur within an identified signal provides a further challenge. Note that in Figure 2.6 it is relatively easy to do so but this is not always the case. Consider Figure 2.8, which displays the input and reconstruction signal from a time-window classified as anomalous. Which of the two highlighted areas is the probable anomaly location? The v-shape at the beginning of the time-window or the extended constant signal at the end of the time-window?

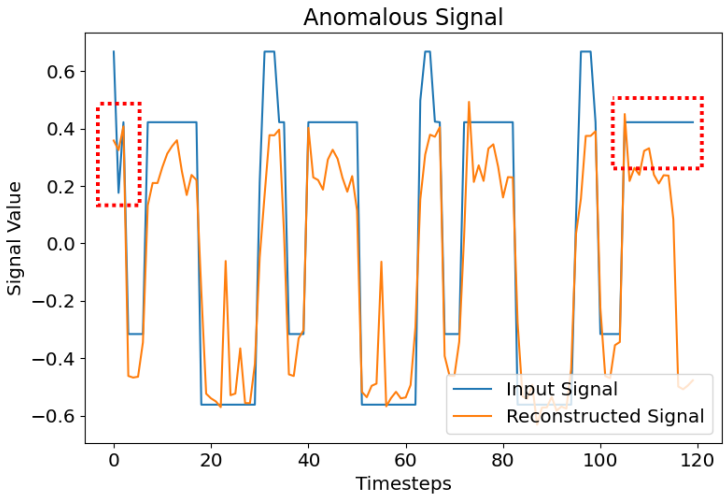


Figure 2.8: Example anomalous input signal (blue) and corresponding reconstructed signal (orange) output from an encoder-decoder model. The two areas highlighted in red correspond to probable anomaly locations.

In order to locate the timesteps of the probable anomaly cause, we propose to use another method to understand what affects the model's reconstruction. What the model pays attention to as it reconstructs a given signal badly is a good candidate for the aforementioned timestep. Therefore, we suggest to study the attention matrix of the model. Figure 2.9 displays three attention matrices from a *Transformer* model evaluated on three time-windows classified as *normal*. As can be seen, the pattern of attention is comparable across matrices, which is to be expected for normal datapoints. However, note that every row within each individual matrix is almost identical to every other row in the same matrix. This implies that the model does not change its focus of attention across timesteps; thus suggesting that the model makes little use of the attention mechanism.

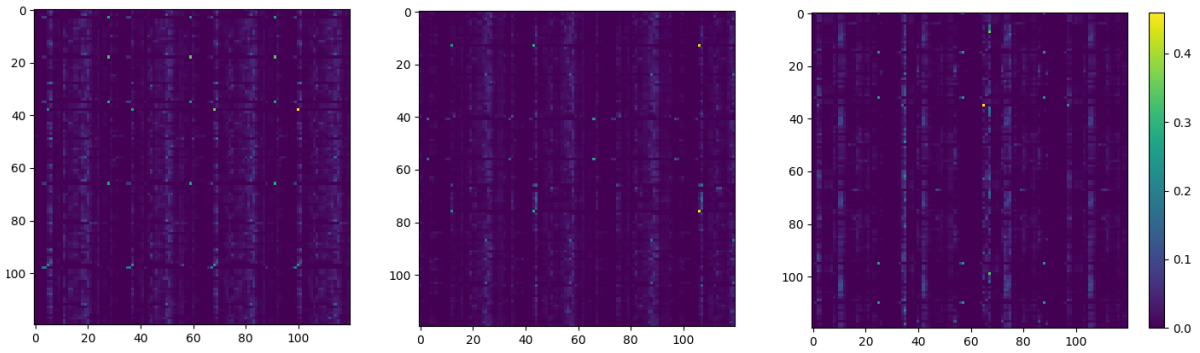


Figure 2.9: Example attention matrices from a Transformer model evaluated separately on three normal time-windows. The pattern of attention does not change significantly between matrices. Additionally, rows within a single matrix are almost identical.

Refer now to Figure 2.10 which displays three attention matrices from a *Transformer* model evaluated on three time-windows classified as *anomalous*. Barely any identifiable difference can be seen if one compares these to the matrices presented in Figure 2.9. Ideally one would be able to easily distinguish between the matrices resulting from normal and anomalous time-windows, and the latter would display structural patterns for locating specific timesteps. This is not the case for Figures 2.9 and 2.10.

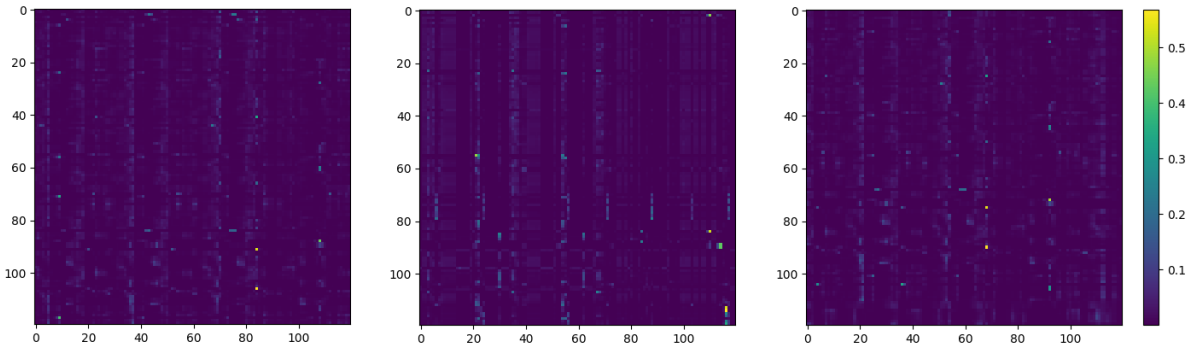


Figure 2.10: Example attention matrices from a Transformer model evaluated separately on three anomalous time-windows. Very little difference can be observed between these attention matrices and the ones resulting from normal time-windows presented in Figure 2.9.

Now consider Figures 2.11 and 2.12, which display the attention matrices from a *Self-Attention Autoencoder* evaluated on three time-windows previously classified as *normal* and *anomalous*, respectively. Figure 2.11's attention matrices show very similar patterns between each other, as do the earlier matrices from the *Transformer model*. However, while the matrices from the *Transformer* model show an attention pattern that remains the same regardless of the timestep, this is not the case for the *Self-Attention Autoencoder* matrices. The latter's matrices (Figure 2.11) show attention is spread more or less evenly across all timepoints and there is a discernible diagonal pattern to the attention of the model. This means that the model uses one pattern of attention but shifts it linearly as different timepoints are reconstructed.

The attention matrices in Figure 2.12 result from a *Self-Attention Autoencoder* evaluated on three time-windows classified as *anomalous*. Note that there is a significant difference between these and the matrices resulting from normal time-windows (Figure 2.11). This is desirable as it implies that the model treats normal and anomalous time-windows differently. The regular diagonal pattern from Figure 2.11 has almost completely disappeared in Figure 2.12. Notice especially column 64 of the left-most matrix containing multiple large attention weights. The model pays a lot of attention to this timestep as it reconstructs others. Further observe that the center attention matrix again contains a column (44) with multiple large attention weights. Finally, the right-most attention matrix also contains a column (24) with multiple large attention weights. In short, the focus of attention is located at column 64, 44 and 24 respectively. It should be noted at this point that the attention matrices displayed in Figure 2.12 result from time-windows exactly 20 timesteps apart. This implies that the model pays particular attention to one specific timestep of the signal as it reconstructs three different anomalous time-windows. As will be expanded on next, this timestep points to the anomaly.

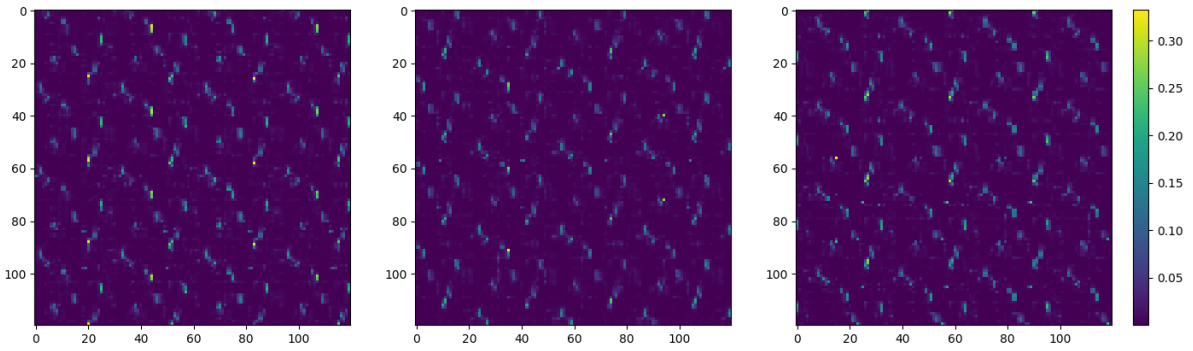


Figure 2.11: Example attention matrices from a Self-Attention Autoencoder model evaluated separately on three normal time-windows. A regular, diagonal pattern of attention can be observed.

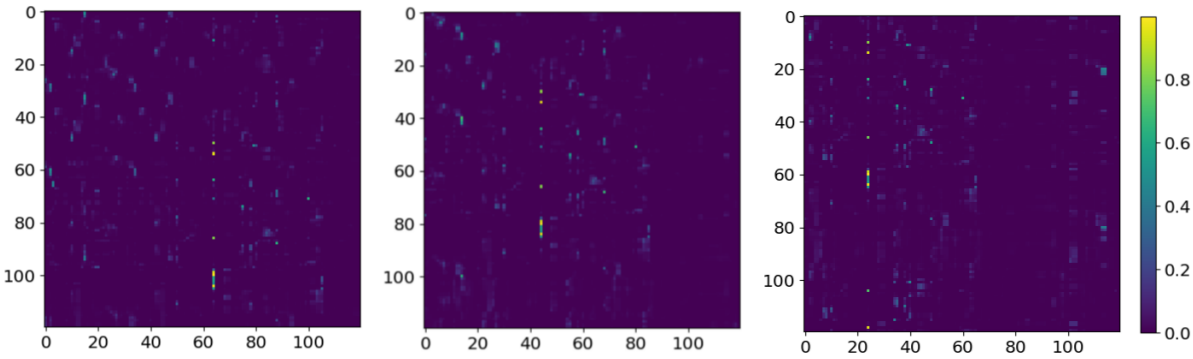


Figure 2.12: Example attention matrices from a Self-Attention Autoencoder model evaluated separately on three anomalous time-windows. The regular attention pattern from Figure 2.11 has almost completely disappeared. A single column in each attention matrix shows very large attention weights.

To summarize, the normal attention matrices resulting from the *Self-Attention Autoencoder* show a regular diagonal attention pattern. The anomalous attention matrices from this model deviate significantly from the normal pattern. Furthermore they show a clear structural pattern identifying a specific part of the signal. How does this help anomaly diagnosis? For this we will identify to which time-window the leftmost attention matrix of Figure 2.12 corresponds. It is the time-window presented in Figure 2.6. The timestep that is being paid most attention to by the model (as identified by the attention matrix) is timestep 64. This is the moment at which the anomalous continuous signal occurs, implying that the *Self-Attention Autoencoder* model pays attention to the location of the anomaly. Furthermore, as we shift the time-window forwards by 20 and 40 timesteps the corresponding attention matrices (Figure 2.12 center and right) also shift their maximum attention by 20 and 40 timesteps, implying that the model continues to focus attention on the location of the anomaly even across varying time-windows.

We thus propose the following anomaly diagnosis methodology: When an anomaly is detected, the signal with maximal anomaly score contribution as well as the attention matrix column (timestep) with

maximal attention are determined. These identify the exact signal and timestep where the model's reconstruction of the input signal was thrown off, giving a clear indicator for engineers to directly focus their cause identification investigation on.[10]

[10]Note that quantitative metrics to measure the performance of this methodology will be presented in Section 3.3.

Chapter 3

Data and Experimental Methodology

3.1 Merry-Go-Round system and dataset

The Merry-Go-Round (MGR) system, from which the data for this thesis was collected, is a simulation of the physical behavior of an industrial machine setup. Figure 3.1 depicts the graphical visualization of the MGR simulation. It contains trays (in blue) that are moved around the system via conveyor belts (in red), lifts (in green) and gantries (in dark grey).

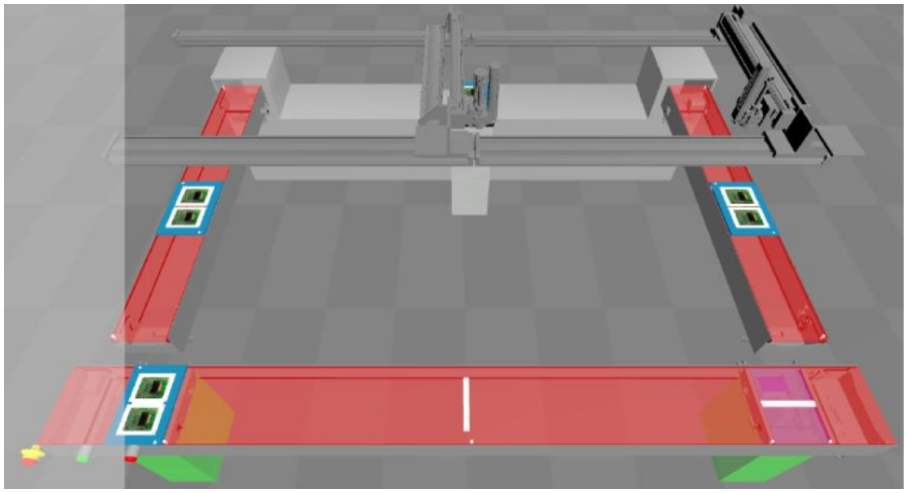


Figure 3.1: Graphical visualization of the MGR simulation.

The components that make up the machine setup contain sensors that record time-series data such as motor positions and velocities. The simulation is accompanied by a machine control application which monitors and records the state-machines of each component and sub-component, recording time-series data in the form of state-transitions. The data resulting from this system is thus a mix of time-series numerical and categorical variables, also termed *observers* within the context of this thesis.

3.1.1 Anomalies in the Merry-Go-Round system

When running the Merry-Go-Round system for an extended period of time, an anomaly can occur. Sometimes a tray is placed too far on the left side of the left conveyor and does not continue to move around the system but remains stuck there until a second tray is placed on top of it. The two trays subsequently travel around the system together. Once this happens, it is likely that one or both of the trays will fall off at one of the lifts. Within this thesis, any point at which a tray is stuck without moving, is stacked on top of another tray or has fallen off is considered anomalous.

Since the MGR system is accompanied by a machine control application it is possible to adapt system settings in order to induce system runs that deviate from the norm. Within this thesis, two types of adapted runs were performed. For the first type the conveyor belt velocities were increased by 0.1m/s, corresponding to a 20% increase compared to the norm. For the second type the gantry velocities were increased by 0.25m/s, corresponding to a 33% increase compared to the norm. Any datapoint that was recorded with either of these modified settings is considered anomalous. Note that all anomalies investigated for the MGR system are of the contextual group anomaly type. This is because observer values during anomalies stay within nominal value ranges and can only be identified as anomalies when taking into consideration the context of the time-series distribution.[1]

3.1.2 Data recording

Within the context of this thesis, 30 MGR-system runs were recorded. 10 MGR-runs were recorded with standard settings and 10 runs each were recorded with either of the two modified settings. The standard MGR-runs were manually monitored via the graphical visualization and datapoints labeled nominal or anomalous as described in Section 3.1.1. The datapoints of all runs with modified settings were labeled anomalous. A total of 6 315 nominal and 24 300 anomalous datapoints were recorded. This imbalance is due to the setting-based anomalies.

3.1.3 Data handling and formal data analysis

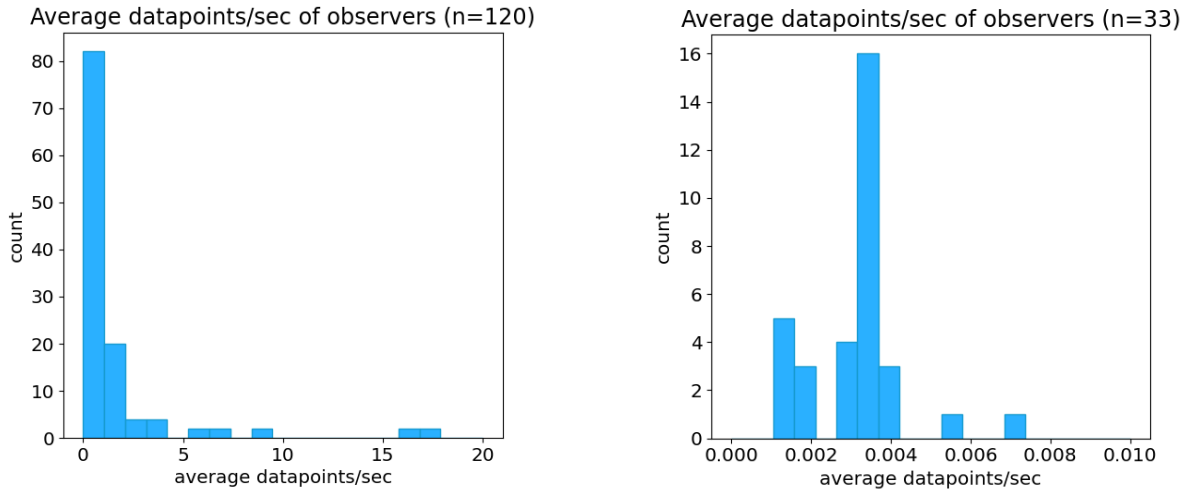
In order to obtain a concise overview of the data as well as make well-founded decisions on data handling, this section outlines the results of a formal data analysis on the MGR dataset. Starting with an initial total count of 237 observers, data cleaning was performed - all observers that are constant throughout all recorded system runs were excluded from the data as they do not add any information. Furthermore, it was noted that certain observers were not recorded on every system run. These were also excluded from the data to ensure a consistent set of observers across all runs. Finally, based on the advice of MGR system engineers, a set of observers related to system execution duration were excluded from the data. The values of these observers are strongly dependent on CPU usage when running the simulation and are therefore dependent on outside factors rather than system behavior. After the aforementioned data cleaning steps, 120 observers remained, including 54 numerical and 66 categorical observers. Table 3.1 shows an overview of the number of observers before and after data cleaning.

