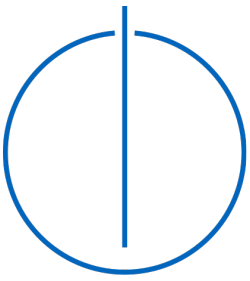
SIMON KLOTZ

MASTER'S THESIS IN DATA ENGINEERING AND ANALYTICS

SEQUENTIAL LATENT VARIABLE MODELS FOR PROGNOSTICS AND HEALTH MANAGEMENT

Technical University of Munich
Department of Informatics



SEQUENTIAL LATENT VARIABLE MODELS FOR PROGNOSTICS AND HEALTH MANAGEMENT

SEQUENTIELLE LATENTE VARIABLENMODELLE FÜR PROGNOSTIK UND GESUNDHEITSMANAGEMENT

SIMON KLOTZ

MASTER'S THESIS IN DATA ENGINEERING AND ANALYTICS

Technical University of Munich
Department of Informatics

Simon Klotz: *Sequential Latent Variable Models for Prognostics and Health Management*

SUPERVISOR:
PD Dr. Georg Groh

ADVISORS:
Maximilian Soelch, M.Sc. & Dr. Justin Bayer

SUBMISSION DATE:
November 13, 2019

DECLARATION

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, November 13, 2019

Simon Klotz

SEQUENTIAL LATENT VARIABLE MODELS FOR PROGNOSTICS AND HEALTH MANAGEMENT

SIMON KLOTZ

ABSTRACT

In this work, we propose a novel method using sequential latent variable models for the two Prognostics and Health Management tasks of anomaly detection and prognostics. The approach is applicable to high-dimensional time series data and neither requires domain knowledge nor is it bound to a specific sequential latent variable model. It is evaluated on a dataset containing bearing sensor data for anomaly detection and on the commonly used CMAPSS dataset to demonstrate that it is also suited for prognostics. Our results prove that a universal approach for both anomaly detection and prognostics is feasible.

SEQUENTIELLE LATENTE VARIABLENMODELLE FÜR PROGNOSTIK UND GESUNDHEITSMANAGEMENT

SIMON KLOTZ

ZUSAMMENFASSUNG

In dieser Thesis entwickeln wir eine neuartige Methode für Anomaliedetektion und Prognostik unter Verwendung von sequentiellen latenten Variablenmodellen. Der Ansatz ist auf mehrdimensionale Zeitreihendaten anwendbar und erfordert weder domänenspezifisches Fachwissen, noch ist er an ein bestimmtes sequentielles Modell mit latenten Variablen gebunden. Der Algorithmus wird auf einem Datensatz mit Kugellagersensordaten zur Anomaliedetektion angewandt und anschließend zeigen wir auf dem gängigen CMAPSS-Datensatz, dass er auch für die Prognostik geeignet ist.

CONTENTS

NOTATION

x	scalar
$\boldsymbol{x}$	vector
$\boldsymbol{X}$	matrix
n	sample index, $n \in \{1, ..., N\}$
t	time index, $t \in \{1, ..., T\}$
d	dimension index, $d \in \{1, ..., D\}$

ACRONYMS

NN	Neural Network
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
GAN	Generative Adversarial Network
EncDec	Encoder-Decoder
VAE	Variational Autoencoder
STORN	Stochastic Recurrent Network
SSM	State-Space Model
DVBF	Deep Variational Bayes Filter
SVM	Support Vector Machine
KNN	K-Nearest Neighbors
PCA	Principal Component Analysis
RUL	Remaining Useful Life
PHM	Prognostics and Health Management
CMAPSS	Commercial Modular Aero-Propulsion System Simulation
EDM	Electric Discharge Machining

ELBO Evidence Lower Bound

MSE Mean Squared Error

RMSE Root Mean Squared Error

ROC Receiver Operating Characteristic

ROC-AUC Area Under the Receiver Operating Characteristic Curve

TP True Positives

TN True Negatives

FP False Positives

FN False Negatives

TPR True Positive Rate

FPR False Positive Rate

1 INTRODUCTION

Prognostics and Health Management (PHM) has been extensively studied by researchers in engineering-related fields to improve the availability of industrial systems. It provides a framework for several tasks including anomaly detection and prognostics [58]. The main objective of PHM is to increase a system's reliability and safety in an automated way to decrease maintenance costs and prevent total system failures. As it is mainly focused on mechanical components, most approaches rely on time series data recorded from available sensors.

Anomaly detection is the task of detecting previously unseen patterns that do not match the expected behavior of a system [9]. It can be applied to detect anomalies as diverse as credit card fraud, intrusions in a cyber-physical system, or faults in mechanical components [9]. Detecting anomalous behavior as early as possible is of critical importance as it can prevent a total system failure by allowing effective countermeasures to be deployed in time.

However, there are several difficulties when it comes to developing an anomaly detection system. While it is comparably easy to record data from a healthy system, it is hard to obtain from a failing one as components can be expensive or have a long lifespan. Also, many systems are not properly equipped to record relevant sensor data [52]. Additionally, defining normality is not straightforward as it needs to encompass all potential normal behaviors and the boundary between normal and anomalous samples is often not precisely defined.

Prognostics has the objective of accurately estimating the Remaining Useful Life (RUL) of a system. RUL is defined as "remaining time before the system's health falls below a defined failure threshold" [25]. Common examples where prognostics is applied are medical equipment, aircraft engines, or power plants [56]. A prognostics system is expected to provide an early estimate of a components health to successfully prevent failure occurrences. This allows companies to save costs associated with maintenance operations and component failures. The field of prognostics faces similar obstacles as anomaly detection since run-to-failure time series are generally difficult to obtain and it might not be possible to operate a system until failure.

In the past few decades, researchers have developed a plethora of specialized algorithms for both anomaly detection [9] and prognostics [35] to deal with the aforementioned issues. In several cases, similar approaches are applied independently to the field of anomaly detection and to the field of prognostics (e.g. [42] and [20]). However, to the best of our knowledge, it has never been investigated whether it is

feasible to develop a general approach applicable to both PHM tasks. A universal approach would have the advantage that only one model has to be learned which can then perform both anomaly detection and prognostics.

Sequential latent variable models are often used to model the behavior of a system. They relate observable variables (e.g. the sensor recordings of a mechanical component) to latent variables that capture the inherent factors describing the behavior of a system. They have been successfully applied to both anomaly detection [61] and prognostics [69] and are thus a promising candidate for a universal PHM approach.

In this thesis, we investigate whether a single sequential latent variable model can be used for both anomaly detection and prognostics. The contributions of this work are fourfold. To build a general approach that can be applied to multiple PHM tasks, we first extend the anomaly detection method developed by Soelch et al. [61] to prognostics. Second, we show that sequential latent variable models can be used in novel ways using latent space representations to do anomaly detection. Then, we demonstrate that they are not constrained to Stochastic Recurrent Networks (STORNs) [4] but can be extended to other sequential latent variable models such as Deep Variational Bayes Filters (DVBFs) [27, 28]. Lastly, we apply our approach on a dataset for anomaly detection to verify its performance and subsequently on a dataset covering both anomaly detection and prognostics to empirically prove that a universal approach is feasible.

The remainder of this thesis is structured as follows. In Chapter 2 we provide the background on sequential latent variable models and variational inference. Chapter 3 gives an overview of the field of anomaly detection and prognostics and justifies why a general approach for multiple PHM tasks is feasible. In Chapter 4 the datasets used for our experiments are outlined and Chapter 5 introduces the novel methods used in this thesis and how they are evaluated. Finally, Chapter 6 presents the qualitative and quantitative results of the conducted experiments, leading to a conclusion in Chapter 7.

2 BACKGROUND

In this chapter, we introduce sequential latent variable models. We start by describing the concept of autoencoding and the Encoder-Decoder (EncDec) model in Section 2.1. Then, we outline variational inference in Section 2.2 which forms the basis for STORN and DVBF described in Sections 2.3 and 2.4 respectively.

We assume basic knowledge on Neural Networks (NNs) and refer the reader to Goodfellow et al. [18] for an in-depth coverage of the topic.

2.1 AUTOENCODING

Autoencoding [7] can be used for tasks such as dimensionality reduction [22] and denoising [65]. It refers to the process of learning efficient encodings $\boldsymbol{z}$ from observations $\boldsymbol{x}$ in an unsupervised way. In most cases, these encodings are of lower dimensionality compared to $\boldsymbol{x}$.

An autoencoder consists of two NNs which share a common layer acting as a capacity bottleneck: The first NN, the so-called encoder, takes the input $\boldsymbol{x}$ and maps it to an encoding $\boldsymbol{z}$, which is also referred to as latent representation or code. The second NN, called decoder, takes $\boldsymbol{z}$ as input and outputs an approximate reconstruction $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$. The bottleneck layer is often smaller to prevent the autoencoder from learning the identity function. Autoencoders can be trained on a dataset $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}\}$ by minimizing a reconstruction error, for example, the Mean Squared Error (MSE), $\sum_{n=1}^{N} ||\boldsymbol{x}^{(n)} - \hat{\boldsymbol{x}}^{(n)}||^2$, between inputs $\boldsymbol{x}$ and reconstructions $\hat{\boldsymbol{x}}$.

After successful training, an autoencoder has learned an efficient representation $\boldsymbol{z}$ of the input from which it can be reconstructed using the decoder. The most basic version of the autoencoder uses feedforward NNs with linear activations as depicted in Figure 2.1, which is provably equivalent to Principal Component Analysis (PCA) [46].

The EncDec model [63] can be viewed as a sequential extension of the autoencoder when trained on minimizing the reconstruction error. Instead of using feedforward NNs the EncDec model uses Recurrent Neural Networks (RNNs).