[1] Section 3.2 will present additional datasets that were created to evaluate the developed methods for other general types of anomalies, namely point, contextual point and group anomalies.

	Total Observers	Numerical Observers	Categorical Observers
Before data cleaning	237	145	92
After data cleaning	120	54	66

Table 3.1: Number of total, numerical and categorical observers before and after data cleaning.

For the first part of the data analysis, the average number of datapoints recorded every second were computed for every observer across all system runs. The corresponding histogram is shown in Figure 3.2a. It should be noted that most observers record a low number of datapoints per second. Further investigating the 81 observers with less than an average of 1 datapoint recorded per second revealed that there are a substantial number (33) of observers that record on average fewer than 0.01 datapoints per second. These observers, shown in a histogram in Figure 3.2b, are exclusively categorical. They describe states that power on the system and are only active at the very beginning of every system run. Including these observers would unnecessarily obscure reconstruction without adding relevant anomaly information. Therefore these observers were excluded from the data, leading to a total of 87 observers (54 numerical and 33 categorical) in the dataset.



(a) (b)

Figure 3.2: (a) Histogram for the average number of datapoints recorded per second for every observer. (b) Histogram for the average number of datapoints recorded per second for observers recording fewer than 0.01 datapoints per second.

The remaining categorical observers were analyzed according to the number of different states they show throughout all system runs. Figure 3.3 depicts the histogram of state numbers of categorical observers. It should be noted that the vast majority of categorical observers are either binary (12 observers) or show three or four states (17 observers). An example of a categorical observer with four different states can be seen in Appendix A.1 in Figure A.1.

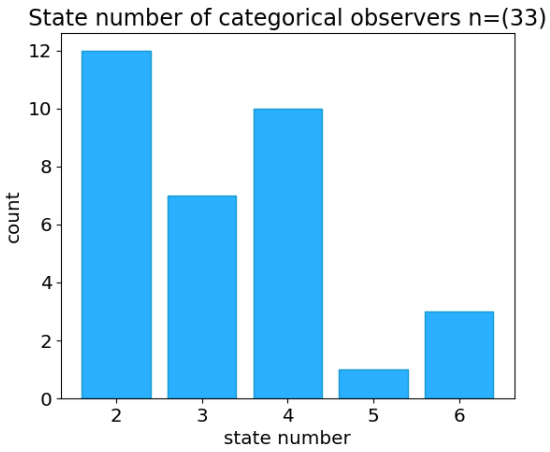
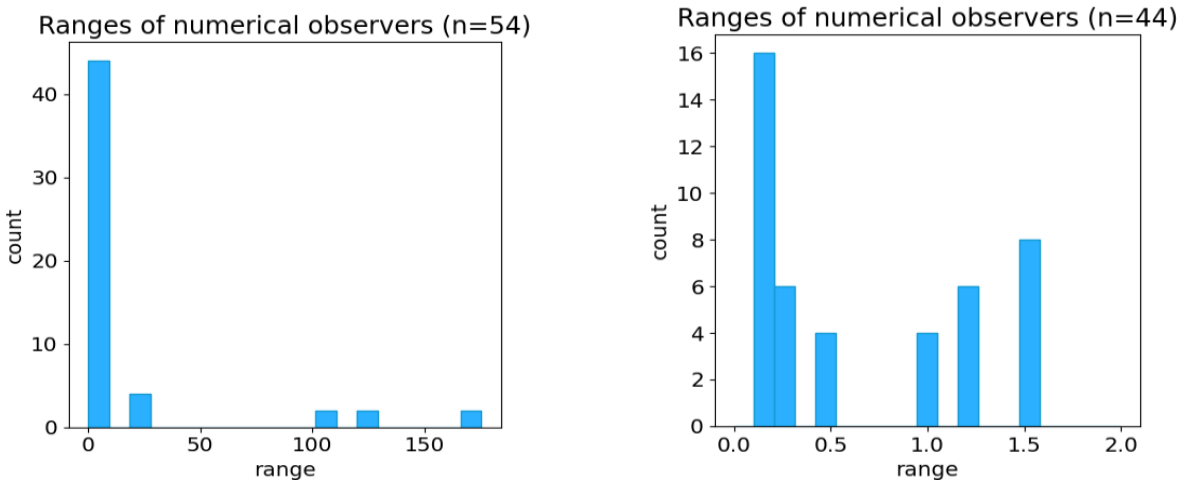


Figure 3.3: Histogram of the number of different states that categorical observers show throughout all system runs.

A similar analysis was performed for all numerical observers by recording their maximum range of values across all system runs. The corresponding histogram is shown in Figure 3.4a. The values of most numerical observers stay within a small range. 10 numerical observers however show a larger range of values. These observers record the positions of the conveyor belts in the MGR system, which increases over time as the conveyor belts only move in a single direction. An example of such an observer can be seen in Appendix A.1 in Figure A.2. Figure 3.4b shows a histogram of the numeric observers that stay within a small value range. These observers are almost exclusively observers with a periodic behavior. An example of such an observer can be seen in Appendix A.1 in Figure A.3. Due to the large discrepancies in the value ranges of different observers it is absolutely necessary to normalize the MGR data. The data was scaled using statistics robust to outliers. Namely, the median of each dimension was removed and data was scaled according to the 5th to 95th quantile range of each dimension.

Since the MGR system is strongly periodic the final step of the data analysis focused on the distribution of the periods of observers. A total of 69 observers show periodic behavior. Using the real-valued Fourier transformation, the periods of these observers were determined by computing the location of the maximum of the magnitude spectrum [39]. Figure 3.5 shows a histogram of the periods of all periodic observers. The vast majority of periodic observers show a period of 16 or 8 seconds. Note that it takes a tray 64 seconds to travel around the system and that there are four trays in total. Thus, observers with a period of 16 seconds record a full set of values for every tray that passes their corresponding machine part. Time windows for training statistical methods on periodic systems should be chosen in such a way that at least 2 periods are present [39]. For the MGR system this means that methods should be trained on time windows of at least 32 seconds. In order to account for potential disruption of periods through anomalies, the data was split into longer time windows of 60 seconds. Furthermore, the data sampling rate was informed by the minimum period present in the data. As can be seen in Figure 3.5, the minimum period is 2 seconds, corresponding to a maximum frequency of 0.5 Hz. The Nyquist-Shannon sampling theorem states that a signal can be fully reconstructed if the sampling frequency is at least twice the highest frequency component [39]. For the MGR data this minimal sampling frequency is thus 1Hz. In order to account for potential frequency disruption through anomalies, the data was recorded at a higher sampling rate of 2 Hz.



(a) (b)

Figure 3.4: (a) Histogram of the maximum range of values for every observer across all system runs. (b) Histogram of the maximum range of values for observers across all system runs, excluding observers with a value range larger than 2.

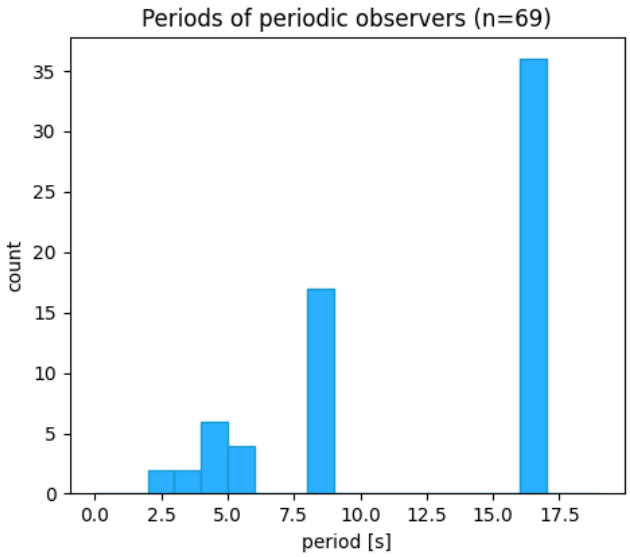


Figure 3.5: Histogram of the periods of all periodic observers.

3.2 Synthetic datasets

As previously mentioned, the MGR dataset only contains contextual group anomalies. In order to evaluate the developed methods for all other major types of anomalies as well as determine their performance under different generalized types of data, three synthetic datasets were created. The generated datasets contain mathematical and machine-control-inspired signals. Each dataset is 100-

dimensional and consists of 10 000 time-series datapoints. Table 3.2 describes the signal content of the first synthetic dataset, focusing on mathematical signals.[2] Examples of all mathematical signal types can be found in Appendix A.2.1 in Figures A.4 to A.7.

Function Type	Dimensions	Variation
Sine	33	frequency $\in [0.2, 2]$; phase $\in [0, 2\pi]$
Square wave	33	frequency $\in [0.2, 2]$; phase $\in [0, 2\pi]$
Composite sine	32	frequency $\in [0.2, 2]$; phase $\in [0, 2\pi]$
Linear	2	positive or negative slope

Table 3.2: Signal content of the first synthetic dataset. This dataset focuses on mathematical signals. Composite sine functions are made up of the product of two sine functions of different frequency and phase.

Table 3.3 describes the signal content of the second synthetic dataset, focusing on signals common in machine control settings. Sawtooth and triangle wave functions correspond to measuring angles when a machine moves in circles and measuring distance when it forwards and backwards. Motor functions correspond to the position, speed, acceleration and jerk signals resulting from a motor that periodically speeds up and slows down. Third-order point-to-point motion profiles were used. State patterns correspond to the signals resulting from internal state transitions of statemachines. Examples of all domain-inspired signal types can be found in Appendix A.2.2 in Figures A.8 to A.10.

Function Type	Dimensions	Variation
Sawtooth	20	frequency $\in [0.2, 2]$; sawtooth or triangle wave
Motor	40	frequency $\in [1, 5]$; jerk duration $\in [1, 5]$
State patterns	40	variable state pattern made up of 5 different states

Table 3.3: Signal content of the second synthetic dataset. This dataset focuses on on signals common in machine control settings.

Table 3.4 describes the signal content of the third synthetic dataset, combining both mathematical signals and signals common in machine control settings.

Function Type	Dimensions	Variation
Sine	16	frequency $\in [0.2, 2]$; phase $\in [0, 2\pi]$
Square wave	16	frequency $\in [0.2, 2]$; phase $\in [0, 2\pi]$
Composite sine	16	frequency $\in [0.2, 2]$; phase $\in [0, 2\pi]$
Linear	2	positive or negative slope
Sawtooth	10	frequency $\in [0.2, 2]$; sawtooth or triangle wave
Motor	20	frequency $\in [1, 5]$; jerk duration $\in [1, 5]$
State patterns	20	variable state pattern made up of 5 different states

Table 3.4: Signal content of the third synthetic dataset. This dataset combines both mathematical signals and signals common in machine control settings.

All synthetic datasets were created to contain clearly defined and localized point, contextual point, group and contextual group anomalies. Furthermore the number of dimensions in which an anomaly is

[2]Note that only two linear functions are included as min-max normalization equalizes linear functions of any slope and offset (except for positive and negative slope).

present can be varied, allowing for investigations into the responsiveness of a given anomaly detection method. Examples of all anomaly types of the synthetic datasets can be found in Appendix A.2.3 in Figures A.11 to A.14.

3.3 Performance evaluation metrics

As described in Section 2.1, the dataset setting of the anomaly detection task is a semi-supervised one - i.e. only training data considered to be normal is used to train the models. However, since the objective is to distinguish between normal and anomalous datapoints, performance will be measured with metrics used for binary classification tasks. After using the methods outlined in Section 2.4 to classify the anomaly scores obtained through the reconstruction models, the resulting predicted labels are compared to the true labels. From this comparison, the number of *true positives* (anomalous datapoints correctly classified to be anomalous), *true negatives* (normal datapoints correctly classified to be normal), *false positives* (normal datapoints incorrectly classified to be anomalous) and *false negatives* (anomalous datapoints incorrectly classified to be normal) are obtained. The methods developed in this thesis will be evaluated through sensitivity and specificity [40] in order to measure their performance in terms of the proportion of correctly identified anomalous and normal datapoints. For the purpose of visualizing the performance of a given method under multiple different anomaly score thresholds, receiver operating characteristic (ROC) curves [41] will be employed. The area under the ROC curve (AUC) will be used to quantitatively compare ROC curves resulting from different methods. Note that ROC curve and AUC are invariant under class imbalance [41], which is an important property given the unequal number of normal and anomalous datapoints in the Merry-Go-Round dataset (see Section 3.1.2).

In a machine control setting there are costs associated both with undetected anomalies as well as false alarms. Undetected anomalies may lead to costs in terms of damaged equipment or incorrectly manufactured products. False alarms may incur costs through redundant inspection or unnecessarily delayed production lines [42]. Depending on the setting, the costs associated with mis-classification of anomalous or normal datapoints will vary. Since this thesis aims to give an evaluation of performance without making assumptions on the specific costs of a particular setting, the F1-score [40], giving equal weight to undetected anomalies and false alarms, will be used to determine the best model in a given experiment. To correct for class imbalance, the F1-score will be calculated based on metrics normalized for class distribution.[3] Finally, the best models based on F1-score will be investigated via confusion matrices [40] to give an indication of the difficulty of classification of different types of anomalies.

For the purpose of evaluating architecture performance on the synthetic datasets, a different metric will be used. This metric, termed *test-calibration ratio* in this thesis, compares the mean anomaly score obtained on calibration (normal) data a_c with the mean anomaly score obtained on anomalous test data a_t. The test-calibration ratio r_{tc} is shown in Equation 3.1, with the number of anomalous test time-windows N_t, the number of (normal) calibration time-windows N_c, the anomaly score of a given calibration time-window a_c^w for time-window w and anomaly score of a given test time-window a_t^v for time-window v. This metric is used for the synthetic datasets since it directly evaluates

[3] I.e. the F1-score is calculated based on true positives, true negatives, false positives and false negatives which are normalized based on the number of nominal and anomalous datapoints. Note that the inherently balanced Matthews correlation coefficient [43] cannot be used since it runs into division-by-zero problems when a classifier predicts only a single class.

architecture performance without the intermediary of threshold evaluation. It was thus used for the purpose of generalizing performance in experiments on synthetic datasets. The test-calibration ratio can be interpreted as a kind of signal-to-noise ratio as it compares architecture response on anomalous data (ideally maximal) to architecture response on normal calibration data (ideally minimal). This metric is in the range of $[0, \infty]$ with 0 corresponding to the worst performance and larger values corresponding to better performance.[4]

$$r_{tc} = \frac{N_t}{N_c} \frac{\sum_{w=1}^{N_t} a_c^w}{\sum_{v=1}^{N_a} a_t^v} \tag{3.1}$$

In order to evaluate the performance of the anomaly diagnosis methodology laid out in Section 2.5, two metrics are proposed in this thesis. Recall that the objective of the observer contribution analysis is to identify anomalous dimensions. Thus, a model's performance in terms of this tool can be measured by evaluating how much anomalous dimensions contribute to the anomaly score compared to the total anomaly score. This metric, termed *anomalous observer contribution ratio* r_{obs}, is shown in Equation 3.2, with the number of anomalous time-windows N, anomaly score a_d^w for time-window w and dimension d, the number of anomalous dimensions D_a and total number of dimensions D. This metric is in the range of $[0, 1]$ with 0 corresponding to the worst performance and 1 corresponding to ideal performance.

$$r_{\text{obs}} = \frac{1}{N} \sum_{w=1}^{N} \frac{\sum_{d=1}^{D_a} a_d^w}{\sum_{d=1}^{D} a_d^w} \tag{3.2}$$

The objective of the attention matrix analysis is to identify the timestep-location of the anomaly. Thus, a model's performance in terms of this analysis can be measured by evaluating how often the attention matrix column with maximum attention corresponds to a timestep containing the anomaly compared to the total number of anomalous time-windows. This metric, termed *anomaly attention ratio* r_{att} is shown in Equation 3.3, with the number of anomalous time-windows N, indicator function $\mathbb{1}_A$ indicating whether a given timestep contains an anomaly, number of timesteps T and attention matrix M^n corresponding to time-window n.[5] This metric is in the range of $[0, 1]$ with 0 corresponding to the worst performance and 1 corresponding to ideal performance. Note that for the synthetic datasets anomalous timesteps take up 13.3% of all possible anomalous time-windows. Thus, random attention matrices would be expected to score 0.133 on this metric.

$$r_{\text{att}} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_A \left(\underset{c}{\text{argmax}} \Big(\sum_{t=1}^{T} M_{t\,c}^n \Big) \right), \tag{3.3}$$

where $M_{t\,c}$ is the element with row t and column c of attention matrix M.