The EncDec model also consists of an encoder and decoder, where the encoder takes a time series $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T)$ as input and updates its hidden states $\boldsymbol{h}_t^E$ at each time step t. The last hidden state of the

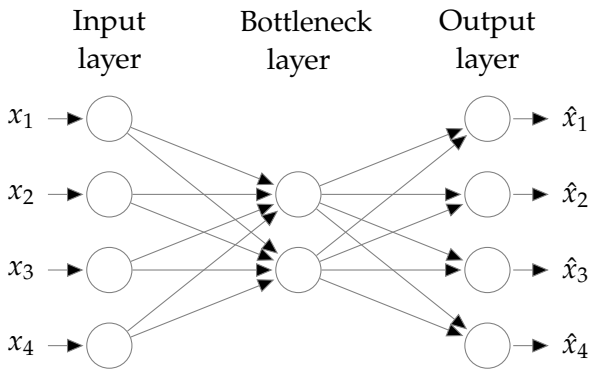


Figure 2.1: Linear autoencoder with bottleneck layer of size two.

encoder $\boldsymbol{h}_T^E$ then acts as a fixed size latent state representation $\boldsymbol{z}$ of $\boldsymbol{x}$ and is used as the first hidden state of the decoder $\boldsymbol{h}_T^D$.

For decoding, a feed-back mechanism is applied where the previous output $\hat{\boldsymbol{x}}_{t-1}$ during inference and ground truth $\boldsymbol{x}_{t-1}$ during training is taken as input to the decoder to update its hidden states $\boldsymbol{h}_t^D$. During preliminary experiments, we found that this feed-back mechanism leads to the decoder learning the identity function. Therefore, we propose a simplification of the original model omitting this feed-back.

Finally, a linear layer on top of the decoder is used to compute the reconstruction $\hat{\boldsymbol{x}}_t = \boldsymbol{w}^T \boldsymbol{h}_t^D + \boldsymbol{b}$.

Both the encoder and decoder are jointly trained to minimize the MSE on the reversed time series $(\boldsymbol{x}_T, \boldsymbol{x}_{T-1}, ..., \boldsymbol{x}_1)$ which was found to increase the performance of EncDec models [63].

The full architecture of the EncDec model is depicted in Figure 2.2.

2.2 VARIATIONAL AUTOENCODER

The Variational Autoencoder (VAE) framework [31, 53] combines the autoencoding concept with probabilistic models to model complex data distributions by sampling from simpler distributions and then applying non-linear transformations.

For the VAE framework, the deterministic code $\boldsymbol{z}$ and reconstruction $\hat{\boldsymbol{x}}$ of the original autoencoder are both replaced by distributions:

- The encoder Multi-Layer Perceptron (MLP), also referred to as recognition model, produces a distribution $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ called approximate posterior. Thus, we do not only have a single representation of the input in the latent space, but a distribution of codes $\boldsymbol{z}$. Often, diagonal Gaussians are used for which the encoder outputs the distribution parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The parameters ϕ are comprised of the weights and biases of the NN.
- The decoder, also called generating model, is also an NN and takes a sample from $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ as input and outputs the parameters of a distribution over the observation space $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$. The

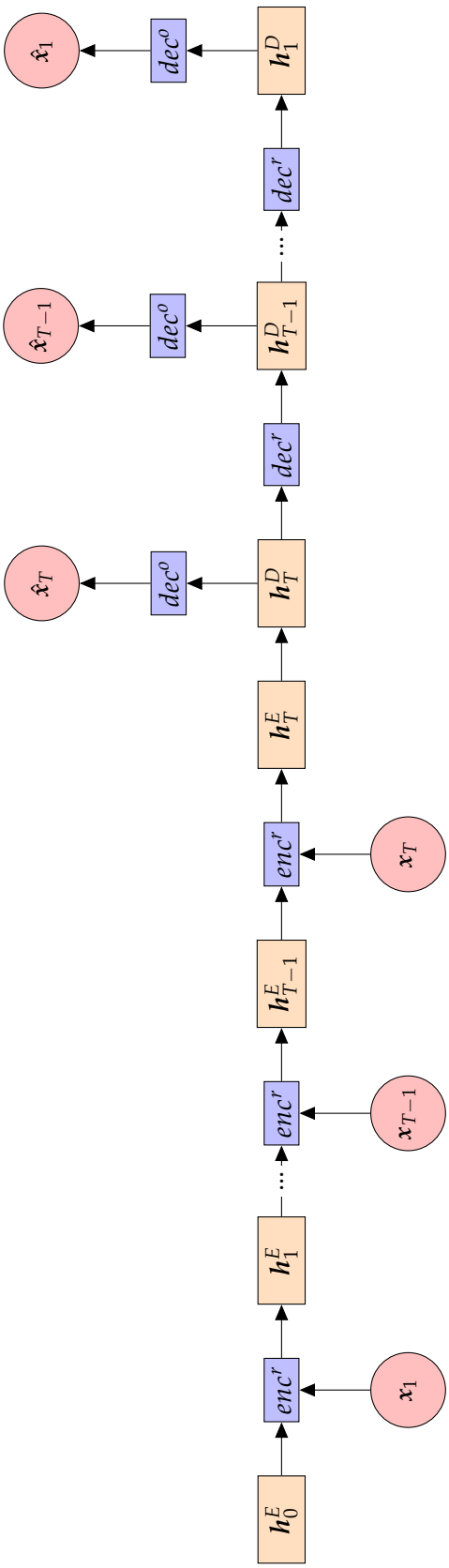


Figure 2.2: Computational flow of the EncDec model depicted as a directed graph. Red circles mark the inputs and reconstructions while orange rectangles correspond to the deterministic hidden states of the encoder (E) and decoder (D). Blue rectangles correspond to functional blocks of the NNs and represent a forward pass through either the recurrent connections (r) or output connections (o).

distribution type depends on the data but is assumed to be a diagonal Gaussian for the remainder of this thesis. This distribution is also called likelihood or emission. The parameters θ consist of the weights and biases of the MLP.

It should be noted that the VAE framework is not restricted to MLPs or Gaussian distributions. MLPs can be replaced by any function approximation and any parametric probability distribution can be used to represent the latent variables and the emissions. Choosing a different distribution or function approximation might however affect the training procedure outlined below.

Our objective using the VAE is to learn complex non-linear distributions of our data. The VAE belongs to the class of latent variable models which model the probability distribution of a dataset by relating the observations $\boldsymbol{x}$ to unobserved latent variables $\boldsymbol{z}$ as a directed graphical model parameterized by θ using the law of total probability:

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\theta(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} \tag{2.1}$$

Our objective is to maximize the log probability of our data using this model:

$$\arg\max_\theta \log p_\theta(\boldsymbol{x}) = \arg\max_\theta \log \int p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\theta(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} \tag{2.2}$$

Unfortunately, a closed-form solution for θ does not exist as the marginalization over $\boldsymbol{z}$ occurs inside of a logarithm. This problem can be solved using variational inference to find an approximation $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ of the unknown true posterior distribution $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$ from a variational family Q of distributions parameterized by ϕ. The log probability can then be reformulated to

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}) &= \log p_\theta(\boldsymbol{x}) \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \, \mathrm{d}\boldsymbol{z} \\
&= \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log p_\theta(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{z} \\
&= \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z}) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \, \mathrm{d}\boldsymbol{z} \\
&= \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \, \mathrm{d}\boldsymbol{z} + \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{p_\theta(\boldsymbol{z} \mid \boldsymbol{x})} \, \mathrm{d}\boldsymbol{z} \\
&= \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \, \mathrm{d}\boldsymbol{z} + \mathrm{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z} \mid \boldsymbol{x})\big).
\end{aligned} \tag{2.3}$$

Since $\mathrm{KL}(\cdot \,||\, \cdot) \geq 0$ holds, we have

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) &= \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \,\mathrm{d}\boldsymbol{z} + \mathrm{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z} \mid \boldsymbol{x})\big) \\ &\geq \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \,\mathrm{d}\boldsymbol{z} \\ &=: \mathscr{L}_{\mathrm{VAE}},\end{aligned} \tag{2.4}$$

where $\mathscr{L}_{\mathrm{VAE}}$ is called Evidence Lower Bound (ELBO) or variational lower bound as it is a lower bound on the evidence $\log p_\theta(\boldsymbol{x})$. This bound becomes tight when the approximate posterior distribution $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ is equal to the true posterior distribution $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$.

It can be rewritten as

$$\begin{aligned}\mathscr{L}_{\mathrm{VAE}} &= \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \,\mathrm{d}\boldsymbol{z} \\ &= \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} - \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \log \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{p_\theta(\boldsymbol{z})} \,\mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}_{q_\phi}[p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] - \mathrm{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z})\big).\end{aligned} \tag{2.5}$$

From Equation (2.5) an intuitive interpretation of the ELBO becomes clear: the first term acts as a probabilistic equivalent to the reconstruction error and encourages q_ϕ to place probability mass on latent variables that explain the observed data. The second term acts as a regularizer and encourages q_ϕ to be close to the prior. This results in a trade-off between a compact latent space structure and an accurate reconstruction.

If we replace $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ by the parameterized generating model and introduce a standard Normal prior $p(\boldsymbol{z})$ for $p_\theta(\boldsymbol{z})$ we arrive at a tractable way to maximize the lower bound of our data. Then, we can find the optimal parameters θ and ϕ of the VAE by using the backpropagation algorithm commonly used to train NNs.

However, finding these weights for both the recognition and generating model requires the gradients for $\mathscr{L}_{\mathrm{VAE}}$ with respect to θ and ϕ. Obtaining them for the encoder is straightforward when using Gaussian distributions as there is a closed form solution for the KL divergence. However, for the decoder, a Monte Carlo approximation of the expectation $\mathbb{E}_{q_\phi}[p_\theta(\boldsymbol{x} \mid \boldsymbol{z})]$ is necessary which is not easily differentiable as it involves sampling from a distribution. For Gaussian distributions, this can be solved with the so called reparameterization trick. Instead of directly drawing $z \sim \mathcal{N}(\mu, \sigma^2)$ we can draw $\epsilon \sim \mathcal{N}(0, 1)$ and then reparameterize z with a deterministic variable such that $z = \mu + \sigma\epsilon$. Using this trick we obtain a differentiable estimator of the ELBO.

2.3 STOCHASTIC RECURRENT NETWORK

VAEs have the inherent disadvantage that they can only model data with a fixed length. In order to model sequences of arbitrary length, it is necessary to derive a sequential version of the VAE framework leading to sequential latent variable models.

RNNs are able to approximate any measurable sequence-to-sequence mapping of arbitrary length and are Turing-complete [59]. Therefore, they are a suitable choice to replace the MLPs of the VAE which leads to Stochastic Recurrent Networks (STORNs).

Similar to the VAE, our objective is to learn complex non-linear distributions of our data. Again, we can model a datasets probability distribution by relating latent variables $z_{1:T}$ to observations $x_{1:T}$ using the multiplication rule and the law of total probability:

$$\begin{aligned} p_\theta(x_{1:T}) &= \prod_{t=1}^{T} p_\theta(x_t \mid x_{1:t-1}) \\ &= \int_{z_{1:T}} p_\theta(z_{1:T}) \prod_{t=1}^{T} p_\theta(x_t \mid x_{1:t-1}, z_{1:t}, z_{t+1:T}) \, \mathrm{d}z_{1:T} \end{aligned} \tag{2.6}$$

The second term $p_\theta(x_t \mid x_{1:t-1}, z_{1:t}, z_{t+1:T})$ corresponds to the temporal transition and is modeled by an RNN. In contrast to the VAE, x is not only conditioned on $z_{1:T}$ but on both $x_{1:t-1}$ and $z_{1:T}$.

There exist several RNN architectures that can approximate this transition term. For this section, we will restrict ourselves to the following simplistic RNN architecture with a single layer, weights $W_{in}, \bar{W}_{in}, W_{out}, W_{recurr}$, and biases $b_{init}, b_{hidden}, b_{out}$:

$$h_0 = b_{init}, \tag{2.7}$$

$$h_t = f_h(x_{t-1} W_{in} + z_t \bar{W}_{in} + h_{t-1} W_{recurr} + b_{hidden}), \tag{2.8}$$

$$y_t = f_y(h_t W_{out} + b_{out}), \tag{2.9}$$

where f_h and f_y are both nonlinear activation functions. The output y_t of the RNN then consists of the parameters of the transition distribution $p_\theta(x_t \mid x_{1:t-1}, z_{1:t}, z_{t+1:T})$ parameterized by the weights and biases of the RNN. In general, the framework can be adapted to other RNN architectures with reasonable effort as long as they preserve the dependencies of the transition term.

Choosing this architecture introduces two assumptions. First, there is no effect of future latent variables on present observations:

$$p_\theta(x_t \mid x_{1:t-1}, z_{1:t}, z_{t+1:T}) = p_\theta(x_t \mid x_{1:t-1}, z_{1:t}) \tag{2.10}$$

For the second assumption we introduce the hidden states $\boldsymbol{h}_{0:T}$ of the RNN to the transition term $p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t})$ by again making use of the law of total probability:

$$p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}) = \int_{\boldsymbol{h}_{0:t}} p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}, \boldsymbol{h}_{0:t}) p_\theta(\boldsymbol{h}_{0:t} \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}) \, \mathrm{d}\boldsymbol{h}_{0:t}. \tag{2.11}$$

By Equation (2.9) $\boldsymbol{x}_t$ is independent of $\boldsymbol{z}_{1:t}$, $\boldsymbol{x}_{1:t-1}$ and $\boldsymbol{h}_{0:t-1}$ given $\boldsymbol{h}_t$. Furthermore, $\boldsymbol{h}_t$ only depends on $\boldsymbol{x}_{t-1}$, $\boldsymbol{h}_{t-1}$, and $\boldsymbol{z}_t$ which allows us to simplify Equation (2.11) to

$$\begin{aligned} p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}) &= \int_{\boldsymbol{h}_{0:t}} p_\theta(\boldsymbol{x}_t \mid \boldsymbol{h}_t) p_\theta(\boldsymbol{h}_{0:t} \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}) \, \mathrm{d}\boldsymbol{h}_{0:t} \\ &= \int_{\boldsymbol{h}_{0:t}} p_\theta(\boldsymbol{x}_t \mid \boldsymbol{h}_t) p_\theta(\boldsymbol{h}_0) \prod_{s=1}^{t} p_\theta(\boldsymbol{h}_s \mid \boldsymbol{x}_{s-1}, \boldsymbol{z}_s, \boldsymbol{h}_{s-1}) \, \mathrm{d}\boldsymbol{h}_{0:t}. \end{aligned} \tag{2.12}$$

The hidden state $\boldsymbol{h}_t$ is a deterministic function of $\boldsymbol{x}_{1:t-1}$, $\boldsymbol{h}_{1:t-1}$, and $\boldsymbol{z}_t$ according to Equation (2.8) and $\boldsymbol{h}_0$ is a scalar. Thus, $p_\theta(\boldsymbol{h}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{z}_s, \boldsymbol{h}_{t-1})$ follows a Dirac distribution for which integrals over the hidden states are replaced by a single point. Making this explicit we get the final form of the transition for STORN:

$$p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t}) = p_\theta(\boldsymbol{x}_t \mid \boldsymbol{h}_t(\boldsymbol{x}_{t-1}, \boldsymbol{z}_t, \boldsymbol{h}_{t-1})). \tag{2.13}$$

After deriving the transition, we return to the first term $p_\theta(\boldsymbol{z}_{1:T})$ of Equation (2.6) which corresponds to the prior. In the original STORN paper [4] a standard Normal distribution is chosen that is independent across time steps:

$$p_\theta(\boldsymbol{z}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{z}_t). \tag{2.14}$$

It has been pointed out in literature [13] that this prior is not adaptive as it is not able to incorporate trends from previous samples. Soelch et al. [61] addressed this criticism by deriving a different factorization of Equation (2.6) using an adaptive prior $p_\psi(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t-1})$ conditioned on previous inputs and latent states. However, their anomaly detection results did not show a significant improvement in using the adaptive prior and in many cases the static prior achieved better results which we confirmed during preliminary experiments on our datasets. Therefore, we use the factorizing prior and refrain from describing the adaptive prior in more detail here. We instead refer to Soelch et al. [61] for an in-depth description.

Putting everything together, the likelihood of the data can be expressed as:

$$p_\theta(\boldsymbol{x}_{1:T}) = \int_{\boldsymbol{z}_{1:T}} \prod_{t=1}^{T} p(\boldsymbol{z}_t) p_\theta(\boldsymbol{x}_t \mid \boldsymbol{h}_t(\boldsymbol{x}_{t-1}, \boldsymbol{z}_t, \boldsymbol{h}_{t-1}))\, \mathrm{d}\boldsymbol{z}_{1:T} \tag{2.15}$$

During preliminary experiments, we found that STORN experiences a similar problem as the EncDec model: Conditioning $\boldsymbol{h}_t$ on $\boldsymbol{x}_{1:t-1}$ led to the model relying almost exclusively on $\boldsymbol{x}_{1:t-1}$ by learning an approximation of the identity function. Therefore, we propose a modification of STORN for which $\boldsymbol{h}_t$ does not depend on $\boldsymbol{x}_{1:t-1}$:

$$p_\theta(\boldsymbol{x}_{1:T}) = \int_{\boldsymbol{z}_{1:T}} \prod_{t=1}^{T} p(\boldsymbol{z}_t) p_\theta(\boldsymbol{x}_t \mid \boldsymbol{h}_t(\boldsymbol{z}_t, \boldsymbol{h}_{t-1}))\, \mathrm{d}\boldsymbol{z}_{1:T}. \tag{2.16}$$

As we do not have the ground truth values for $\boldsymbol{z}_{1:T}$, we need a trainable recognition model that approximates the true posterior $p(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})$. Again, we use variational inference to solve this issue by using a second RNN that outputs the parameters of the approximate posterior distribution $q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})$ parameterized by the weights and biases of the RNN. In contrast to the generating model, any recurrent architecture can be used, including bidirectional RNNs.

Similar to the VAE we can now derive a lower bound to the data likelihood which can be used to find the optimal parameters for the generating and recognition model and is similar to $\mathscr{L}_{\text{VAE}}$:

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}_{1:T}) &= \int_{\boldsymbol{z}_{1:T}} q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \log \frac{p_\theta(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}) q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})}{p_\theta(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})}\, \mathrm{d}\boldsymbol{z}_{1:T} \\
&= \int_{\boldsymbol{z}_{1:T}} q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \log \frac{p_\theta(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T})}{q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})}\, \mathrm{d}\boldsymbol{z}_{1:T} \\
&\quad + \int_{\boldsymbol{z}_{1:T}} q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \log \frac{q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})}{p_\theta(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})}\, \mathrm{d}\boldsymbol{z}_{1:T} \\
&\geq \int_{\boldsymbol{z}_{1:T}} q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \log \frac{p_\theta(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T})}{q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})}\, \mathrm{d}\boldsymbol{z}_{1:T} \\
&= \mathbb{E}_{q_\phi}\left[\sum_{t=1}^{T} \log p_\theta(\boldsymbol{x}_t \mid \boldsymbol{h}_t)\right] - \mathrm{KL}\big(q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \,\big|\big|\, p(\boldsymbol{z}_{1:T})\big) \\
&=: \mathscr{L}_{\text{STORN}}
\end{aligned} \tag{2.17}$$

The full architecture of STORN is depicted in Figure 2.3.