[4] Note that an infinite test-calibration ratio will never be reached for properly set-up encoder-decoder architectures for anomaly detection since these have a loss of information in the encoding of the data.

[5] Note that Equation 3.3 performs the sum over the rows of the attention matrix as this sum corresponds to the attention pointing towards each timestep.

3.4 Experiments overview

This section presents an overview of the performed experiments, including their motivation with respect to the research questions, a description of dataset and training specifications as well as model architecture definitions. Experiments were performed on either the Merry-Go-Round (MGR) or synthetic datasets with individual datapoints corresponding to time-windows of length 120 as motivated in Section 3.1.3. The MGR dataset was split into training (containing 70% of normal datapoints), calibration (containing 10% of normal datapoints) and testing (containing 20% of normal and 100% of anomalous datapoints). The synthetic datasets were split into training (containing 90% of normal datapoints), calibration (containing 10% of normal datapoints) and testing (containing 100% of anomalous datapoints). This difference is due to different performance evaluation metrics used for the different datasets and will be explained below. Architectures for all experiments were trained using the Adam optimizer [44] with a learning rate of 0.001, beta-coefficients of $(0.9, 0.999)$ and ϵ numerical stability parameter of 10^{-8}. These parameter choices correspond to those suggested by Kingma et al. [44]. With the exception of Experiment 8, all architectures were trained for 100 epochs on the respective dataset. All experiments on the MGR dataset were realized in terms of 5-fold cross-validation where training, calibration and normal testing dataset are split. All experiments on the synthetic datasets were realized in terms of 5-fold cross-validation where training and calibration dataset are split. Performance metrics were recorded according to their mean and standard deviation across cross-validations.

Experiments 1 to 3 presented in Section 4.1 aim to answer research question (RQ) 1. RQ 1 is concerned with investigating the performance of Undercomplete Autoencoders, LSTM Autoencoders and Transformer encoder-decoders for anomaly detection in machine control applications and machine setups. For these experiments, the aforementioned models are trained and tested on the MGR dataset. The resulting anomaly scores are classified using the threshold types presented in Section 2.4. RQ 1a will be answered by providing anomaly detection performance in terms of anomaly score on calibration and testset, AUC on the ROC-curve, best F1-score and true positive and false positive rates of the threshold corresponding to the best F1-score. In order to answer RQ 1b, ROC curves and anomaly detection performance will be presented separately for each threshold type.[6] To answer RQ 1c classification performance will be analyzed according to anomaly type. In particular, Experiment 1 evaluates all of the above in terms of the Undercomplete Autoencoder model, Experiment 2 in terms of the LSTM Autoencoder and Experiment 3 in terms of the Transformer. The exact architecture definitions can be found in Appendix A.3 in Tables A.1 to A.3.

Experiments 4 to 9 presented in Section 4.2 aim to answer RQ 2. RQ 2 is concerned with investigating the anomaly detection performance of the proposed Self-Attention Autoencoder architecture on digital twin data as well as generalizing its performance for synthetic mathematical and machine-control-domain-based signals. For Experiment 4 the Self-Attention Autoencoder proposed in Section 2.3 was trained and tested on the MGR dataset. The resulting anomaly scores are classified using the threshold types presented in Section 2.4. RQ 2a will be answered by performing all previously described analyses equivalently for the Self-Attention Autoencoder model. The architecture definition for this experiment can be found in Appendix A.3 in Table A.4. Experiments 5 to 9 aim to generalize the performance of the Self-Attention Autoencoder across different data types and anomaly classes.

[6] Threshold factor α was varied in the range of $[0, 10]$ for the mean threshold, and in the range of $[-10, 10]$ for the standard deviation threshold. This is to ensure a full ROC curve for these threshold types. For the series threshold type, every possible permutation of the ruleset was evaluated.

Thus, anomaly detection performance is evaluated in terms of test-calibration ratio, which offers a neutral measure of architecture performance independent of threshold type. Since test-calibration ratio is defined in terms of architecture performance on normal (calibration) and anomalous (test) data, the associated testing data only contains anomalous time-windows. For these experiments the Self-Attention Autoencoder is trained and tested on the mathematical, domain-related and mixed synthetic datasets. Its architecture definition can be found in Appendix A.3 in Table A.5. RQ 2b, 2c and 2d are answered by evaluating architecture performance while varying the number of anomalous dimensions in the test data in the range of $[1, 100]$. This investigation is performed separately for all types of data[7] as well as all anomaly classes.[8] RQ 2e is answered by comparing test-calibration ratio with training error for the Self-Attention Autoencoder trained for 2 to 100 epochs. RQ 2f is answered by evaluating architecture performance when adding varying amounts of normally distributed noise to the calibration and testing data. Noise standard deviation is varied in the range of $[0\%, 25\%]$ of the value-range of the original signals. Note that noise was only added to the calibration and testing data (and not to the training data) in order to simulate an industrial setting where models are trained on synthetic digital twin data and subsequently deployed on the factory floor encountering noisy real data.

Experiments 10 and 11 presented in Section 4.3 aim to answer RQ 3. RQ 3 is concerned with investigating the anomaly diagnosis performance of the Transformer and Self-Attention Autoencoder using the methodology proposed in Section 2.5. For these experiments, both models are trained and tested on the mathematical, domain-related and mixed synthetic datasets. Note that these datasets were created to contain clearly defined and localized anomalies and thus allow for anomaly diagnosis evaluation. RQ 3a and 3b are answered by determining the anomalous observer contribution ratio (Eq. 3.2) and anomaly attention ratio (Eq. 3.3) for the Transformer and Self-Attention Autoencoder. As described in Section 3.3 these metrics serve as a quantitative measure of anomaly diagnosis performance. The architecture definitions for Experiments 10 and 11 can be found in Appendix A.3 in Tables A.5 and A.6.

[7] Mathematical data in experiment 5, domain-related data in experiment 6 and mixed data in experiment 7.

[8] Point anomaly, group anomaly, contextual point anomaly and contextual group anomaly.

Chapter 4

Results

This chapter presents the results of the experiments outlined in Section 3.4, including:

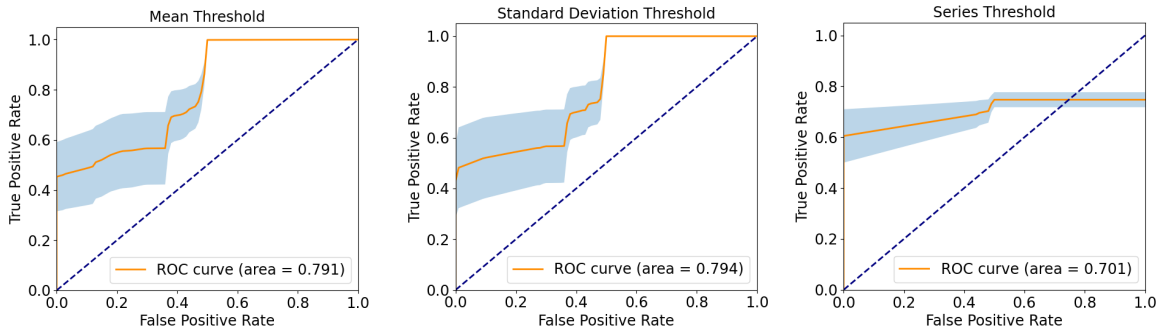
- An evaluation of the anomaly detection performance of existing encoder-decoder architectures on the Merry-Go-Round dataset. [Section 4.1]
- A comprehensive investigation into the anomaly detection performance of the Self-Attention Autoencoder architecture proposed in this thesis. This includes evaluation on the Merry-Go-Round as well as mathematical, domain-based and mixed synthetic datasets. [Section 4.2]
- A quantitative analysis of the interpretability of the Transformer and Self-Attention Autoencoder architectures for the task of anomaly diagnosis on the synthetic datasets. [Section 4.3]

4.1 Anomaly detection performance of existing encoder-decoder architectures

4.1.1 Experiment 1: Undercomplete Autoencoder

Experiment 1 compares the anomaly detection performance of the Undercomplete Autoencoder architecture under the different types of threshold classifiers described in Section 2.4. Figure 4.1 shows the mean ROC-curves for each threshold type evaluated on 5 cross-validation models trained on the MGR dataset. Note the steep incline in *true positive rate* for the mean and standard deviation threshold types. This is a trend that will repeat across all architectures trained and evaluated on the Merry-Go-Round dataset, which suggests that it results from structure present in the data. This interpretation will be further discussed in Section 5. It should additionally be noted that both mean and standard deviation threshold types perform comparatively whereas the Series threshold type relatively underperforms. The full application of all Western Electric rules is not sufficient to detect all anomalies[1] as the maximum mean *true positive rate* reached is 0.75.

[1] As a result, the (1,1) point in the ROC-plot is not reached by the Series threshold type.



(a) (b) (c)

Figure 4.1: ROC plots for the Undercomplete Autoencoder model evaluated on 5 cross-validations on the MGR dataset. Mean ROC curve shown in orange; ± 1 standard deviation shaded in blue. (a) ROC plot for the mean threshold type. (b) ROC plot for the standard deviation threshold type (c) ROC plot for the series threshold type.

Table 4.1 displays the anomaly scores of normal and anomalous time-windows of the test dataset evaluated using the Undercomplete Autoencoder model. The mean anomaly score on normal time-windows is significantly lower than that on anomalous time-windows. However, there is a very large standard deviation for both normal and anomalous time-windows, implying that some time-windows are assigned particularly large anomaly scores. Table 4.2 shows the mean and standard deviation of the AUC, F1-score, true positive rate and false positive rate for each threshold type. Note that the best F1-score across threshold values or rulesets for a given threshold type is displayed and that true and false positive rates are displayed according to best F1-score. We draw attention to the large standard deviation present across all threshold types for the false positive rate of the best F1-score. This implies that the accurate classification of normal data varies considerably across different cross-validations. The significance of this will be further discussed in Section 5.

a_t (norm.)	a_t (anom.)
0.0214 ± 0.41	0.141 ± 1.2

Table 4.1: Anomaly scores on normal and anomalous test data for the Undercomplete Autoencoder model. Mean and standard deviation obtained from 5 cross-validations on the MGR dataset.

Threshold Type	AUC	Best F1	TPR	FPR
Mean	0.791 ± 0.10	0.828 ± 0.065	0.882 ± 0.065	0.249 ± 0.25
Standard Deviation	0.783 ± 0.10	0.818 ± 0.11	0.886 ± 0.056	0.279 ± 0.22
Series	0.701 ± 0.071	0.741 ± 0.10	0.647 ± 0.088	0.0983 ± 0.20

Table 4.2: Anomaly detection performance of the Undercomplete Autoencoder model. Mean and standard deviation of AUC, best F1-score, true positive rate (TPR) and false positive rate (FPR) are displayed for each threshold type. TPR and FPR are displayed according to best F1-score. Threshold types are evaluated on 5 cross-validations of the Undercomplete Autoencoder model trained on the MGR dataset.

Table 4.3 displays the predicted label rates for normal and anomalous datapoints. Predicted label rates

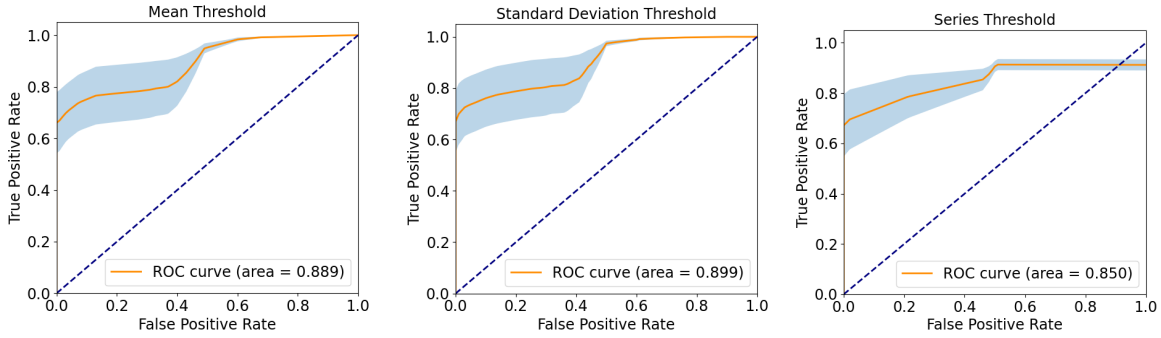
are shown for the best threshold type (Mean) and threshold value according to F1-score. Note that the belt anomaly type is perfectly classified across all cross-validations. Conversely, worst classification performance is found for normal datapoints and the gantry anomaly type.

	Predicted Normal	Predicted Anomalous
Normal	0.751 ± 0.25	0.249 ± 0.25
Tray Anomaly	0.098 ± 0.08	0.902 ± 0.08
Belt Anomaly	0 ± 0	1 ± 0
Gantry Anomaly	0.204 ± 0.12	0.746 ± 0.12

Table 4.3: Predicted label rates for normal and anomalous datapoints for the Undercomplete Autoencoder model evaluated on 5 cross-validations on the MGR dataset. Values are displayed for the best threshold type (Mean) and threshold value according to F1-score.

4.1.2 Experiment 2: LSTM Autoencoder

Experiment 2 compares the anomaly detection performance of the LSTM Autoencoder architecture under the different types of threshold classifiers described in Section 2.4. Figure 4.2 shows the mean ROC-curves for each threshold type evaluated on 5 cross-validation models trained on the MGR dataset. Note the improved performance across every threshold type compared to the Undercomplete Autoencoder presented in the previous section. Similar to the previous experiment it should further be noted that the Series threshold type under-performs in comparison to the other threshold types.



(a) (b) (c)

Figure 4.2: ROC plots for the LSTM Autoencoder model evaluated on 5 cross-validations on the MGR dataset. Mean ROC curve shown in orange; ± 1 standard deviation shaded in blue. (a) ROC plot for the mean threshold type. (b) ROC plot for the standard deviation threshold type (c) ROC plot for the series threshold type.

Table 4.4 displays the anomaly scores of normal and anomalous time-windows of the test dataset evaluated using the LSTM Autoencoder model. While the mean anomaly scores on normal and anomalous time-windows are more similar than for the Undercomplete Autoencoder model, the standard deviation of the normal anomaly score is lower for the LSTM Autoencoder model, allowing it to overall

show a better anomaly detection performance. Table 4.5 shows the mean and standard deviation of the AUC, F1-score, true positive rate and false positive rate for each threshold type. Note that the true positive rates of the best F1-scores are comparable to those of the Undercomplete Autoencoder presented in the previous section. The overall improvement in performance is due to lower false positive rates.

a_t **(norm.)**	a_t **(anom.)**
0.125 ± 0.13	0.393 ± 1.9

Table 4.4: Anomaly scores on normal and anomalous test data for the LSTM Autoencoder model. Mean and standard deviation obtained from 5 cross-validations on the MGR dataset.

Threshold Type	**AUC**	**Best F1**	**TPR**	**FPR**
Mean	0.889 ± 0.075	0.863 ± 0.099	0.881 ± 0.11	0.161 ± 0.23
Standard Deviation	0.899 ± 0.075	0.867 ± 0.095	0.887 ± 0.051	0.160 ± 0.19
Series	0.850 ± 0.062	0.863 ± 0.095	0.869 ± 0.066	0.145 ± 0.20

Table 4.5: Anomaly detection performance of the LSTM Autoencoder model. Mean and standard deviation of AUC, best F1-score, true positive rate (TPR) and false positive rate (FPR) are displayed for each threshold type. TPR and FPR are displayed according to best F1-score. Threshold types are evaluated on 5 cross-validations of the LSTM Autoencoder model trained on the MGR dataset.

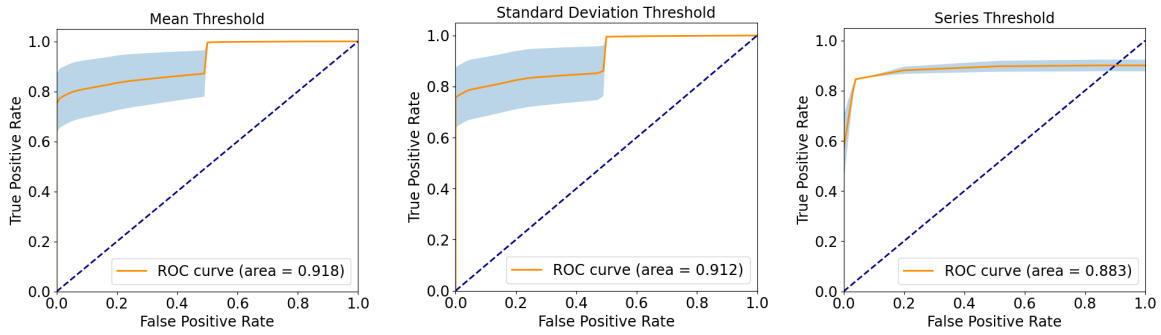
Table 4.6 displays the predicted label rates for normal and anomalous datapoints. The LSTM Autoencoder shows a low rate of mis-classification for both tray and belt anomaly types and a comparatively higher mis-classification rate for normal datapoints and the gantry anomaly type.