2.4 DEEP VARIATIONAL BAYES FILTER

STORN has the inherent disadvantage that the latent space at each time step does not contain all information to reconstruct a sample due to

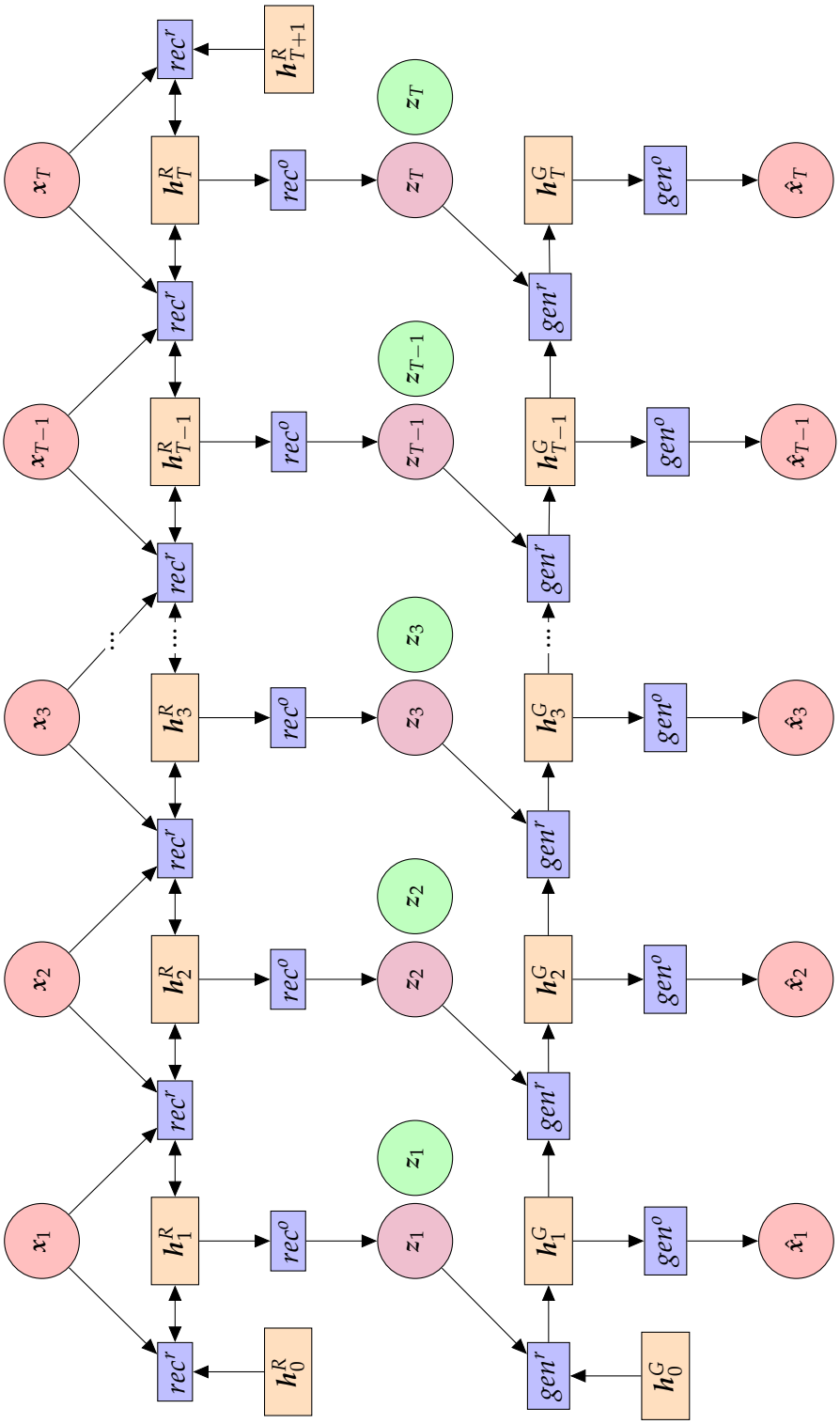


Figure 2.3: Computational flow of STORN depicted as a directed graph. Red circles mark the inputs and reconstructions while orange rectangles correspond to the deterministic hidden states of the recognition (R) and generating model (G). Blue rectangles correspond to functional blocks of the NNs and represent a forward pass through either the recurrent connections (r) or output connections (o). Priors are marked as green circles and the corresponding approximate posteriors as purple circles.

the recurrent connections in the generating model and the feed-back of the predictions (in the original formulation of STORN). Therefore, downstream applications, such as anomaly detection in the latent space, might suffer in performance as an expressive latent space is essential for them.

State-Space Models (SSMs) are an alternative to STORN with a latent space that encapsulates all information at a given point in time to predict the transition from z_{t-1} to z_t and to produce an observation x_t for the current latent state z_t. The distribution of observations for a SSM is decomposed as

$$p(x_{1:T}) = \int_{z_{1:T}} p(x_{1:T} \mid z_{1:T}) p(z_{1:T}) \, \mathrm{d}z_{1:T} \tag{2.18}$$

$$p(z_{1:T}) = p(z_1) \prod_{t=1}^{T-1} p(z_{t+1} \mid z_t) \tag{2.19}$$

$$p(x_{1:T} \mid z_{1:T}) = \prod_{t=1}^{T} p(x_t \mid z_t), \tag{2.20}$$

where $p(z_1)$ corresponds to the initial state, $p(z_{t+1} \mid z_t)$ to the transition, and $p(x_t \mid z_t)$ to the emission.

As for STORN, we use a recognition and generating model for learning a SSM:

- The *recognition model* infers the latent states by producing the parameters of the approximate posterior distribution $q_\phi(z_{1:T} \mid x_{1:T})$ as the true posterior is not known.
- The *generating model* outputs a distribution on the reconstructions and consists of the prior transition $p_{\theta_{trans}}(z_{1:T})$ and the emission $p_{\theta_{emis}}(x_{1:T} \mid z_{1:T})$.

The Deep Variational Bayes Filter (DVBF) [27, 28] explicitly models state-space assumptions to enforce a latent space that captures all information allowing accurate reconstruction and long-term sampling. There are two different implementations of DVBF. We describe the more recent one [27] which reformulates DVBF and was empirically shown to converge faster during training. We omit actions $u_{1:T}$ present in the original formulation, as the datasets used in this thesis do not contain any.

DVBF consists of the following components:

1. The *emission model* is an MLP that takes the latent state z_t as input and outputs the parameters of a probability distribution:

$$p_{\theta_{emis}}(x_{1:T} \mid z_{1:T}) = \prod_{t=1}^{T} p_{\theta_{emis}}(x_t \mid z_t) \tag{2.21}$$

   θ_{emis} are the weights and biases of the MLP.

2. The *recognition model* is divided into a *transition* and *inverse measurement* model. It is implemented as a product of both:

$$q_\phi(\boldsymbol{z}_{t+1} \mid \boldsymbol{z}_t, \boldsymbol{x}_{t+1}) \propto q_{\phi_{meas}}(\boldsymbol{z}_{t+1} \mid \boldsymbol{x}_{t+1}) \times q_{\phi_{trans}}(\boldsymbol{z}_{t+1} \mid \boldsymbol{z}_t) \quad (2.22)$$

   This allows DVBF to decide whether to base its belief about $\boldsymbol{z}_{t+1}$ on $\boldsymbol{z}_t$ or $\boldsymbol{x}_{t+1}$. Furthermore, it is possible to regularize the recognition model by sharing parameters between the *approximate posterior transition* and *prior transition*.

   The *approximate posterior* is then decomposed in an auto-regressive fashion:

$$q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) = q_{\phi_0}(\boldsymbol{z}_1 \mid \boldsymbol{x}_{1:K}) \prod_{t=1}^{T-1} q_\phi(\boldsymbol{z}_{t+1} \mid \boldsymbol{z}_t, \boldsymbol{x}_{t+1}) \quad (2.23)$$

   Again both the *transition* and *inverse emission* are modeled by MLPs with weights and biases ϕ_{meas} and ϕ_{trans}, respectively.

3. The initial latent state q_{ϕ_0} is modeled separately by an RNN denoted *initial inference model* that outputs the parameters of a distribution $q_{\phi_0}(w_1 \mid \boldsymbol{x}_{1:K})$, where K corresponds to the number of initial time steps used to infer w_1. The prior for w_1 is a standard Normal distribution.

   A sample is drawn from $q_{\phi_0}(w_1 \mid \boldsymbol{x}_{1:K})$ to infer the initial latent state $\boldsymbol{z}_1$ by passing it through a deterministic MLP called *initial transition* to ensure that it is not distributed according to a standard Normal prior.

4. The *prior transition* is also modeled by an MLP:

$$p_{\theta_{trans}}(\boldsymbol{z}_{1:T}) = p_{\theta_0}(\boldsymbol{z}_1) \prod_{t=1}^{T-1} p_{\theta_{trans}}(\boldsymbol{z}_{t+1} \mid \boldsymbol{z}_t) \quad (2.24)$$

   The prior for the initial latent state $p_{\theta_0}(\boldsymbol{z}_1)$ is modeled by a distribution $p_{\theta_0}(\boldsymbol{w}_1)$ with parameters θ_0 from which a sample is drawn and passed through the *initial transition model*. In our implementation, a standard Normal distribution is used.

   The parameter sharing between *prior transition* and *approximate posterior* happens between the means of the *prior transition* and *transition*:

$$\mu_{\phi_{trans}} = \mu_{\theta_{trans}}. \quad (2.25)$$

In general, all conditional distributions listed above are modeled as Gaussian distributions using DVBF.

Similar to STORN, we can now derive a lower bound to the data likelihood which can be used to find the optimal parameters for DVBF:

$$
\begin{aligned}
&\log p(\boldsymbol{x}_{1:T}) \\
&= \int_{\boldsymbol{z}_{1:T}} q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \log \frac{p_{\theta_{emis}}(\boldsymbol{x}_{1:T} \mid \boldsymbol{z}_{1:T}) p_{\theta_{trans}}(\boldsymbol{z}_{1:T})}{p_{\theta_{emis}}(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})} \, \mathrm{d}\boldsymbol{z}_{1:T} \\
&\geq \int_{\boldsymbol{z}_{1:T}} q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \log \frac{p_{\theta_{emis}}(\boldsymbol{x}_{1:T} \mid \boldsymbol{z}_{1:T}) p_{\theta_{trans}}(\boldsymbol{z}_{1:T})}{q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T})} \, \mathrm{d}\boldsymbol{z}_{1:T} \\
&= \mathbb{E}_{q_\phi}[\log p_{\theta_{emis}}(\boldsymbol{x}_{1:T} \mid \boldsymbol{z}_{1:T})] - \mathrm{KL}\big(q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \,||\, p_{\theta_{trans}}(\boldsymbol{z}_{1:T})\big) \\
&=: \mathscr{L}_{\text{DVBF}}
\end{aligned}
\tag{2.26}
$$

Because of the parameter sharing between *prior transition* and *approximate posterior*, the gradients from the reconstruction loss flow through the prior transition as well which allows the prior transition to shape the latent space. Without this sharing it would only be present in the KL divergence term of the loss.

The full architecture of DVBF is depicted in Figure 2.4.

2.5 MODEL VARIATIONS

We derived two new model architectures based on existing sequential latent variable models. The Variational EncDec model is a variational extension of the original EncDec model replacing the latent state and reconstruction by distributions and incorporating a prior on the latent space. Fusion STORN extends the bidirectional recognition model of STORN by fusing the beliefs about the latent state of a forward and backward RNN.

In this section, we describe both models in detail and the reasons why they are not used in the remainder of this thesis.

2.5.1 *Variational Encoder-Decoder*

The Variational EncDec model adapts the EncDec model to the VAE framework by replacing the latent state and reconstruction by distributions and adding a standard Normal prior on the fixed-size latent space. This allows us to use variational inference and apply ELBO-based anomaly detection.

The data likelihood then factorizes to

$$
p_\theta(\boldsymbol{x}_{1:T}) = \int_{\boldsymbol{z}} p_\theta(\boldsymbol{x}_{1:T} \mid \boldsymbol{z}) p(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}, \tag{2.27}
$$

where $p_\theta(\boldsymbol{x}_{1:T} \mid \boldsymbol{z})$ corresponds to the decoder of the EncDec model with the difference that it outputs the parameters of a Gaussian distribution.

As for other sequential latent variable models, we introduce an approximation $q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_{1:T})$ of the true posterior using the encoder that

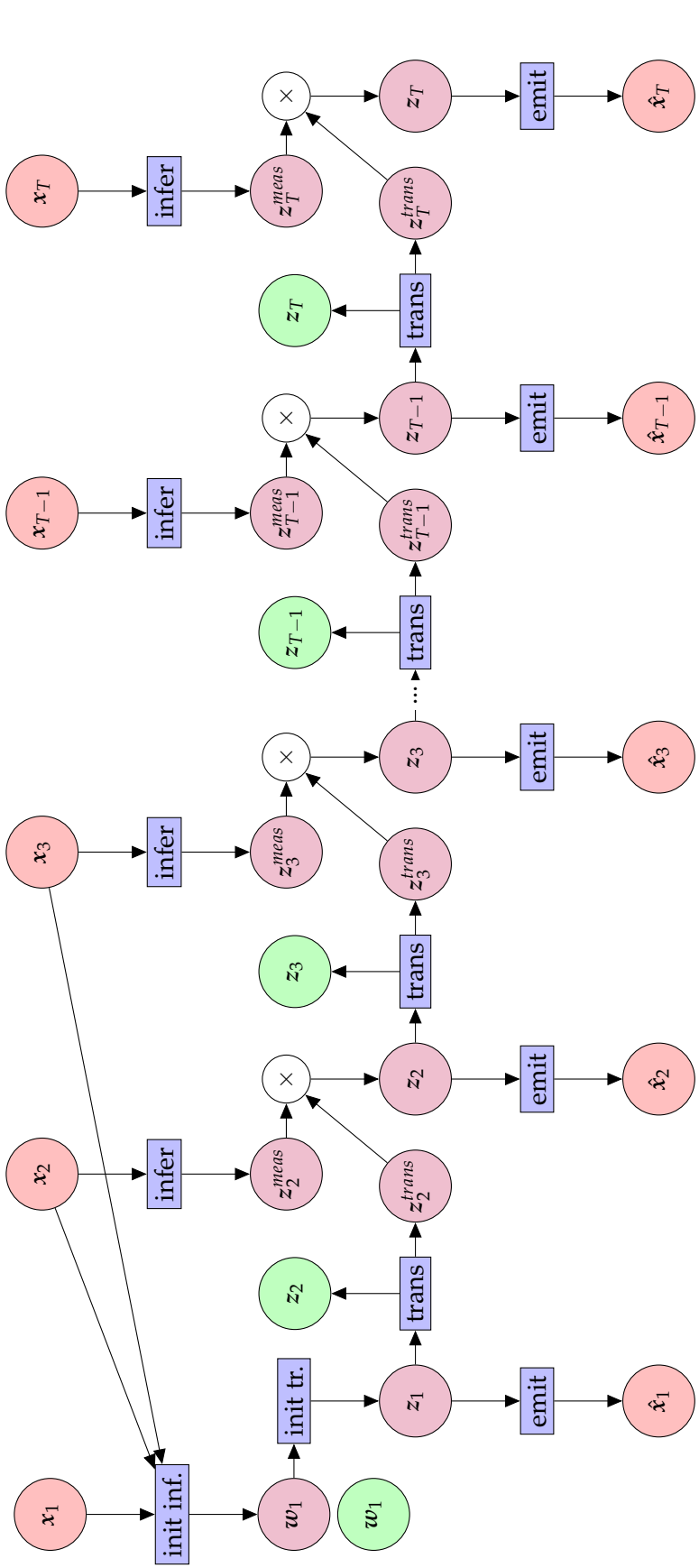


Figure 2.4: Computational flow of DVBF depicted as a directed graph. Red circles mark the inputs and reconstructions while blue rectangles correspond to functional blocks of the NNs and represent a forward pass through the respective MLP (or RNN in case of the initial inference model). Priors are marked as green circles and the corresponding approximate posteriors as purple circles.

outputs the distribution parameters. The model can then be trained using the lower bound:

$$p(\boldsymbol{x}_{1:T}) \geq \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{x}_{1:T} \mid \boldsymbol{z})] - \mathrm{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_{1:T}) \;||\; p(\boldsymbol{z})\big). \quad (2.28)$$

However, during all of our preliminary experiments, we observed that the KL divergence converges to zero, causing the encoding model to degenerate. Thus, no information is passed from the encoder to the decoder, which is why we decided to exclude it from further experiments.

2.5.2 *Fusion Stochastic Recurrent Network*

Fusion STORN adapts the bidirectional recognition model of STORN by replacing it with a forward and backward RNN.

Both RNNs output the parameters of a Gaussian distribution on the latent states $z_{1:T}$. Their beliefs are then merged by taking the product of both distributions similar to the recognition model of DVBF (cf. Equation (2.22)).

The approximate posterior distribution then factorizes to:

$$q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}_{1:T}) \propto \prod_{t=1}^{T} q_{\phi_{forward}}(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t}) \times q_{\phi_{backward}}(\boldsymbol{z}_t \mid \boldsymbol{x}_{t:T}). \quad (2.29)$$

Fusion STORN uses the same generating model as STORN and can be trained using the same lower bound given in Equation (2.17).

During preliminary experiments, we found that Fusion STORN takes significantly longer to converge during training and achieves lower ELBO values compared to STORN. Therefore, we decided to exclude it from further experiments.

3 RELATED WORK

There exists a plethora of different approaches for the two PHM tasks of anomaly detection and prognostics, as both have numerous real-life applications to reduce costs by preventing total system failures. They range from model-based approaches considering the physical properties of a given system to data-driven approaches relying on machine learning models trained on sequences obtained from the given sensors.

This section presents an overview of the fields of anomaly detection and prognostics with an emphasis on methods relying on NNs to justify a novel approach that is applicable to both fields.

3.1 ANOMALY DETECTION

Anomaly detection can be applied to a variety of different problems such as intrusion detection or fault detection for mechanical components. Researchers have adopted concepts from several fields such as statistics, machine learning, and information theory to anomaly detection over the past decades [50].

Pimentel et al. [50] performed a thorough review of existing approaches and derived a taxonomy for anomaly detection methods that separates them into five classes:

1. *Probabilistic methods* estimate a probability distribution of normal data. Anomalous samples can then be detected by applying a threshold on the likelihood as anomalies lie in regions with low probability mass.

   Techniques used to estimate this distribution fall into parametric and non-parametric approaches. Both approaches have the advantage that, after training, only a minimal amount of information needs to be retained compared to other methods. Parametric methods include Gaussian mixture models which do, however, make assumptions on the structure of the data and often fail if these assumptions are not met. Non-parametric models such as kernel density estimators are more flexible as they do not assume a fixed structure for a model. The main drawback of non-parametric models is that they do not scale well to high-dimensional data.

2. *Distance-based methods* assume that normal data clusters in space and use distance measures between samples to define normal-

ity. They encompass clustering- and nearest-neighbor-based approaches.

Detecting anomalies using K-Nearest Neighbors (KNN) is based on the assumption that anomalous samples are located far from their neighbors while normal samples have close neighbors. Clustering algorithms rely on a similar concept detecting anomalies based on the distance to the nearest cluster.

While both approaches do not require any prior knowledge of the data distribution, they rely on the existence of a suitable distance metric. Also, nearest-neighbor-based approaches do not scale well to large datasets as an increasing number of distances needs to be computed during inference.

3. *Reconstruction-based methods* autonomously model the data and detect anomalies based on the reconstruction error.

   Approaches often rely on NNs employing an autoencoding mechanism. Another class of methods project the data onto a specific subspace using for example PCA to detect anomalies.