	Predicted Normal	Predicted Anomalous
Normal	0.840 ± 0.19	0.160 ± 0.19
Tray Anomaly	0.083 ± 0.0088	0.917 ± 0.0088
Belt Anomaly	0.084 ± 0.071	0.916 ± 0.071
Gantry Anomaly	0.174 ± 0.11	0.826 ± 0.11

Table 4.6: Predicted label rates for normal and anomalous datapoints for the LSTM Autoencoder model evaluated on 5 cross-validations on the MGR dataset. Values are displayed for the best threshold type (Standard Deviation) and threshold value according to F1-score.

4.1.3 Experiment 3: Transformer

Experiment 3 compares the anomaly detection performance of the Transformer architecture under the different types of threshold classifiers described in Section 2.4. Figure 4.3 shows the mean ROC-curves for each threshold type evaluated on 5 cross-validation models trained on the MGR dataset. Note once again the sharp increase in anomaly detection performance for the mean and standard deviation threshold types. Further note the low variance in the performance of the series threshold type. Despite lower overall performance of the series threshold type, this provides upsides which will be discussed in Section 5.



(a) (b) (c)

Figure 4.3: ROC plots for the Transformer model evaluated on 5 cross-validations on the MGR dataset. Mean ROC curve shown in orange; ± 1 standard deviation shaded in blue. (a) ROC plot for the mean threshold type. (b) ROC plot for the standard deviation threshold type (c) ROC plot for the series threshold type.

Table 4.7 displays the anomaly scores of normal and anomalous time-windows of the test dataset evaluated using the Transformer model. The mean anomaly score on anomalous time-windows is more than an order of magnitude larger than that on normal time-windows. This leads to further improved anomaly detection performance for the Transformer model compared to the previously evaluated architectures. Table 4.8 shows the mean and standard deviation of the AUC, F1-score, true positive rate and false positive rate for each threshold type. Note that the false positive rates of the best F1-scores are fairly low for the Transformer architecture type. The false positive rate of the best series threshold ruleset in particular is very low and additionally shows very low variance.

a_t (norm.)	a_t (anom.)
0.00692 ± 0.045	0.0981 ± 1.3

Table 4.7: Anomaly scores on normal and anomalous test data for the Transformer model. Mean and standard deviation obtained from 5 cross-validations on the MGR dataset.

Threshold Type	AUC	Best F1	TPR	FPR
Mean	0.912 ± 0.084	0.905 ± 0.074	0.904 ± 0.062	0.094 ± 0.18
Standard Deviation	0.917 ± 0.084	0.902 ± 0.074	0.908 ± 0.058	0.105 ± 0.18
Series	0.883 ± 0.0053	0.904 ± 0.014	0.835 ± 0.0088	0.0124 ± 0.017

Table 4.8: Anomaly detection performance of the Transformer model. Mean and standard deviation of AUC, best F1-score, true positive rate (TPR) and false positive rate (FPR) are displayed for each threshold type. TPR and FPR are displayed according to best F1-score. Threshold types are evaluated on 5 cross-validations of the Transformer model trained on the MGR dataset.

Table 4.9 displays the predicted label rates for normal and anomalous datapoints. The Transformer architecture shows a very low rate of mis-classification for the belt anomaly type and a comparatively higher rate of mis-classification for the gantry anomaly type. Both normal datapoints and the tray anomaly type are mis-classified at a relatively low rate.

	Predicted Normal	Predicted Anomalous
Normal	0.906 ± 0.18	0.094 ± 0.18
Tray Anomaly	0.073 ± 0.015	0.927 ± 0.015
Belt Anomaly	0.039 ± 0.027	0.961 ± 0.027
Gantry Anomaly	0.175 ± 0.149	0.825 ± 0.149

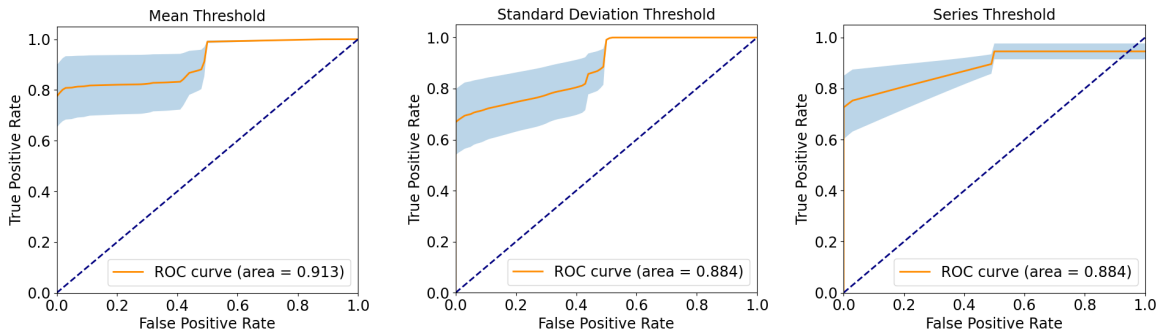
Table 4.9: Predicted label rates for normal and anomalous datapoints for the Transformer model evaluated on 5 cross-validations on the MGR dataset. Values are displayed for the best threshold type (Mean) and threshold value according to F1-score.

4.2 Anomaly detection performance of the Self-Attention Autoencoder

This section presents a comprehensive investigation into the anomaly detection performance of the Self-Attention Autoencoder architecture proposed in this thesis. In order to establish its performance in comparison to the existing encoder-decoder architectures, the Self-Attention Autoencoder is first evaluated on the Merry-Go-Round dataset. Subsequently, it will be evaluated extensively on synthetic datasets to obtain a more general understanding of its performance on different types of data and anomalies.

4.2.1 Experiment 4: Anomaly detection on the Merry-Go-Round dataset

Experiment 4 compares the anomaly detection performance of the Self-Attention Autoencoder architecture under the different types of threshold classifiers described in Section 2.4. Figure 4.4 shows the mean ROC-curves for each threshold type evaluated on 5 cross-validation models trained on the MGR dataset. Note that the ROC-curve of the mean threshold type is very similar to that of the Transformer model discussed in Section 4.1.3. The ROC-curves of the Standard Deviation and Series threshold types differ from those of the Transformer model but still show competitive performance in terms of AUC.



(a) (b) (c)

Figure 4.4: ROC plots for the Self-Attention Autoencoder model evaluated on 5 cross-validations on the MGR dataset. Mean ROC curve shown in orange; ± 1 standard deviation shaded in blue. (a) ROC plot for the mean threshold type. (b) ROC plot for the standard deviation threshold type (c) ROC plot for the series threshold type.

Table 4.10 displays the anomaly scores of normal and anomalous time-windows of the test dataset evaluated using the Self-Attention Autoencoder model. While the mean anomaly score for normal and anomalous datapoints is closer than for the Transformer model, the standard deviation of the anomaly scores is significantly lower (in a relative sense) for the Self-Attention Autoencoder model. Table 4.11 shows the mean and standard deviation of the AUC, F1-score, true positive rate and false positive rate for each threshold type. Overall, performance closely resembles that of the Transformer model. Note however, that the mean and variance of the false positive rate of the Series threshold is very large in comparison to that of the Transformer model.

a_t **(norm.)**	a_t **(anom.)**
0.0719 ± 0.13	0.304 ± 1.8

Table 4.10: Anomaly scores on normal and anomalous test data for the Self-Attention Autoencoder model. Mean and standard deviation obtained from 5 cross-validations on the MGR dataset.

Threshold Type	**AUC**	**Best F1**	**TPR**	**FPR**
Mean	0.913 ± 0.085	0.918 ± 0.086	0.938 ± 0.015	0.105 ± 0.20
Standard Deviation	0.884 ± 0.082	0.877 ± 0.096	0.946 ± 0.0061	0.211 ± 0.25
Series	0.884 ± 0.064	0.894 ± 0.080	0.894 ± 0.092	0.106 ± 0.20

Table 4.11: Anomaly detection performance of the Self-Attention Autoencoder model. Mean and standard deviation of AUC, best F1-score, true positive rate (TPR) and false positive rate (FPR) are displayed for each threshold type. TPR and FPR are displayed according to best F1-score. Threshold types are evaluated on 5 cross-validations of the Self-Attention Autoencoder model trained on the MGR dataset.

Table 4.12 displays the predicted label rates for normal and anomalous datapoints. Note that, similar to the Undercomplete Autoencoder model, the belt anomaly type is perfectly classified across all cross-validations. Aside from this, the classification performance of the Self-Attention Autoencoder model is comparable to that of the Transformer model.

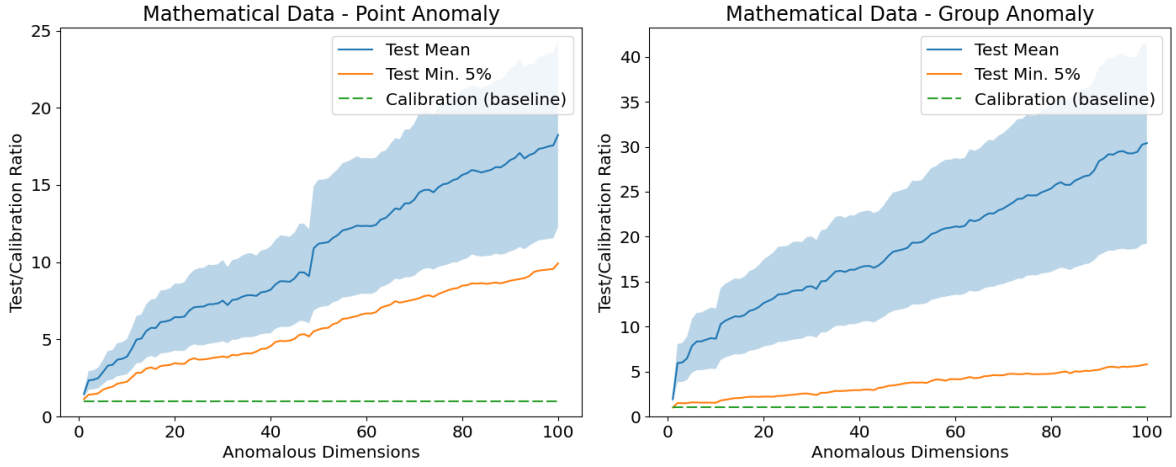
	Predicted Normal	Predicted Anomalous
Normal	0.895 ± 0.20	0.105 ± 0.20
Tray Anomaly	0.063 ± 0.016	0.937 ± 0.016
Belt Anomaly	0 ± 0	1 ± 0
Gantry Anomaly	0.123 ± 0.031	0.877 ± 0.031

Table 4.12: Predicted label rates for normal and anomalous datapoints for the Self-Attention Autoencoder model evaluated on 5 cross-validations on the MGR dataset. Values are displayed for the best threshold type (Mean) and threshold value according to F1-score.

4.2.2 Experiment 5: Architecture performance on mathematical signals under varying anomalous dimensions

Experiment 5 aims to generalize the performance of the Self-Attention Autoencoder to data consisting of mathematical signals, as presented in Table 3.2. As described in Section 3.4 performance is evaluated depending on the number of dimensions in which an anomaly signal was injected. Furthermore,

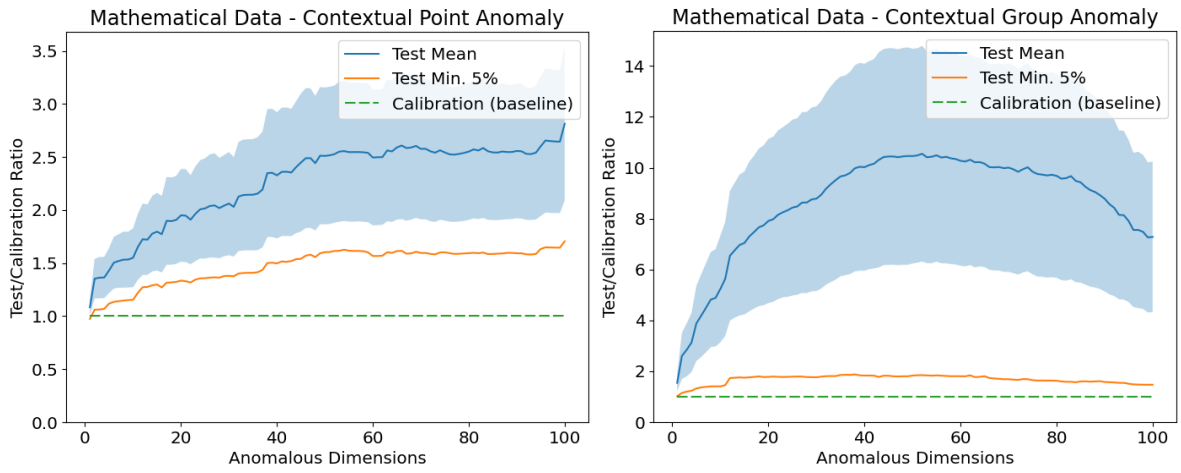
performance is investigated for all main classes of anomalies, as described in Section 2. Figure 4.5 shows architecture performance on point and group anomalies. Note that performance is measured via the ratio of test anomaly score (at anomalies) to calibration anomaly score. As described in Section 3.3 this metric presents a more generalized perspective on architecture performance since it is independent of threshold choice. Note that, as the number of dimensions with anomalous signals increases, the test-calibration ratio also predominantly increases for both point and group type anomalies. At this point the question might arise why the test-calibration ratio does not monotonically increase with increasing number of anomalous dimensions. This is due to the nature of the encoder-decoder model, in which input dimensions are compressed into a smaller representation within the bottleneck. This will be more thoroughly discussed in Section 5.



(a) (b)

Figure 4.5: Architecture performance plots for the Self-Attention Autoencoder model under varying number of anomalous dimensions. Plots resulting from evaluation of 5 cross-validations on the synthetic mathematical dataset. Mean test-calibration ratio shown in dark blue; ± 1 standard deviation shaded in light blue; minimum 5% quantile of the test-calibration ratio shown in orange. (a) Architecture performance plot for the point type anomaly. (b) Architecture performance plot for the group type anomaly.

Figure 4.5 shows architecture performance of the Self-Attention Autoencoder model on contextual point and contextual group anomalies. Note that both contextual anomaly types lead to significantly lower test-calibration ratios than the non-contextual anomalies. Further note that test-calibration ratios level out for the contextual point anomaly type after approximately 50 anomalous dimensions. Finally, note the remarkable trend in decreasing test-calibration ratios for the contextual group anomaly after approximately 50 anomalous dimensions. The cause of these trends will be discussed in Section 5.



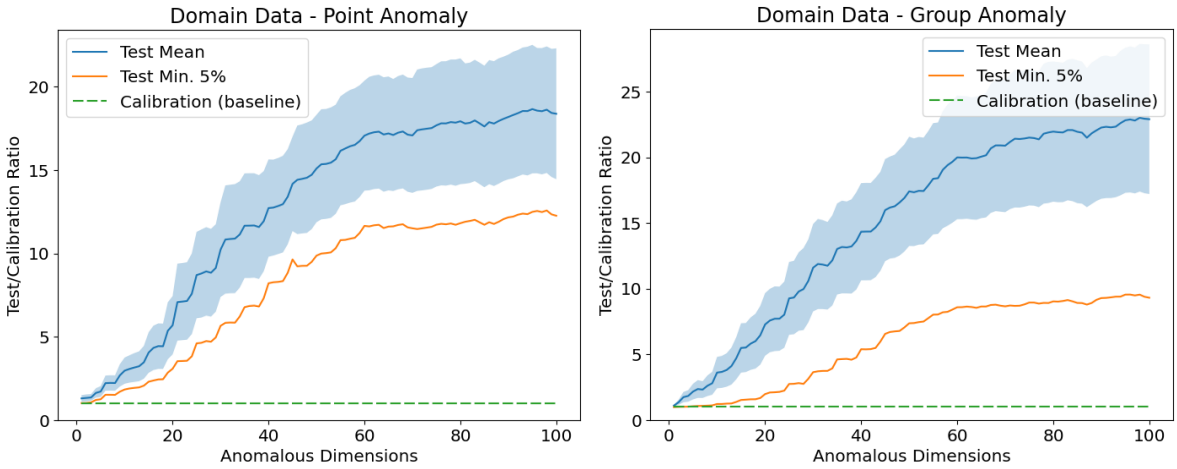
(a) (b)

Figure 4.6: Architecture performance plots for the Self-Attention Autoencoder model under varying number of anomalous dimensions. Plots resulting from evaluation of 5 cross-validations on the synthetic mathematical dataset. Mean test-calibration ratio shown in dark blue; ± 1 standard deviation shaded in light blue; minimum 5% quantile of the test-calibration ratio shown in orange. (a) Architecture performance plot for the contextual point type anomaly. (b) Architecture performance plot for the contextual group type anomaly.