   A major disadvantage of this class of methods is that especially NNs are sensitive to the choice of hyperparameters and not all of the approaches are applicable to time series data.

4. *Domain-based methods* solve the simpler problem of drawing a decision-boundary around normal data instead of finding a model that describes the data. Any sample outside this boundary is classified as anomalous. A popular model for this class of algorithms is the one-class Support Vector Machine (SVM).

   A major difficulty with domain-based approaches is choosing the right parameters for the size of the boundary region. Furthermore, the results are highly dependent on the chosen kernel function.

5. *Information-theoretic methods* rely on measures such as the entropy to detect anomalies. They assume that anomalies alter the information content of an otherwise normal dataset.

   While information-theoretic approaches do not make any assumptions on the data distribution, they are highly sensitive to the selection of the information-theoretic measure. Furthermore, often these measures only detect anomalies if there is a sufficiently large number of them present in the dataset.

All of the outlined approaches suffer from one or multiple drawbacks previously identified by Soelch [60] when applied on multivariate time series data:

- The model imposes strict assumptions on the data distribution which do not match the data.

- The approach is tailored to a specific application or type of anomaly and does not generalize.
- The model is not suited for multivariate time series data with high temporal dependencies often inherent to sequential sensor data.

RNNs have been proven to be universal function approximators [59] and thus can handle multivariate time series data with high temporal dependencies without making any assumptions on the data. They are therefore suitable for our given problem of anomaly detection on sequences.

We distinguish the approaches relying on deterministic RNNs for anomaly detection into several classes:

1. *Classification-based methods* learn a mapping from a sequence $\boldsymbol{x}$ to a probability y indicating whether a sequence is anomalous:

$$y = f(\boldsymbol{x}) \tag{3.1}$$

   Cheng et al. [11] and Kim et al. [29] both applied Long Short-Term Memory (LSTM)-based classifiers on an anomaly detection problem and demonstrated their superiority to other classification methods.

   However, classification-based approaches have the significant drawback that they require a great amount of labeled data for both normal and anomalous samples. For anomaly detection in general, a semi-supervised or unsupervised method is preferable due to the scarcity of anomalous data and since regular classification methods often have difficulties in dealing with an imbalanced dataset [62]. Also, classification-based approaches do not generalize to new types of anomalies that are not present in the training dataset.

2. *Prediction-based methods* fall into the class of reconstruction-based approaches and learn to predict future values based on past observations by learning the following mapping

$$\hat{\boldsymbol{x}}_{t+1:t+k} = f(\boldsymbol{x}_{t-l:t}), \tag{3.2}$$

   where l corresponds to the amount of time steps used as input to the model and k to the amount of predicted future time steps. Models are only trained on normal data by minimizing the error between observations and reconstructions. Anomalies are then detected based on the reconstruction error.

   Several researchers [10, 44, 48] proved the viability of prediction-based methods for anomaly detection on different datasets. However, the authors of [42] conclude that they are not suited for

systems influenced by external factors not recorded in the dataset as it becomes difficult to make predictions even into the near future.

3. *Autoencoder-based methods* belong to the class of reconstruction-based approaches. They learn a lower-dimensional representation of the input used to reconstruct the original sample as described in Section 2.1. Similar to prediction-based methods, models are only trained on normal data by minimizing the error between observations and reconstructions. Anomalies are then detected based on the reconstruction error.

   Malhotra et al. [42] successfully applied an EncDec model using LSTMs on several time series datasets for anomaly detection. Their method has the disadvantage that a fixed-size latent space representation is passed from the encoder to the decoder. While LSTMs are generally able to capture long-term dependencies [23], in practice the reconstruction ability of the EncDec model is often limited and does not scale well to long sequences [12].

4. *GAN-based methods* are similar to autoencoder-based approaches. Instead of using autoencoders, a Generative Adversarial Network (GAN) consisting of a generator and discriminator is trained on normal samples following the procedure described by Goodfellow et al. [19]. In order to do anomaly detection for a new sample, the GAN is used to find the closest latent space representation for this sample. Then, it is reconstructed using the generator as described by Schlegl et al. [57]. Finally, anomalies can be detected based on the reconstruction error.

   Li et al. [37] adapted the approach to time series using LSTMs and successfully applied it to an anomaly detection problem.

   GAN-based methods have the significant drawback that the mapping of a sample into the latent space is an optimization problem and thus computationally expensive. Also, training GANs is challenging due to a possible imbalance between generator and discriminator which prohibits learning [45].

A promising class of models used for anomaly detection are sequential latent variable models such as STORN. Soelch et al. [61] successfully applied STORN for anomaly detection on robot time series data. They are similar to the EncDec model but scale better to long sequences as there is a latent space representation passed from the encoding to the decoding model at every time step. This reduces the distance to relate signals from two arbitrary input and output positions compared to using a single fixed-size latent space representation. Furthermore, they have several other advantages justifying their application to anomaly detection problems:

1. Instead of only using the reconstruction error, they can take into account the variability of the reconstructions as they are given not as single values but as distributions. This was done by An et al. [2], who empirically proved on several datasets that a VAE outperforms a regular autoencoder.

2. Also, latent variable models offer other possibly more meaningful measures to define normality such as the lower bound. Soelch et al. [61] used STORN to successfully detect anomalies in robot time series using the lower bounds to discriminate between normal and anomalous sequences. Furthermore, since the latent space corresponds to a compact representation of the input it can also be used to detect anomalies.

3. Lastly, they incorporate priors on the latent space acting as a regularizer.

Given the drawbacks of most examined approaches and the successful application by Soelch et al. [61], a further investigation of sequential latent variable models for anomaly detection seems justified.

3.2 PROGNOSTICS

The field of prognostics has seen a significant increase in the number of papers published in the past ten years [35] due to its promise to increase systems health by enabling preventive measures. It has been applied to numerous mainly mechanical components such as bearings, milling machines, and turbofan engines [35].

A study on major challenges in the area of prognostics [14] identified two approaches to estimate the RUL of a system:

- *Physics-based approaches* formalize the degradation process based on the physical understanding of a mechanical component by using system-specific knowledge and degradation formulas.

  This approach requires a sufficient understanding of the components degradation and availability of the required parameters as sensor data.

- *Data-driven approaches* rely on the assumption that the degradation of a component is visible with statistical significance in the collected sensor data. They use statistical and machine learning methods to build models based on the available sensor data.

  The main problem of data-driven approaches is collecting a sufficient amount of high-quality data since most industrial components are not allowed to run until failure or take a long time until failures occur.

Choosing an approach for a given prognostics problem depends mainly on the available data, the knowledge of the underlying physics, and the general complexity of a component. However, with increasingly complex systems and an increasing amount of sensor data available, data-driven approaches and especially methods relying on NNs are gaining popularity.

There are two classes of neural methods used to estimate the RUL. Supervised approaches directly map the sensor input to the RUL and are trained on all available data. Semi-supervised approaches either drop a fraction of the available labels or rely on unsupervised pretraining before a mapping from input to RUL is learned:

- *Supervised approaches*: Babu et al. [3] use Convolutional Neural Networks (CNNs) to map sensor recordings to RUL estimates by applying convolutional filters along the time dimension of the data. They compared their approach with MLPs and other traditional machine learning algorithms and found that CNNs are generally superior. Li et al. [38] improved upon their initial results using a more sophisticated data preprocessing procedure.

  Given the temporal nature of the data, RNNs are an obvious choice. Zheng et al. [70] apply a multi-layered LSTM on the sensor data of the Commercial Modular Aero-Propulsion System Simulation (CMAPSS) dataset to directly predict the RUL. They outperformed the CNN used by Babu et al. [3] by a significant margin. Other researchers [24, 68] also obtained promising results using LSTMs on the CMAPSS dataset.

- *Semi-supervised approaches*: Yoon et al. [69] apply two semi-supervised approaches on the CMAPSS dataset. For both approaches, they drop different percentages of the labels during training. The first approach uses self-learning where the model is trained iteratively on its predictions of the previous iteration for samples with no labels available. The second approach uses a sequential latent variable model similar to STORN which is pretrained to reconstruct the whole run-to-failure trajectories. Afterwards, an RNN is used to predict the RUL based on the latent space representations.

  Malhotra et al. [43] employ an EncDec model on subsequences of the CMAPSS dataset to obtain latent space representations. The EncDec model is only trained on normal data at the beginning of an engines life and then a linear regression model is applied on the latent space representations to infer a health index modeled by an exponential function. From the training data, a database of health index curves is built and during inference, a matching is performed to find the closest representation in the database to estimate the RUL. Gugulothu et al. [20] extend this approach

by training the EncDec model on full run-to-failure trajectories instead of just normal data which slightly improves the results.

In summary, several researchers proved that using latent space representations [20, 43] and sequential latent variable models [69] for prognostics lead to state-of-the-art results.

However, they still rely on the full labeled dataset for their training procedure. Also, none of the listed approaches covers both anomaly detection and prognostics using the same model. This justifies the investigation into a novel approach using sequential latent variable models able to do both anomaly detection and prognostics while relying only on a subset of run-to-failure data.

4 DATASETS

In order to evaluate a universal approach for both anomaly detection and prognostics, we decided to use two distinct datasets. The first is exclusively used to evaluate the anomaly detection performance and the second for both anomaly detection and prognostics.

In general, any prognostics dataset can be used for anomaly detection by defining sequences close to the end of a component's life as anomalies. Datasets specifically used for anomaly detection are, however, scarce and the existing datasets used in literature are often not suited for our experiments.

Accordingly, in this section, we first list commonly used datasets for anomaly detection on sequential data and describe why they are not suitable for evaluating our methods. In Section 4.2 we then present the Paderborn dataset that is used in other research domains but fulfills all requirements to be used as a benchmark for anomaly detection. Finally, in Section 4.3 we describe the CMAPSS dataset which is used in the field of prognostics and can be adapted to anomaly detection serving as a benchmark for both PHM disciplines.