4.2.3 Experiment 6: Domain data

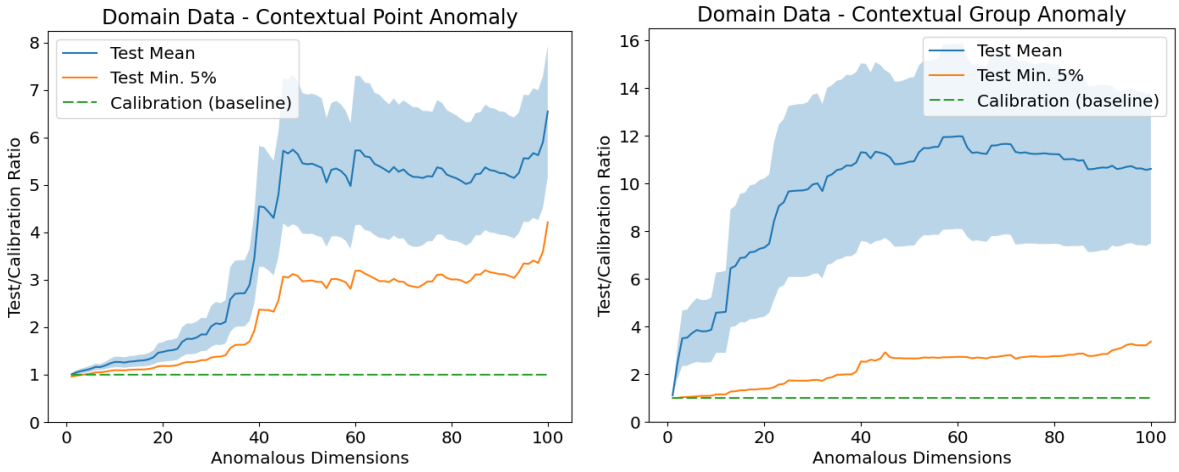
Experiment 6 evaluates the performance of the Self-Attention Autoencoder on data consisting of machine-control domain-related signals, as presented in Table 3.3. Figure 4.7 shows architecture performance on point and group anomalies. Similarly to the performance on the synthetic mathematical dataset, the test-calibration ratio on the domain-related data predominantly increases for both point and group type anomalies. However, there are clearly visible "steps" in performance for the domain-related data. The likely cause of these will be discussed in Section 5.

Figure 4.7 shows architecture performance of the Self-Attention Autoencoder model on contextual point and contextual group anomalies. Note the generally lower test-calibration ratios for both contextual anomaly types compared to the non-contextual anomalies. Further note that test-calibration ratios level out for the point contextual anomaly type and slightly decrease for the contextual group anomaly type after approximately 50 anomalous dimensions.



(a) (b)

Figure 4.7: Architecture performance plots for the Self-Attention Autoencoder model under varying number of anomalous dimensions. Plots resulting from evaluation of 5 cross-validations on the synthetic domain-related dataset. Mean test-calibration ratio shown in dark blue; ± 1 standard deviation shaded in light blue; minimum 5% quantile of the test-calibration ratio shown in orange. (a) Architecture performance plot for the point type anomaly. (b) Architecture performance plot for the group type anomaly.



(a) (b)

Figure 4.8: Architecture performance plots for the Self-Attention Autoencoder model under varying number of anomalous dimensions. Plots resulting from evaluation of 5 cross-validations on the synthetic domain-related dataset. Mean test-calibration ratio shown in dark blue; ± 1 standard deviation shaded in light blue; minimum 5% quantile of the test-calibration ratio shown in orange. (a) Architecture performance plot for the contextual point type anomaly. (b) Architecture performance plot for the contextual group type anomaly.

4.2.4 Experiment 7: Mixed data

Experiment 7 evaluates the performance of the Self-Attention Autoencoder on mixed data as presented in Table 3.4. Architecture performance for this experiment appears to be a combination of the performance of Experiment 5 and 6. This is unsurprising given the data consists of both mathematical and machine-control domain-related signals. For the sake of brevity, the performance graphs for this experiment are shown in Appendix A.4.1 in Figures A.15 and A.16.

4.2.5 Experiment 8: Training performance indicativeness

Experiment 8 investigates whether the training performance of the Self-Attention Autoencoder is a good indicator for its test performance. Figure 4.9 shows the test-calibration ratio[2] as well as training reconstruction error for an increasing number of epochs of training on the mathematical data. Note that as the number of epochs increases, the training error decreases and the test-calibration ratio increases. However, it should be pointed out that the relationship between training error and test-calibration error is non-linear. Note in particular the last 20 epochs of training, in which the training error only decreases slightly, but the test-calibration ratio still increases considerably. Since the training error is only measured on normal data, this suggests that adaptations to the model's parameters which slightly improve reconstruction on normal data can have significant impact on reconstruction of anomalous data.

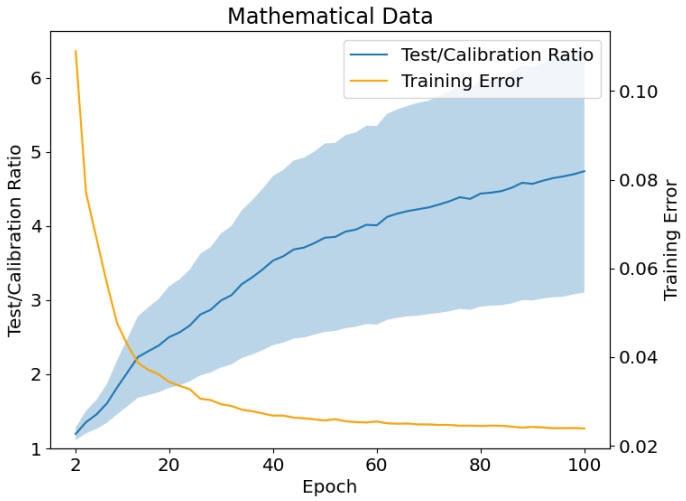


Figure 4.9: Test-calibration ratio and training reconstruction error for varying number of epochs of training on the synthetic mathematical data. Note that the number of anomalous dimensions was set to 10 for this experiment. Further note that the test/calibration ratio is averaged over all types of anomalies.

Similar trends were found for the domain-based and mixed synthetic data. Figures displaying the test-calibration ratio and training reconstruction error for varying number of epochs of training on these datasets can be found in Appendix A.4.2 in Figures A.17 and A.18.

[2]Note that the test-calibration ratio was averaged over all types of anomalies for this figure.

4.2.6 Experiment 9: Robustness to Noise

Experiment 9 aims to evaluate the robustness of the Self-Attention Autoencoder to noise on the calibration and test data. As described in Section 3.4, this experimental setup is chosen to simulate training the reconstruction architecture on data from a digital twin and subsequently deploying it in a noisy real world environment. Note that the standard deviation of the noise was varied between 0 to 25% of the value-range of the original signals. A visualization of what a noise of 25% corresponds to can be found in Appendix A.4.3 in Figure A.19. Figure 4.10 shows the test-calibration ratio on the synthetic mathematical data under varying amounts of noise. Note the clear decrease in architecture performance as increasing amounts of noise are added.

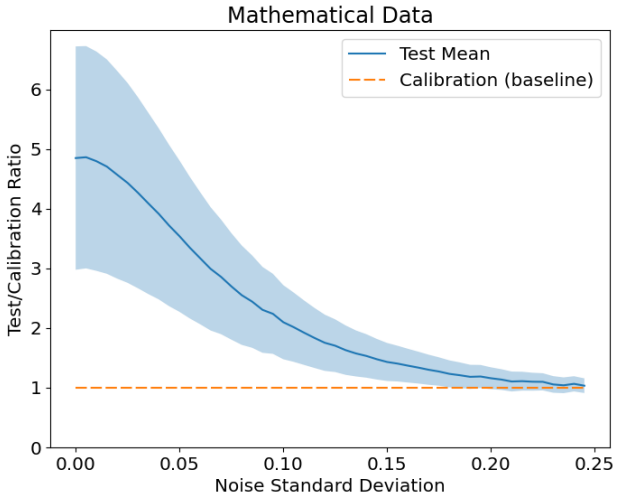


Figure 4.10: Test-calibration ratio on the synthetic mathematical data under varying amounts of noise added to test and calibration data. Noise standard deviation is given in % of the value-range of the original signals.

Similar trends were found for the domain-based and mixed synthetic data. Figures displaying the test-calibration ratio under varying amounts of noise for these datasets can be found in Appendix A.4.3 in Figures A.20 and A.21.

4.3 Diagnosis and interpretability evaluation

The experiments within this section investigate the performance of the anomaly diagnosis methodology proposed in Section 2.5. The methodology will be evaluated quantitatively in terms of anomalous observer contribution ratio (see: Equation 3.2) and anomaly attention ratio (see: Equation 3.3) on both the Transformer and Self-Attention Autoencoder model.

4.3.1 Experiment 10: Quantitative diagnosis and interpretability of the Transformer model

Table 4.13 shows the anomalous observer contribution ratio r_{obs} and anomaly attention ratio r_{att} for the Transformer model trained on the mathematical, domain-related and mixed synthetic datasets. Note that the number of anomalous dimensions was set to 10 for this experiment. Since the total number of dimensions in the synthetic datasets is 100, the expected r_{obs} of a random model would be 0.1. For the Transformer model about 20% of anomaly score contributions originate from actually anomalous dimensions. Section 5 will discuss why this contribution is not higher yet why it is still sufficient for high test-calibration ratios.[3]

Regarding the anomaly attention ratio r_{att} it should be pointed out that its expected value under random or un-informative attention matrices is 0.133.[4] As can be seen in Table 4.13, the Transformer model shows very low values of r_{att} for both the mathematical and domain-related synthetic datasets and better (yet still low) values on the mixed dataset.

Data Type	$r_{\textbf{obs}}$	$r_{\textbf{att}}$
Mathematical	0.217 ± 0.16	0.146 ± 0.14
Domain	0.174 ± 0.12	0.102 ± 0.087
Mixed	0.195 ± 0.14	0.289 ± 0.27

Table 4.13: Anomalous observer contribution ratio r_{obs} and anomaly attention ratio r_{att} for the Transformer model trained on the mathematical, domain-related and mixed synthetic datasets. Performance evaluated on point and contextual point anomalies as group and contextual group anomalies take up the entire time-window and are not indicative for evaluation under r_{att}.

4.3.2 Experiment 11: Quantitative diagnosis and interpretability of the Self-Attention Autoencoder

Table 4.14 shows the anomalous observer contribution ratio r_{obs} and anomaly attention ratio r_{att} for the Self-Attention Autoencoder trained on the mathematical, domain-related and mixed synthetic datasets. The model's performance in terms of r_{obs} is very similar to the Transformer model. Note however, that the standard deviation of this metric is considerably lower for the Self-Attention Autoencoder compared to the Transformer.

The anomaly attention ratios r_{att} obtained from the Self-Attention Autoencoder are significantly higher than those obtained from the Transformer model. In 50 to 60% of cases the Self-Attention Autoencoder focuses its attention on the actual timestep-location of the anomaly. In other words, the attention matrices of the Self-Attention Autoencoder contain valuable information for anomaly diagnosis.[5]

[3] Note that despite similarly low r_{obs} for the *Self-Attention Autoencoder* overall high test-calibration ratios are observed for it in Sections 4.2.3 to 4.2.5.

[4] This is due to anomalous timesteps taking up 13.3% of all possible anomalous time-windows.

[5] It should however be noted that for cases in which attention is focused on non-anomalous timesteps, the focus of attention does not seem to follow a clear pattern.

Data Type	$r_{\textbf{obs}}$	$r_{\textbf{att}}$
Mathematical	0.216 ± 0.075	0.511 ± 0.23
Domain	0.186 ± 0.054	0.556 ± 0.24
Mixed	0.187 ± 0.083	0.610 ± 0.28

Table 4.14: Anomalous observer contribution ratio r_{obs} and anomaly attention ratio r_{att} for the Self-Attention Autoencoder model trained on the mathematical, domain-related and mixed synthetic datasets. Performance evaluated on point and contextual point anomalies.

Chapter 5

Discussion

This chapter will discuss the experimental results obtained in order to answer the research questions posed in this thesis. It will discuss noteworthy observations resulting from the experiments, describe the limitations of the presented work and propose suggestions for further research.

5.1 Research question 1

Research question (RQ) 1 concerns the performance of existing machine learning methods for the detection of anomalies in industrial systems. The methods investigated on the MGR dataset included Undercomplete Autoencoders in Experiment 1, LSTM Autoencoders in Experiment 2 and Transformer Encoder-Decoders in Experiment 3. Anomaly detection performance (RQ 1a) was the worst for the Undercomplete Autoencoder, with a mean AUC of 0.758 across threshold types. This model was chosen as a baseline due to its simplicity and its poor performance makes sense given that it is not specifically designed for time-series data. The LSTM Autoencoder, which is designed specifically for time-series data, performed better, with a mean AUC of 0.879 across threshold types. The best performing architecture among existing machine learning methods was the Transformer model, which had a mean AUC of 0.904 across threshold types. As argued in Section 2.2.3, the non-sequential nature of Transformers provides advantages for long-range temporal dependencies, which might have contributed to their better performance.

Regarding anomaly score evaluation methods (RQ 1b), similar performance was observed for the mean and standard deviation threshold types across experiments 1 to 3, while the series threshold type performed unanimously worse. However, note that the series threshold type for the Transformer architecture had a particularly low false positive rate. This would be a desirable property in a setting where false alarms are costly.

With respect to different anomaly types (RQ 1c), across experiments 1 to 3 the belt type anomaly was correctly classified in 95.9% of cases, the tray type anomaly was correctly classified in 91.5% of cases and the gantry type anomaly was correctly classified in 79.9% of cases. The worse performance on gantry type anomalies is potentially due to the small effect that gantry velocity modifications have on

the signal frequencies of the system.[1]

5.2 Research question 2

RQ 2 concerns the anomaly detection performance of the proposed Self-Attention Autoencoder architecture evaluated on the MGR dataset and generalized on synthetic datasets. Performance on the MGR dataset (RQ 2a) in Experiment 4 is similar to the performance of the Transformer architecture, which is promising given the additional methodological advantages of the Self-Attention Autoencoder (refer to Section 2.3).

RQ 2b to 2d were investigated together in Experiments 5 to 7. Performance trends were comparable across mathematical, domain-related and mixed synthetic data. As previously noted, the test-calibration ratio does not generally increase monotonically with increasing number of anomalous dimensions. This might seem strange; however, recall that the investigated architecture is an encoder-decoder model. It compresses input data into a reduced representation, meaning input dimensions are not fully conserved in the bottleneck. The bottleneck represents a low-dimensional abstraction over the high-dimensional input dimensions. As input dimensions are abstracted through fully connected neural network layers, they interact and every single signal affects the representation of all other signals. Thus, adding an additional anomalous dimension can sometimes decrease the test-calibration ratio through improving the reconstruction of other anomalous dimensions. While this phenomenon does not decrease test-calibration ratios below acceptable rates for the data investigated in this thesis, it is an important observation to keep in mind during research into anomaly detection using reconstruction approaches. Moving on from this point, it should be noted that there were significant differences in test-calibration ratios depending on anomaly class. Point and group anomalies showed both high and mainly increasing test-calibration ratios across all data types. However, contextual point anomalies exhibit a notable leveling of test-calibration ratios after approximately 50 dimensions containing anomalous signals. Furthermore, note the remarkable trend in decreasing test-calibration ratios after approximately 50 anomalous dimensions for the contextual group anomaly (especially visible on the mathematical data). In contextual group anomaly scenarios the anomaly corresponds to a doubling of a signal's frequency. The trend of decreasing test-calibration ratios can thus be explained if one assumes that the model learns to encode signals by abstracting their frequencies with regards to one another. As the number of anomalous dimensions increases from 0 to 50, the model encounters an increasing number of deviating frequencies compared to what it encountered in the training data. However, as the number of anomalous dimensions surpasses 50, more than half of the signals contain doubled frequencies. Thereafter, additional anomalous doubled frequencies simplify reconstruction because the data to reconstruct becomes overall more uniform in terms of frequency.[2] Another point of interest is the step-like pattern observed for the test-calibration ratio of the domain-related data. This is due to the method of adding new anomalous dimensions, which involves repeatedly alternating between the type of signals in the data. Certain types of signals have a stronger effect on anomaly

[1] Since gantry operation incorporates significant pauses, the total time taken is not severely affected by modifying the velocity of movements.

[2] In contextual point anomaly scenarios the anomaly corresponds to a break in a signal's frequency. Once again assuming that the model learns to encode signals by abstracting their frequencies with regards to one another, the trend of leveling anomaly scores for contextual point anomalies can be reasoned. As the number of anomalous dimensions surpasses 50, more than half of the signals contain broken frequencies. At this point the model likely reconstructs a similar deviation for all signals regardless of their being anomalous or not. Thus, additional anomalous broken frequencies tend not to increase the reconstruction error.

scores. When these are made anomalous the test-calibration ratio increases significantly, showing up as a step in the plot.

RQ 2e regarding the indicativeness of the training performance for later test performance was investigated in Experiment 8. From the results obtained it can be concluded that training performance is indeed a good indicator for test performance. This is important for practical application because test performance cannot be predicted easily since anomalies that might be encountered during the operation of an industrial system are usually not fully known in advance.

RQ 2f examines the effect of noise on architecture performance and was investigated in Experiment 9. As increasing amounts of noise are added to the calibration and test data, architecture performance decreases. Further note that the architecture becomes strongly unreliable from a noise standard deviation of approximately 20% of the value-range of the original signal onwards. It should be emphasized at this point that the decrease in standard deviation of the test-calibration ratio (the light blue shaded area in Figure A.20) over increasing standard deviation of the noise (the x-axis) is a natural consequence of the metric chosen. It does *not* mean that the architecture becomes more reliable as more noise is added. It simply means that anomaly scores on both calibration and test data increase to large and similar values since normal and anomalous data becomes indistinguishable to the architecture as more noise is added (refer to Figure A.19 for a visual example).

5.3 Research question 3

RQ 3 concerns the anomaly diagnosis performance of the Transformer and Self-Attention Autoencoder using the methodology proposed in this thesis. For the Transformer model (RQ 3a) the anomalous observer contribution ratio is 0.195 across data types.[3] This means that only about 20% of anomaly score contributions originate from actually anomalous dimensions. In order to explain this, recall that the model abstracts input dimensions through fully connected neural network layers. Thus, anomalous dimensions also influence the reconstruction of non-anomalous dimensions. As a result, high test-calibration ratios sufficient for anomaly detection are still obtained with low anomalous observer contribution ratios. Note that the signal with maximal anomaly score contribution always corresponded to an actual anomalous signal, validating the signal identification method of the proposed anomaly diagnosis methodology. Regarding the anomaly attention ratio however, a poor mean value of 0.179 across data types is observed for the Transformer model. This means that the Transformer focuses its attention on timesteps corresponding to the anomaly in only 17.9% of cases. This suggests that the attention matrices resulting from the Transformer model are mainly un-informative and could not be efficiently used for identifying the timestep-location of anomalies.

For the Self-Attention Autoencoder model (RQ 3b) the anomalous observer contribution ratio is 0.196 making it almost identical to the Transformer in this metric. Equally, the signal with maximal anomaly score contribution for the Self-Attention Autoencoder also always corresponded to an actual anomalous signal. In contrast to the Transformer however, the anomaly attention ratio of the Self-Attention Autoencoder is significantly higher at a mean of 0.559 across data types. This means that the Self-Attention Autoencoder focuses its attention on timesteps corresponding to the anomaly in 55.9% of cases. The attention matrices from this architecture are thus informative and can be used for identifying the timestep-location of anomalies. The proposed anomaly diagnosis methodology in combination

[3]The expected anomalous observer contribution ratio for a random model for this experiment would be 0.1.

with the Self-Attention Autoencoder thus provides valuable information towards the causes of anomalies.

5.4 Limitations and further research

While this work presents promising results, various important limitations should be discussed. First, the steep increase in true positive rates at around 0.5 false positive rate will be considered. This increase was observed for every architecture type across both mean and standard deviation threshold types, which suggests that it results from structure in the data. When looking into the anomaly scores of normal MGR test data as well as those of the gantry anomaly type, two observations were made which likely explain this pattern. The first observation is that anomaly scores of normal test data can be located on either of two levels, with one being on average about 20% higher than the other.[4] This is likely due to the data recording, where a high workload of the computer leads to slightly more noisy signal data from the simulation. This occurs for about half of the time-windows, leading to the slightly higher level of anomaly scores for these time-windows. Models such as the LSTM Autoencoder, Transformer and Self-Attention Autoencoder were still able to distinguish most normal and anomalous time-windows. However, the gantry anomaly type resulted in overall low anomaly scores compared to other anomalies.[5] The second observation made is that these particularly low gantry anomaly scores are often located between the two levels of normal anomaly scores. As the threshold is decreased and starts to include these gantry anomalies, the lower level of the normal anomaly scores is not yet reached. Thus an increase in true positive rates can be observed without a significant increase in false positive rates, causing the steep incline noticed in the ROC plots. More sophisticated anomaly score evaluation methods would likely have to be investigated in order to reliably differentiate gantry anomaly scores from both levels of normal anomaly scores.

Secondly, while efforts were made to generalize the performance of the architecture and methodology proposed in this thesis, these do not relieve future practitioners from adapting the model to their specific problem-domain. Noteworthy in particular is the size of the bottleneck, which defines the extent of data compression. The ideal value of this parameter will vary across systems and will have to be adapted. Future work could thus investigate the influence of the bottleneck size on the representational capabilities of the model. Of particular interest would be methods for the algorithmic determination of ideal bottleneck sizes. This matter has been investigated for Autoencoders in terms of textual representations by Gupta et al. [45] and in terms of traffic forecasting by Boquet et al. [46] but has, to the best of the author's knowledge, not yet been investigated for reconstruction models in the field of anomaly detection and diagnosis.

On a different point, the accurate classification of normal data for the MGR dataset varied considerably across different cross-validations for all investigated architectures and threshold types.[6] This presents an issue in terms of the reliability of the models on normal MGR data across different training runs. The high variance is presumably due to the comparatively low number of nominal datapoints recorded.[7] By obtaining a larger number of samples and running a higher number of cross-validation

[4] Both levels still show a large variance, but their means are different by about 20%.

[5] This is likely due to the aforementioned small influence of gantry velocity modifications on the signal frequencies of the system.

[6] With the previously discussed exception of the Transformer model under Series threshold type.

[7] A total of 6 315 nominal datapoints were recorded, which were split $70/20/10\%$ into training, calibration and testing datasets.

folds the variance could likely be lowered.

Moving on from this point, it should be noted that the methods developed and evaluated within this thesis are set within the semi-supervised learning task, i.e. the training objective is solely informed by normal datapoints. This setting was adopted since it does not require labeled anomalies, making it applicable across industrial systems. However, in certain systems labeled datapoints for specific anomalies might nonetheless be available. In the case of one-class classification methods it was shown by Görnitz et al. [47] that the incorporation of labeled anomalies can improve anomaly detection performance significantly. Thus, it would be of great interest for further research to extend the training objective of reconstruction models to include labeled anomalies. A simple approach would be to maximize the reconstruction error on anomalous datapoints. To make this approach viable however, the reconstruction error would likely have to be bounded. Furthermore, a method of balancing between the objectives of minimizing error on normal datapoints and maximizing error on anomalous datapoints would have to be developed.

Chapter 6

Conclusions

Anomaly detection and diagnosis in industrial systems is a critical component for the optimization of performance. This thesis investigated Undercomplete Autoencoders, LSTM Autoencoders and Transformers for their performance in anomaly detection on digital twin data. Out of these architectures, the Transformer model performed best in terms of AUC and F1-score. Additionally, an architecture termed Self-Attention Autoencoder was developed, which performed on a similar level to the Transformer in terms of anomaly detection performance. In contrast to the Transformer, its design includes a controllable bottleneck and interpretable attention matrices, both desirable features for industrial anomaly detection and diagnosis. The Self-Attention Autoencoder's capabilities were generalized to different types of data and anomalies through evaluation on synthetic datasets. Furthermore, a methodology for the diagnosis of anomalies was proposed, which identifies the signal and timestep in which a detected anomaly is likely to be located. While the Transformer architecture does not perform favorably, the Self-Attention Autoencoder showed good performance using this methodology.

For practitioners in the field of industrial anomaly detection this thesis offers the following recommendations. First, it is advised to carefully consider the costs associated with missed anomalies as well as false alarms. If the latter is particularly expensive, a Transformer Encoder-Decoder in combination with a rule-based threshold type, such as the Series threshold evaluated in this thesis, offers very low false positive rates. If an approach with a more balanced trade-off is desired, the Self-Attention Autoencoder developed in this thesis offers overall good anomaly detection performance with the additional advantage of being easily controllable and more interpretable. In particular, it is recommended to implement anomaly diagnosis methodologies such as the one proposed in this thesis, as these have been demonstrated to provide valuable information towards the spatial and temporal location of anomalies and have the potential of greatly speeding up the efforts of engineers in resolving said anomalies.

Bibliography

[1] Ray Y. Zhong, Xun Xu, Eberhard Klotz, and Stephen T. Newman. Intelligent manufacturing in the context of industry 4.0: A review. *Engineering*, 3(5):616–630, 2017.

[2] Shen Yin, Steven X. Ding, Xiaochen Xie, and Hao Luo. A review on basic data-driven approaches for industrial process monitoring. *IEEE Transactions on Industrial Electronics*, 61(11):6418–6428, 2014.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009.

[4] Clauber Gomes Bezerra, Bruno Sielly Jales Costa, Luiz Affonso Guedes, and Plamen Parvanov Angelov. A comparative study of autonomous learning outlier detection methods applied to fault detection. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7, 2015.

[5] Leonardo Barbini, Carmen Bratosin, and Emile van Gerwen. Model based diagnosis in complex industrial systems: a methodology. *PHM Society European Conference*, 5(1), 2020.

[6] Steven Ding. *Data-driven design of fault diagnosis and fault-tolerant systems*. Springer, London, England, 2014.

[7] Machinaide: Innovative concepts for accessing, searching, analysing and using multiple digital twins. https://www.machinaide.eu/about.html. [Accessed Jul. 14, 2021].

[8] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, 8, 2020.

[9] Lukas Ruff, Jacob Kauffmann, Robert Vandermeulen, Gregoire Montavon, Wojciech Samek, Marius Kloft, Thomas Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109:1–40, 02 2021.

[10] Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, 8(3-4):237–253, 2000.

[11] Manqi Zhao and Venkatesh Saligrama. Anomaly detection with score functions based on nearest neighbor graphs. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, NIPS'09, pages 2250–2258, Red Hook, NY, USA, 2009. Curran Associates Inc.

[12] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[13] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.

[14] Wolfgang Hardle. *Applied nonparametric regression*. Cambridge University Press, Cambridge England New York, 1990.

[15] William Press. *Numerical recipes: the art of scientific computing*. Cambridge University Press, Cambridge, UK New York, 2007.

[16] Peter Rousseeuw. *Robust regression and outlier detection*. Wiley-Interscience, Hoboken, NJ, 2003.

[17] Alberto Munoz and Javier M. Moguerza. Estimation of high-density regions using one-class neighbor machines. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3), March 2006.

[18] Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander Smola, and Robert Williamson. Estimating support of a high-dimensional distribution. *Neural Computation*, 13, 07 2001.

[19] David Tax and Robert Duin. Support vector data description. *Machine Learning*, 54, 01 2004.

[20] Shehroz S. Khan and Michael G. Madden. A survey of recent trends in one class classification. In *Artificial Intelligence and Cognitive Science*, pages 188–197, Berlin, Heidelberg, 2010. Springer.

[21] Ian Jolliffe. *Principal component analysis*. Springer, New York, 2002.

[22] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw.*, 2(1):53–58, January 1989.

[23] Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, New York, NY, USA, 2017. Association for Computing Machinery.

[24] Chao Wang, Bailing Wang, Hongri Liu, and Haikuo Qu. Anomaly detection for industrial control system based on autoencoder neural network. *Wireless Communications and Mobile Computing*, 2020:1–10, 08 2020.

[25] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

[26] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 12 1997.

[27] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. LSTM-based encoder-decoder for multi-sensor anomaly detection. *CoRR*, abs/1607.00148, 2016.

[28] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54:1–38, 03 2021.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[30] Hengyu Meng, Zhang Yuxuan, Yuan-xiang Li, and Honghua Zhao. *Spacecraft Anomaly Detection via Transformer Reconstruction Error*, pages 351–362. Proceedings of the International Conference on Aerospace System Science and Engineering, 01 2019.

[31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, Cambridge, Massachusetts, 2016.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[33] Nathalie Japkowicz, Catherine Myers, and Mark Gluck. A novelty detection approach to classification. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 518–523, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[34] Muhammad Ayaz Hussain, Muhammad Saif-ur Rehman, Christian Klaes, and Ioannis Iossifidis. Comparison of anomaly detection between statistical method and undercomplete autoencoder. In *Proceedings of the 2020 5th International Conference on Big Data and Computing*, ICBDC 2020, pages 32–38, New York, NY, USA, 2020. Association for Computing Machinery.

[35] M. Tanjid Hasan Tonmoy, Saif Mahmud, A. K. M. Mahbubur Rahman, M. Ashraful Amin, and Amin Ahsan Ali. Hierarchical self attention based autoencoder for open-set human activity recognition. In *Advances in Knowledge Discovery and Data Mining*, pages 351–363. Springer International Publishing, 2021.

[36] Minghua Zhang and Yunfang Wu. An unsupervised model with attention autoencoders for question retrieval. In *AAAI*, 2018.

[37] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[38] Douglas Montgomery. *Introduction to statistical quality control*. John Wiley, Hoboken, N.J, 2005.

[39] John Proakis. *Digital signal processing : principles, algorithms, and applications*. Prentice Hall, Upper Saddle River, N.J, 1996.

[40] Claude Sammut and Geoffrey I. Webb, editors. *Encyclopedia of Machine Learning*. Springer US, 2010.

[41] Peter Flach. *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2017.

[42] Angelos Angelopoulos, Emmanouel T. Michailidis, Nikolaos Nomikos, Panagiotis Trakadas, Antonis Hatziefremidis, Stamatis Voliotis, and Theodore Zahariadis. Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. *Sensors*, 20(1):109, December 2019.

[43] Brian W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, October 1975.

[44] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[45] Parth Gupta, Rafael E. Banchs, and Paolo Rosso. Squeezing bottlenecks: Exploring the limits of autoencoder semantic representation capabilities. *Neurocomputing*, 175:1001–1008, 2016.

[46] Guillem Boquet, Edwar Macias, Antoni Morell, Javier Serrano, and Jose Lopez Vicario. Theoretical tuning of the autoencoder bottleneck layer dimension: A mutual information-based algorithm. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1512–1516, 2021.

[47] Nico Goernitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262, Feb 2013.

Appendix A

Supplementary Material

A.1 Formal Data Analysis

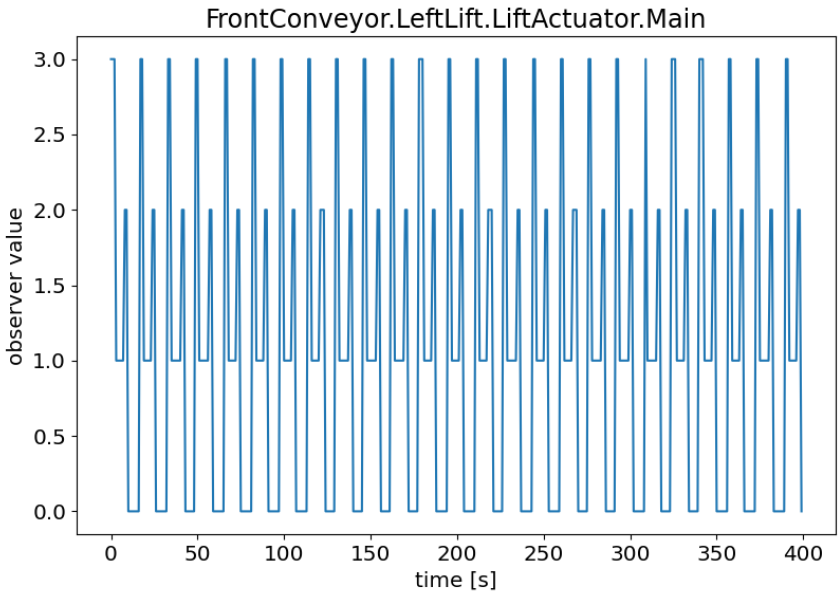


Figure A.1: Example of a categorical observer with four states.

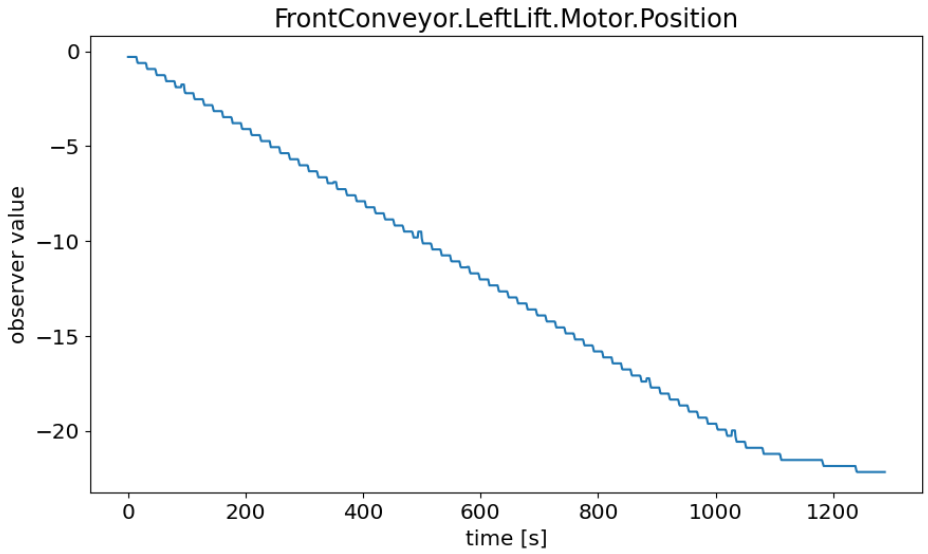


Figure A.2: Example of a numerical observer with large value range. These observers are all related to the conveyor belt positions.