4.1 ANOMALY DETECTION DATASETS

As stated earlier, gathering data to evaluate anomaly detection methods is difficult as components are often not allowed to run until failure or it requires a lot of time which limits the amount of available anomaly detection datasets.

In literature, several datasets are used to evaluate anomaly detection methods. However, each of them has one or more of the following issues that make them suboptimal as a possible benchmark for our anomaly detection methods:

1. *Not enough data available*: The most common problem is that not enough data is available to allow a reliable estimation of the anomaly detection performance. The anomaly detection methods require a large number of sequences as several data splits of both normal and anomalous data are necessary to train a model and evaluate it for anomaly detection. If only a few anomalous sequences are available, the results will not be meaningful due to high variance.

   Datasets with an insufficient amount of data include the Power Demand, the ECG, and the Space Shuttle dataset used by Malhotra et al. [42] as well as the Numenta Anomaly Detection

Benchmark [33] and the Turbomachinery dataset used by Vishnu et al. [66].

2. *Labels set by author*: Another issue is that some datasets do not have labels separating them into normal and anomalous samples. Therefore, researchers resorted to labeling the data themselves. As a conclusion, the anomaly detection method can only be as good as the labeling process.

   This is the case for the Power Demand dataset used by Malhotra et al. [42].

3. *Proprietary datasets*: Other papers rely on proprietary data sources which are not available to the public and can therefore not be used as a benchmark.

   This applies to the Engine dataset used by Malhotra et al. [42].

Given these problems, we decided to look into datasets that were previously not used for anomaly detection but do not exhibit any of the listed issues.

4.2 PADERBORN DATASET

The Paderborn Bearing Data Center provides a benchmarking dataset [36] for condition monitoring of rolling bearings. It consists of multivariate time series and includes an extensive documentation of bearing damages and operating conditions.

Most bearing datasets are restricted to artificially introduced single point damages which stimulate vibrations at the characteristic frequencies of a bearing, making them comparably easy to detect. Natural damages are often distributed and induce broadband vibrations that are harder to distinguish from noise. This leads to a high discrepancy between methods proposed in scientific literature and their application in industry [36]. The Paderborn dataset addresses this issue by using distributed damages and different methods to introduce faults which makes them harder to distinguish from noise.

It contains data from 6 healthy and 26 damaged bearings. The bearings can be separated into different classes based on the damage location (no damage, damage at the inner ring, damage at the outer ring) and how the damages are introduced.

Artificial faults include damage created using Electric Discharge Machining (EDM) (with different trench lengths), drilling (with different diameters), and electric engraving (with different damage lengths) as depicted in Figure 4.1.

To obtain natural damages in bearings, accelerated lifetime test rigs are used in which the radial force is higher compared to real applications to decrease the time until fatigue damages appear. Damages then arise due to pitting or plastic deformation as depicted in Figure 4.2.

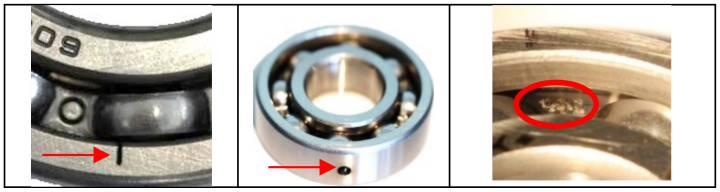


Figure 4.1: Artificial damages introduced by EDM (left), drilling (middle) and pitting with an electric engraver (right) (Image Source: [36]).

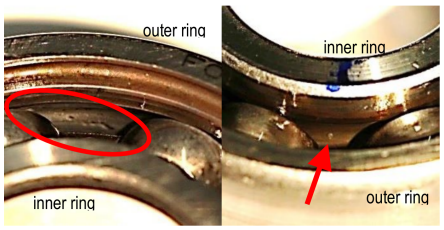


Figure 4.2: Real damages including indentation at the raceway of the outer ring (left) and pitting at the raceway of the inner ring (right) (Image Source: [36]).

After introducing faults, the Paderborn dataset was recorded from a test rig as depicted in Figure 4.3. The recordings were conducted under four different operating conditions with changing rotational speed, load torque, and radial load. For each fault and operating condition, 20 samples for a duration of 4 seconds are recorded. Each record is taken after dismantling and reassembling the bearing module to incorporate assembly variability which can be expected for bearings used in practice. During the experiment, the electric motor current and vibration signals are measured with a sampling rate of 64 kHz.

Currently, the Paderborn Bearing Data Center dataset is exclusively used for fault detection, i.e. classifying sequences into fault classes based on the location of the fault (healthy, inner ring, outer ring) [26, 36, 71]. However, the dataset is also perfectly suited as a benchmark for anomaly detection methods because of its clear separation into normal and faulty trajectories, a sufficient amount of normal and anomalous data, and accurate labeling.

There are several other datasets containing sensor data of faulty bearings which are commonly used for fault detection and could be adapted to anomaly detection. These include the CWRU dataset [40], the FEMTO bearing dataset [47], the MFPT fault dataset [5] and the IMS bearing dataset [34]. However, they all have multiple disadvantages compared to the Paderborn dataset, such as a smaller number of fault trajectories (FEMTO, MFTP, IMS), an almost trivial separability of normal and anomalous trajectories (CWRU), and no naturally occurring damages (CWRU, MFPT). Therefore, we decided to use the Paderborn dataset.

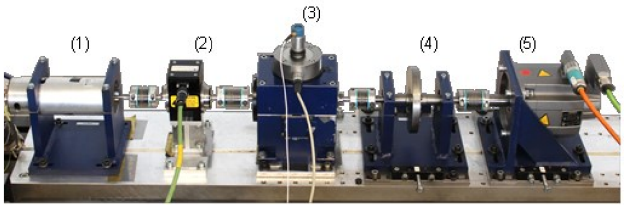


Figure 4.3: Modular test rig consisting of an electric motor (1), a torque-measurement shaft (2), a rolling bearing test module (3), a flywheel (4) and a load motor (5). It is used to record vibration and motor currents for all damaged and healthy bearings (Image Source: [36]).

For our experiments, we follow the recommendation of Lessmeier et al. [36] and use data from only one operating condition with a rotational speed of 900 RPM, a load torque of 0.70 Nm and a radial load of 1000 N. Furthermore, we decided to only use the vibration signal and discard the motor current as it was shown to be less meaningful for detecting faults.

In contrast to other researchers, we do not extract any statistical or frequency-based features and use our models on raw data. The only preprocessing we apply is smoothing the data using a mean filter with window size 10 and then downsampling it by a factor of 5 to increase computational efficiency. Furthermore, we scale it to zero mean and unit standard deviation based on the mean and standard deviation estimated on normal data. Finally, the time series are divided into sequences of length 400 with no overlap each containing several rotations of a bearing.

4.3 CMAPSS DATASET

The CMAPSS dataset was published by the NASA AMES research center [55] and consists of multivariate time series reflecting the degradation of a commercial-grade turbofan engine. The dataset was generated using the Commercial Modular Aero Propulsion System Simulation [49].

The main components of the simulated engine are fan, booster, high- and low-pressure compressor (HPC, LPC), burner, high- and low-pressure turbines (HPT, LPT), mixer, afterburner, and nozzle [49] as depicted in Figure 4.4. The simulation uses 13 health-parameters and fuel flow as input to model the degradation. Its output are the different sensor responses such as temperature and pressure. Each engine is run until a specified health threshold is reached, at which the engine is considered damaged.

The dataset is a collection of four subdatasets FD001–FD004 each consisting of a training and a test set. The train datasets contain several run-to-failure time series, while trajectories in the test datasets are

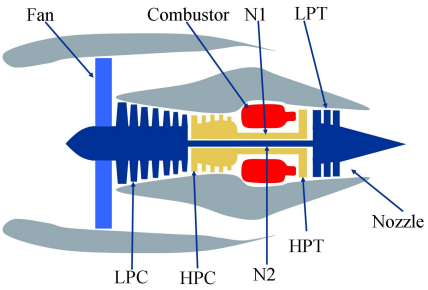


Figure 4.4: Main components of a turbofan engine simulated with CMAPSS (Image Source: [55]).

	CMAPSS			
	FD001	FD002	FD003	FD004
Train Trajectories	100	260	100	248
Test Trajectories	100	259	100	248
Fault Types	1	1	2	2
Operating Conditions	1	6	1	6

Table 4.1: Detailed overview of CMAPSS datasets.

pruned to stop some time prior to failure. Each unit has a lifespan between 128 and 543 cycles, where a cycle represents one snapshot of all sensors taken during a flight with the corresponding engine. The datasets contain different fault types and varying operating parameters in terms of altitude, Mach number, and throttle resolver angle. Different combinations of them are used which are randomly chosen for each flight and affect the degradation of a component. Table 4.1 gives an overview of all datasets, their number of trajectories, fault modes, and operating conditions.

The CMAPSS datasets are mainly used to evaluate prognostics methods although they fulfill all requirements for anomaly detection except that there is no label defining at which point the behavior of an engine becomes anomalous. Other researchers previously divided the run-to-failure trajectories into multiple classes based on the degradation using different methods such as clustering [67] or predefined RUL thresholds [64]. We decided to adopt the thresholding method to split the dataset into normal and anomalous sequences. Sequences with a RUL higher than 130 are defined as normal following the observations of Heimes [21] that degradation starts around that point in time. Sequences with a RUL lower than 50 are defined as anomalous following the threshold set by Tamilselvan et al. [64].