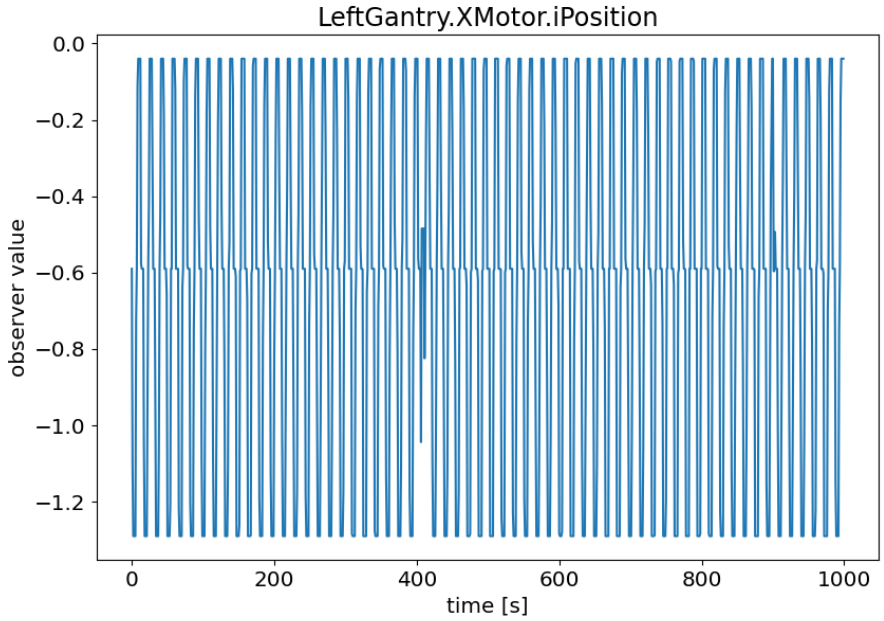


Figure A.3: Example of a numerical observer with a small value range. These observers are related to machineparts with periodic behavior.

A.2 Synthetic Dataset

A.2.1 Mathematical signals

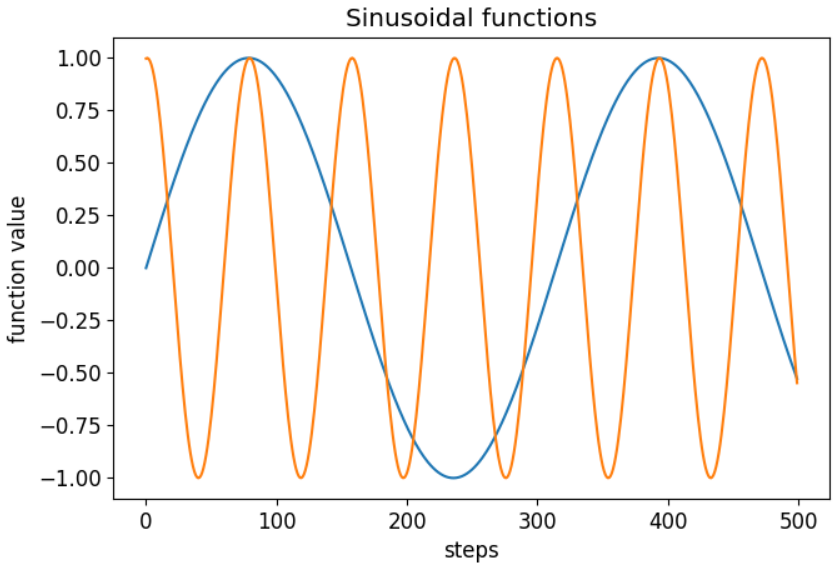


Figure A.4: Two example sinusoidal functions from the synthetic, mathematical dataset.

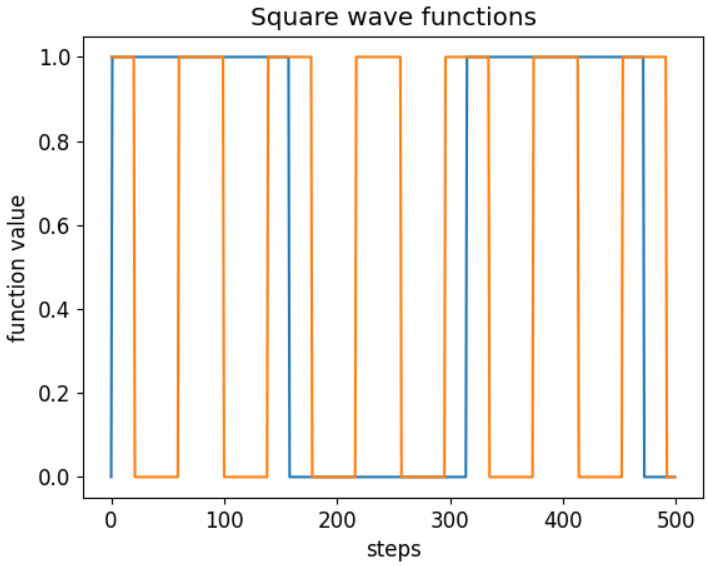


Figure A.5: Two example square wave functions from the synthetic, mathematical dataset.

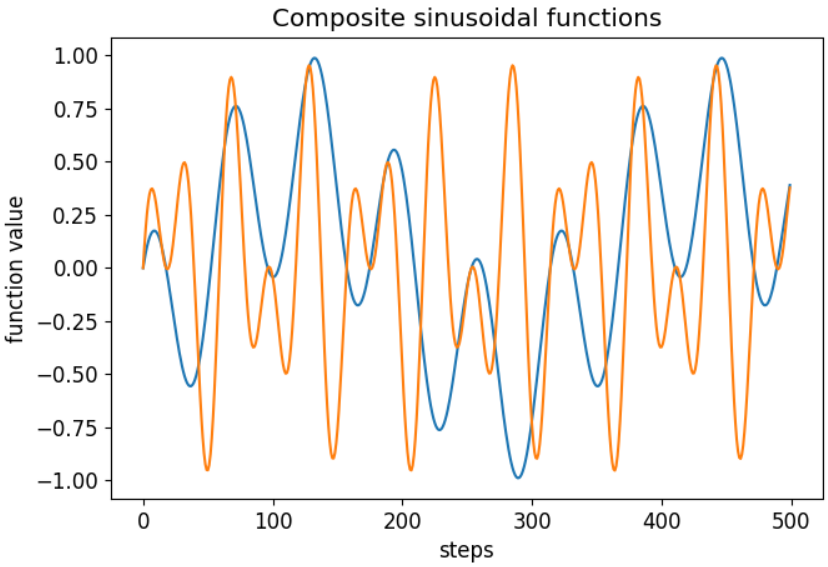


Figure A.6: Two example composite sinusoidal functions from the synthetic, mathematical dataset.

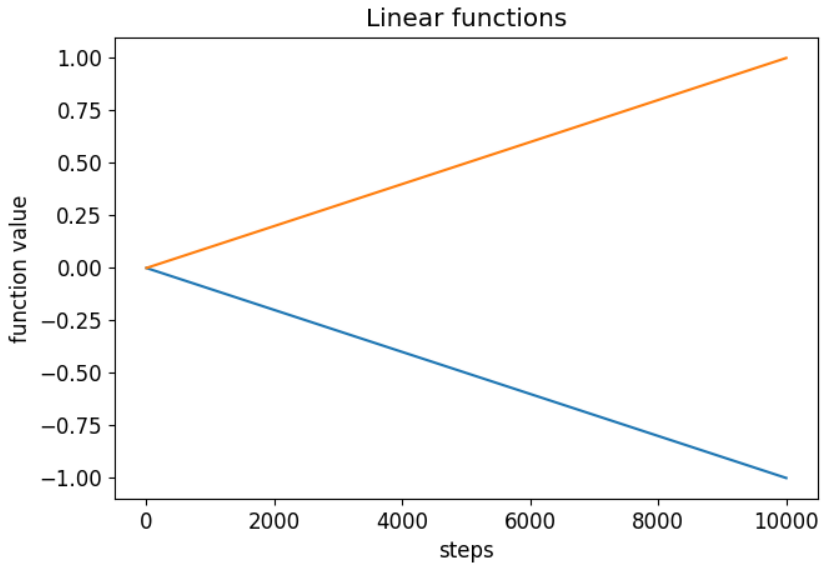


Figure A.7: The two linear functions from the synthetic, mathematical dataset. Note that only two linear functions were added to the dataset as min-max normalization equalizes linear functions of any slope and offset (except for positive or negative slope).

A.2.2 Domain-inspired signals

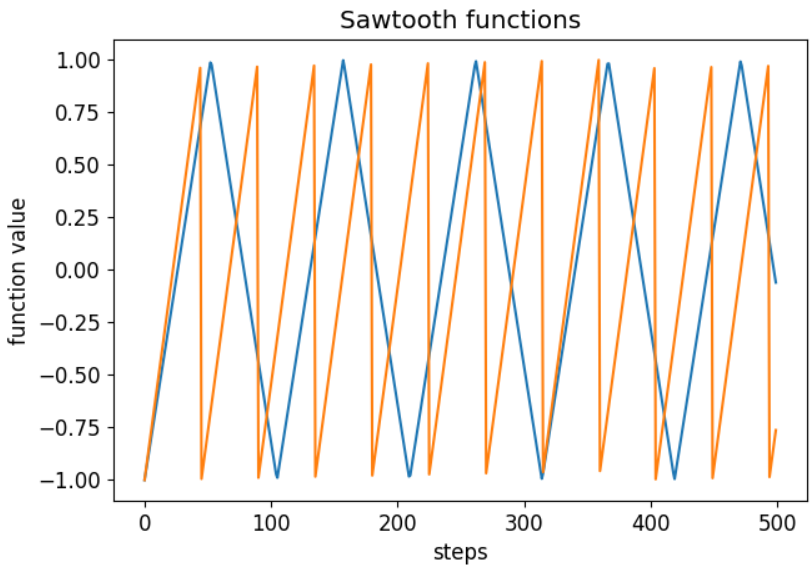


Figure A.8: Two example sawtooth functions from the synthetic, domain-inspired dataset.

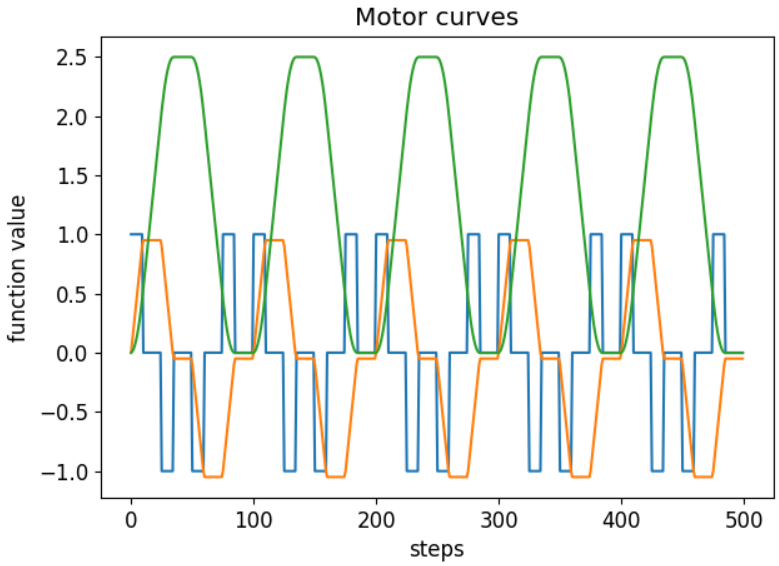


Figure A.9: Three motor curves from the synthetic, domain-inspired dataset. The functions shown here correspond to speed, acceleration and jerk signals are modeled after a motor that periodically speeds up and slows down.

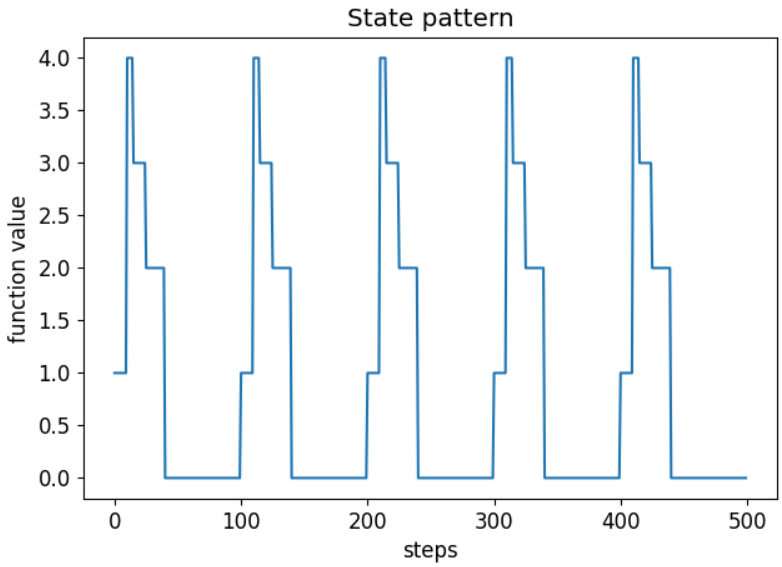


Figure A.10: An example state pattern from the synthetic, domain-inspired dataset.

A.2.3 Anomaly signals

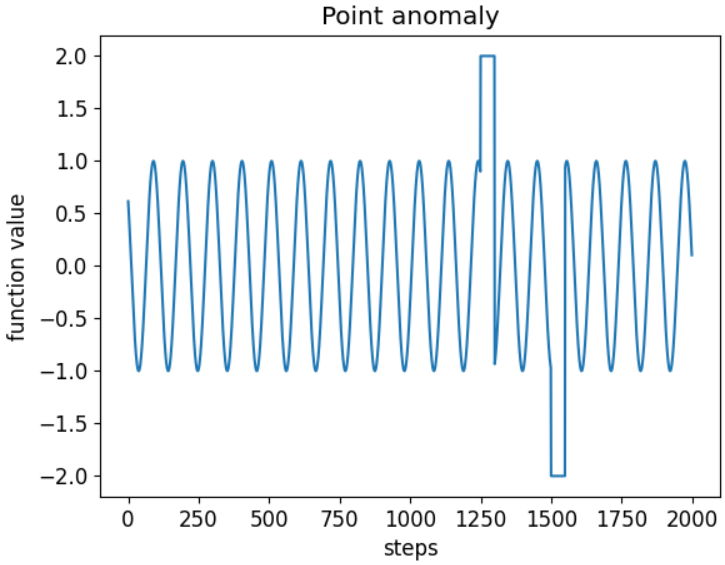


Figure A.11: A point anomaly of a sinusoidal function within the synthetic, mathematical dataset. Note that the function values during the anomaly are outside of the nominal range of the function.

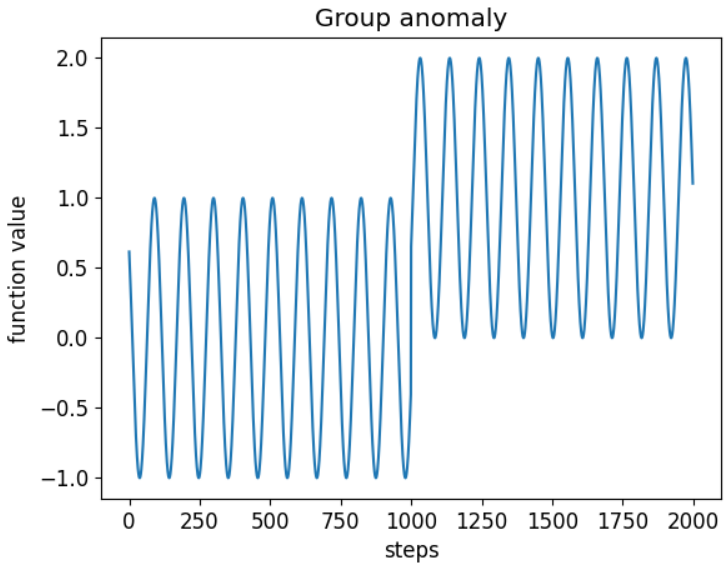


Figure A.12: A group anomaly of a sinusoidal function within the synthetic, mathematical dataset. Note that the function values during the anomaly are outside of the nominal range of the function.

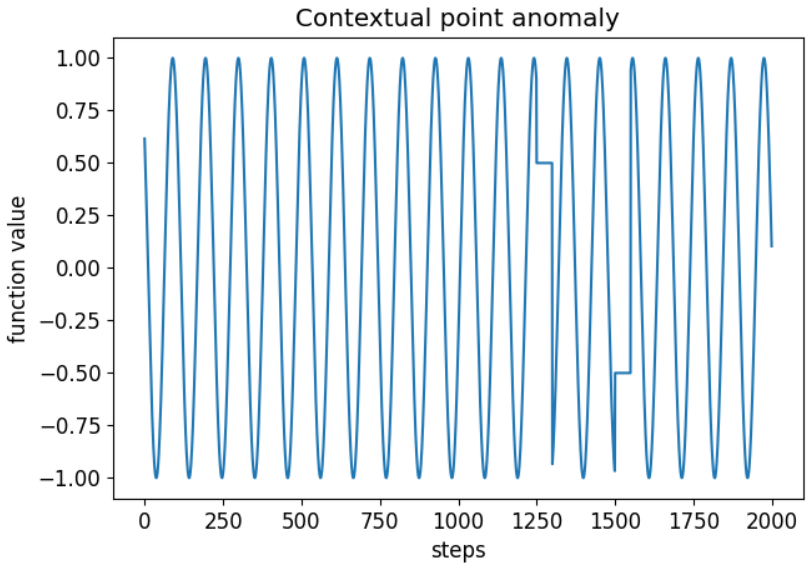


Figure A.13: A contextual point anomaly of a sinusoidal function within the synthetic, mathematical dataset. Note that the function values during the anomaly are within the nominal range of the function. The anomaly is only apparent through contextual time-series information.

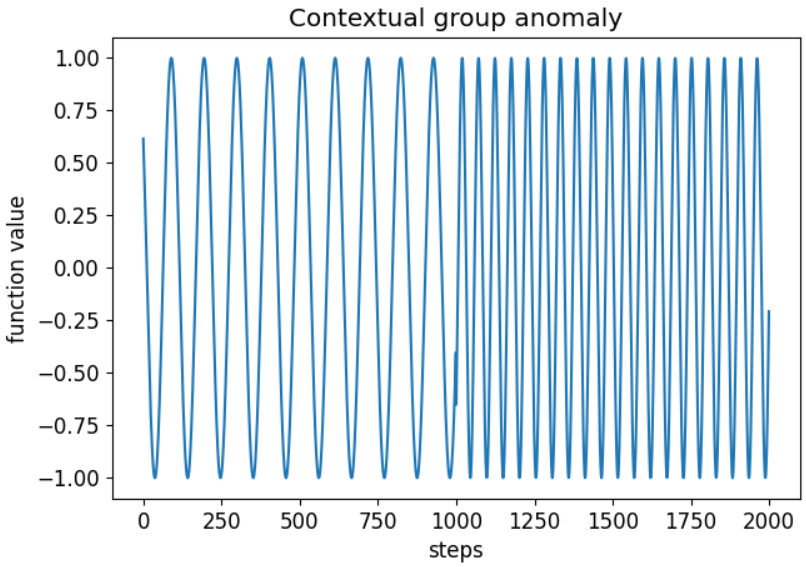


Figure A.14: A contextual group anomaly of a sinusoidal function within the synthetic, mathematical dataset. Note that the function values during the anomaly are within the nominal range of the function. The anomaly is only apparent through contextual time-series information.

A.3 Architecture Specifications

Layer	Dimensionality/Neurons	Activation Function
0 (Input)	87	-
1	20	Sigmoid
2	10	Sigmoid
3	20	Sigmoid
4	87	-

Table A.1: Undercomplete Autoencoder architecture used in experiment 1. Architecture definitions for experiment 1 to 4 were chosen for similar bottleneck, thus similar encoding sizes.

Layer	Dimensionality/Cells	Type
0	87	Input
1	10	LSTM
2	10	Bottleneck: last hidden LSTM-state
3	10	LSTM
4	87	Linear

Table A.2: LSTM Autoencoder architecture used in experiment 2. Architecture definitions for experiment 1 to 4 were chosen for similar bottleneck, thus similar encoding sizes. For more information on the exact working of the LSTM Autoencoder architecture used refer to [27].

Number of Heads	1
Number Encoder Layers	1
Number Decoder Layers	1
Encoding Dimension (Bottleneck)	10
Activation Function	ReLU

Table A.3: Architecture parameters of the Transformer architecture used in experiment 3. Architecture definitions for experiment 1 to 4 were chosen for similar bottleneck, thus similar encoding sizes. For more information on what parameters refer to see [29].

Layer	**Dimensionality/Neurons**	**Type**
0a	87	Input
0b	87	Positional Encoding
1	87	Scaled Dot-Product Attention
2	87	Layer Normalization
3	10	Linear
4	87	Linear

Table A.4: Self-Attention Autoencoder architecture used in experiment 4. Architecture definitions for experiment 1 to 4 were chosen for similar bottleneck, thus similar encoding sizes. For more information refer to Section 2.3.

Layer	**Dimensionality/Neurons**	**Type**
0a	100	Input
0b	100	Positional Encoding
1	100	Scaled Dot-Product Attention
2	100	Layer Normalization
3	20	Linear
4	100	Linear

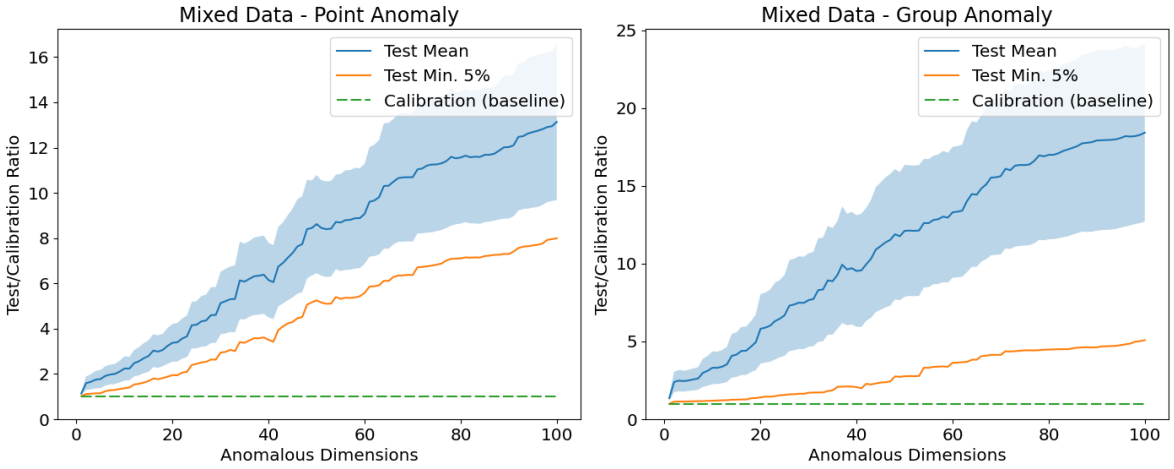
Table A.5: Self-Attention Autoencoder architecture used in experiments 5,6,7,8,9 and 11. Architecture definitions for experiment 5 to 11 were chosen for similar bottleneck, thus similar encoding sizes. For more information refer to Section 2.3.

Number of Heads	1
Number Encoder Layers	1
Number Decoder Layers	1
Encoding Dimension (Bottleneck)	20
Activation Function	ReLU

Table A.6: Architecture parameters of the Transformer architecture used in experiment 10. Architecture definitions for experiment 5 to 11 were chosen for similar bottleneck, thus similar encoding sizes. For more information on what parameters refer to see [29].

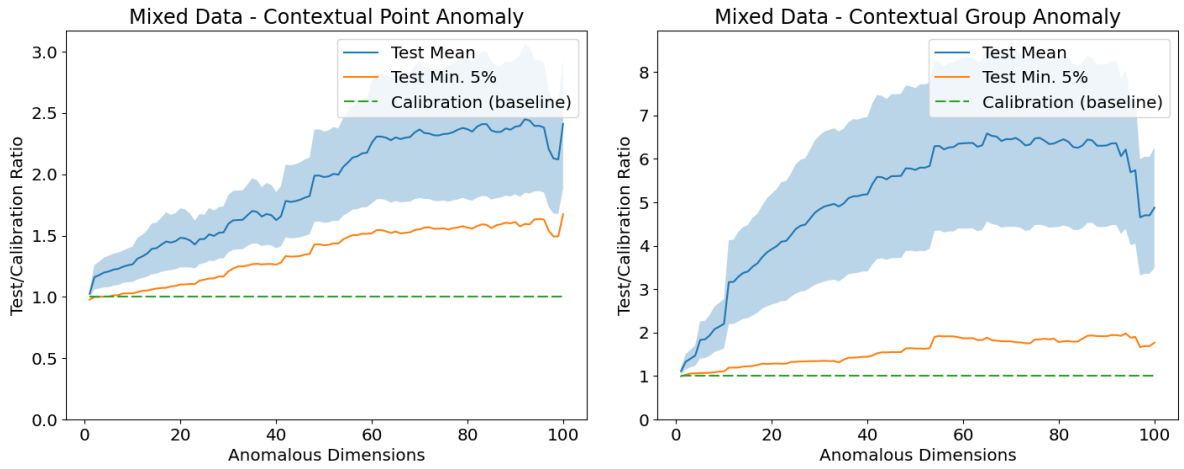
A.4 Anomaly detection performance of the Self-Attention Autoencoder

A.4.1 Architecture performance on the mixed synthetic dataset under varying anomalous dimensions



(a) (b)

Figure A.15: Architecture performance plots for the Self-Attention Autoencoder model under varying number of anomalous dimensions. Plots resulting from evaluation of 5 cross-validations on the synthetic mixed dataset. Mean test-calibration ratio shown in dark blue; ± 1 standard deviation shaded in light blue; minimum 5% quantile of the test-calibration ratio shown in orange. (a) Architecture performance plot for the point type anomaly. (b) Architecture performance plot for the group type anomaly.



(a) (b)

Figure A.16: Architecture performance plots for the Self-Attention Autoencoder model under varying number of anomalous dimensions. Plots resulting from evaluation of 5 cross-validations on the synthetic mixed dataset. Mean test-calibration ratio shown in dark blue; ± 1 standard deviation shaded in light blue; minimum 5% quantile of the test-calibration ratio shown in orange. (a) Architecture performance plot for the contextual point type anomaly. (b) Architecture performance plot for the contextual group type anomaly.

A.4.2 Training performance indicativeness

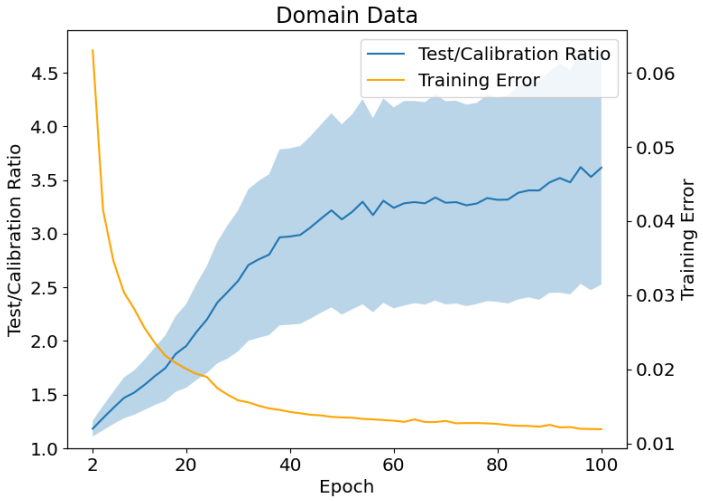


Figure A.17: Test-calibration ratio and training reconstruction error for varying number of epochs of training on the synthetic domain-related data. Note that the number of anomalous dimensions was set to 10 for this experiment. Further note that the test/calibration ratio is averaged over all types of anomalies.

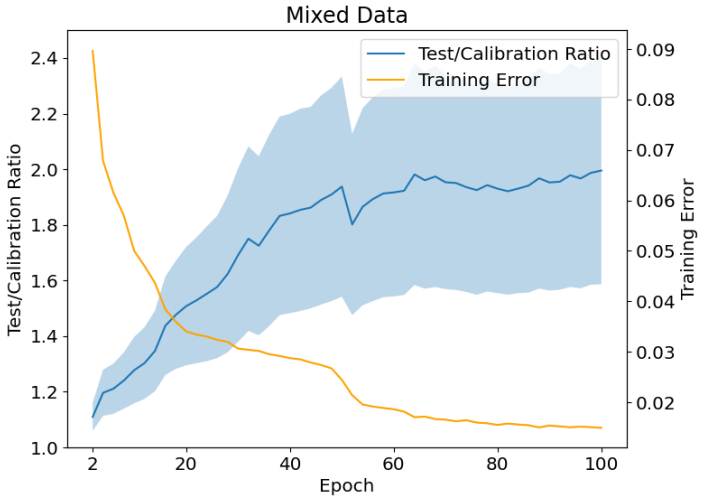
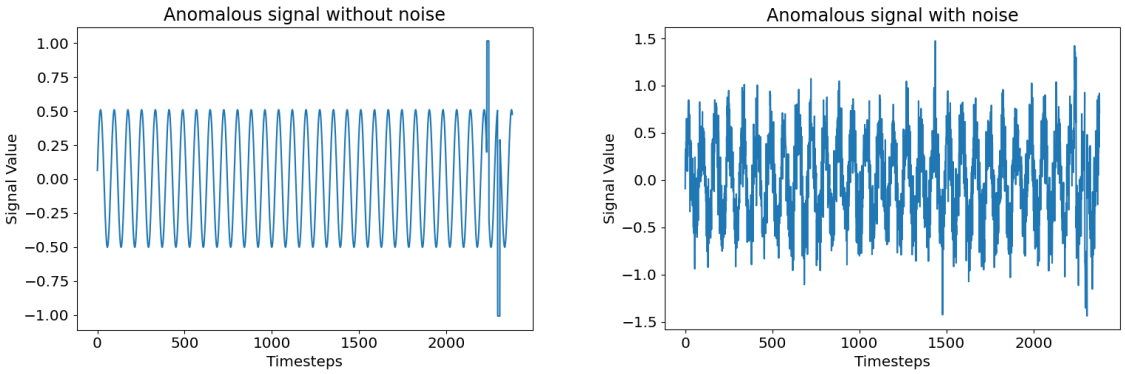


Figure A.18: Test-calibration ratio and training reconstruction error for varying number of epochs of training on the synthetic mixed data. Note that the number of anomalous dimensions was set to 10 for this experiment. Further note that the test/calibration ratio is averaged over all types of anomalies.

A.4.3 Robustness to Noise



(a) (b)

Figure A.19: Example anomalous signal from the synthetic mathematical dataset with and without noise. Note that the anomaly is located in the last 200 timesteps. (a) Example signal without noise. (b) Example signal with normally distributed noise added. Noise standard deviation equals 25% of the value-range of the original signal (this is the maximum amount of noise evaluated in Section 4.2.6).

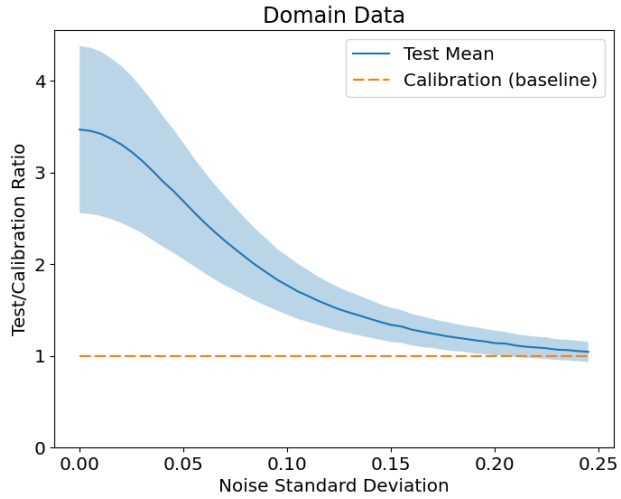


Figure A.20: Test-calibration ratio on the synthetic domain-related data under varying amounts of noise added to test and calibration data. Noise standard deviation is given in % of the value-range of the original signals.

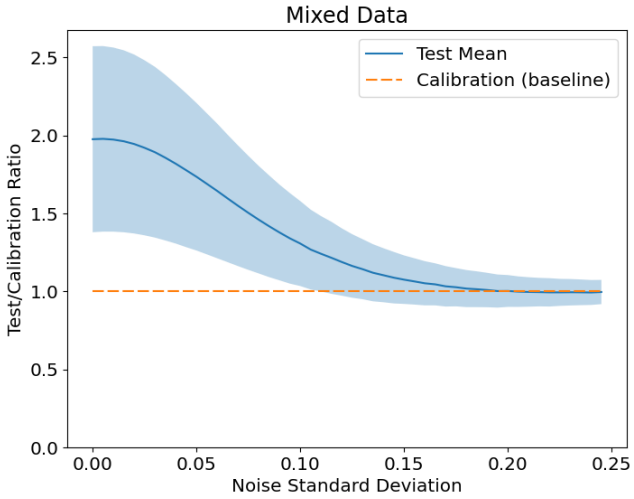


Figure A.21: Test-calibration ratio on the synthetic mixed data under varying amounts of noise added to test and calibration data. Noise standard deviation is given in % of the value-range of the original signals.