Several other datasets exist that are commonly used to evaluate prognostics methods. They include the FEMTO and IMS bearing dataset and the milling dataset [35]. However, they have significantly less run-to-failure trajectories compared to the CMAPSS dataset. Therefore,

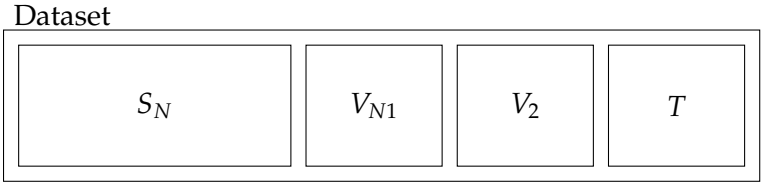


Figure 4.5: Dataset split used for anomaly detection and prognostics.

and since a large body of research is focused on the CMAPSS dataset [52], we decided to use it to evaluate our methods on both anomaly detection and prognostics.

Again, to avoid the introduction of domain-specific knowledge, we do not extract features commonly used by other researchers [51]. We only scale the data to zero mean and unit variance based on the mean and standard deviation estimated on the normal data. We also apply a rolling mean filter of size 5 to reduce noise in the data and exclude constant features from the dataset. Finally, the time series are divided into sequences of length 15 with an offset of 1.

4.4 DATASET SPLIT

In order to apply methods for anomaly detection and prognostics, the data needs to be split into several parts. We follow the common methodology used by other researchers in the field of anomaly detection [10, 42, 44] and split the data into four parts as depicted in Figure 4.5.

The splits have the following purpose:

- S_N: Normal data used for training a sequential latent variable model.
- V_{N1}: Normal data utilized for early stopping and selecting the best model during hyperparameter search.
- $V_2 = V_{N2} \cup V_A$: Normal and anomalous data to fit anomaly detection and prognostics methods.
- $T = T_N \cup T_A$: Normal and anomalous data to evaluate final model performance.

For prognostics, V_2 contains all run-to-failure trajectories from the CMAPSS training dataset and T contains the pruned run-to-failure trajectories from the CMAPSS test datasets. For anomaly detection, both V_2 and T contain data from the training dataset as the anomalous data for a RUL smaller than 50 is required for each engine which is not always present in the pruned test datasets.

To avoid information leakage, the splits are done trajectory-wise. Trajectories are randomly assigned to each split. In case of the Paderborn dataset this means that samples from the same faulty bearing

will only appear in one dataset split. However, since only four normal bearing recordings are available for the Paderborn dataset, the split for normal data is time-based. Thus, earlier samples are used for the S_N split and the latest normal samples for the T_N split. For the CMAPSS dataset, a specific engine will only appear in one of the splits as enough run-to-failure trajectories are available.

5 METHODS

This chapter describes existing and newly introduced anomaly detection and prognostics methods and how they are evaluated on the Paderborn and CMAPSS dataset described in Chapter 4.

Each anomaly detection and prognostics method relies on the outputs of a trained sequential latent variable model. Therefore, we first outline the training procedure in Section 5.1. Afterwards, Sections 5.2 and 5.3 describe the used anomaly detection and prognostics methods. Finally, the last section proposes a novel approach to select models during the hyperparameter search to improve results for both anomaly detection and prognostics.

5.1 TRAINING

In order to do both anomaly detection and prognostics, we first need to fit the sequential latent variable models on normal data. We use three different models: the EncDec model, STORN and DVBF.

Each of them has a variety of different hyperparameters that influence the performance. Therefore, a hyperparameter search is conducted using random search to find the best architectures. For each model, 64 different hyperparameter configurations are tested. The best configurations are then selected based on the lower bound (or MSE in case of the EncDec model). After training, the models with the best architecture can be used for both anomaly detection and prognostics.

The models are trained until convergence on the S_N dataset split. We apply early stopping based on the loss on the V_{N1} dataset. Optimization is done using Adam [30] in which the learning rate and β_1 are optimized during the hyperparameter search, while β_2 is left at its default value of 0.999. A batch size of 32 is used during training.

For the EncDec model, we test different parameters for the activation function, recurrent cell type, and weight initialization. Also, different dimensionalities for the latent space are tested to vary the amount of information that can be transmitted from the encoder to the decoder. There is no additional parameter for the hidden dimension for both encoder and decoder as it is directly coupled to the size of the latent space.

For STORN using the static prior we optimize the model architecture of the unidirectional generating and bidirectional recognition model including the recurrent cell type, number of hidden layers, and the number of units per layer. Also, different weight initialization schemes and activation functions are tested. Varying dropout rates are used

on the latents and the recurrent connections of both the generating and recognition model. Lastly, the size of the latent space is tunable as well.

For DVBF we test different latent space sizes, activation functions, and weight initialization schemes. Furthermore, we optimize the number of hidden units and layers for each MLP of DVBF separately.

A separate hyperparameter search is conducted for the Paderborn dataset and each of the CMAPSS datasets. Because of the large number of sequences available in the Paderborn dataset, the models are only trained on 10% of the sequences which are randomly selected. Then, only the best architecture for each sequential latent variable model is retrained on the full dataset. For the CMAPSS dataset, each model is trained on the full dataset during the hyperparameter search.

The complete hyperparameter search space for each model can be found in Appendix A.

5.2 ANOMALY DETECTION

After the hyperparameter search, an anomaly detector can be built using the trained models with the respective best architecture.

In this section, we will first describe the general anomaly detection process and the different methods relying on reconstruction errors, lower bounds, and latent space representations to detect anomalies. Then, we will outline how they are evaluated on the Paderborn and CMAPSS dataset.

5.2.1 *Methods*

Anomaly detection using a trained sequential latent variable model can be performed in two steps:

1. Compute anomaly score δ for each sequence based on outputs of the trained model.
2. Find optimal threshold $\hat{\kappa}$ on anomaly scores of V_2 dataset split to discriminate between normal and anomalous samples.

We focus on the first step of deriving anomaly scores and restrict ourselves to a single algorithm [60] to find a threshold which simultaneously minimizes the False Positive Rate (FPR) (probability that a normal sample is classified as anomaly) and maximizes the True Positive Rate (TPR) (probability that an anomalous sample is classified as anomaly):

$$\hat{\kappa} = \arg\min_{\kappa} (1 - \mathrm{TPR}^2) + \mathrm{FPR}^2 \tag{5.1}$$

This threshold corresponds to the spot in the top-left corner of the Receiver Operating Characteristic (ROC) curve explained in Section 5.2.2.

The following sections outline the different methods used to compute the anomaly scores.

5.2.1.1 *Reconstruction-based Approaches*

Reconstruction-based methods rely on the difference between reconstruction and observation to detect anomalies. For anomalous data this reconstruction error is assumed to be higher, as the model has only learned to reconstruct normal data during training.

Existing approaches using the reconstruction error for anomaly detection are:

- *Likelihood [42]*: This method first calculates the absolute difference between reconstruction and emission for each time step: $\boldsymbol{e}_t = |\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t|$. The samples in V_{N1} are then used to estimate the parameters of a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ on the reconstruction errors $\boldsymbol{e}_t$. The anomaly score a_t for a single time step is then computed as $a_t = (\boldsymbol{e}_t - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{e}_t - \boldsymbol{\mu})$ and the anomaly score for a whole sequence is calculated as $\delta = \max_t a_t$.

- *Global One-Step Prediction Error Threshold [61]*: This method is similar to the previous method but uses the reconstruction error directly as an anomaly score $a_t = ||\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t||^2$. Also, due to the risk of outliers, the 0.995 percentile is used instead of the maximum to compute the anomaly score δ for a whole sequence.

The *Likelihood* method has the inherent flaw that a standard Gaussian is used to model absolute differences. Since we assume that anomalous samples have a higher reconstruction error than normal samples, a half-normal distribution would be a better fit. Then, the method would achieve the same anomaly detection results as the *Global One-Step Prediction Error Threshold* since the probability density function is monotonically decreasing and the anomaly scores would thus only be a scaled version of the reconstruction error. Therefore, we do not apply it for anomaly detection on our datasets.

Anomalous samples in the used datasets often have higher reconstruction errors for most time steps compared to normal samples. The *Global One-Step Prediction Error Threshold* discards this information and only relies on the percentile. Therefore, we propose a novel method called *Global Prediction Error Threshold* which calculates the sum of all squared error residuals instead of using percentiles:

$$\delta = \sum_{t=0}^{T} ||\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t||^2 \tag{5.2}$$

For both the *Global One-Step Prediction Error Threshold* and the *Global Prediction Error Threshold* the emissions of a model are used to compute an anomaly score. During preliminary experiments, we found that

using the modes of the latent and emission distributions leads to better anomaly detection results compared to sampling from these distributions. Therefore, all subsequent experiments are conducted using the mode during inference.

5.2.1.2 *ELBO-based Approaches*

ELBO-based approaches rely on the lower bound to distinguish between normal and anomalous samples. They can only be used with STORN and DVBF as the EncDec model is not trained using variational inference.

After training, a sequential latent variable model has learned an approximation of the data likelihood of normal training sequences. As anomalies are by definition in low-density regions of that distribution, their likelihood will be lower which allows to distinguish them from normal data.

Existing approaches using the lower bound for anomaly detection are:

- *Lower Bound [61]*: The variational lower bound is an obvious choice for an anomaly score as it is a lower bound on the data likelihood.
- *Global Stepwise Lower Bound Threshold [61]*: Following a similar idea as for the *Global Stepwise Prediction Error Threshold*, the 0.005 percentile is taken of the lower bounds for each time step to make the anomaly detection less error-prone.

We also developed a novel approach using a convex combination of the KL divergence and reconstruction error to detect anomalies. Since it did not lead to any improvement during preliminary experiments we refrain from using it for the remainder of this thesis.

5.2.1.3 *Latent Space Approaches*

After training, all sequential latent variable models provide a representation of the observations in the latent space that can be used to derive an anomaly score.

STORN and DVBF are both trained using variational inference with a Gaussian prior. Thus, after successful training, normal samples from high-density regions of the unknown data distribution $p(\boldsymbol{x}_{1:T})$ are mapped to high-density regions of the prior. Anomalies are by definition rare events and therefore in low-density regions of $p(\boldsymbol{x}_{1:T})$. Anomaly detection methods on the latent space rely on the assumption that once we have learned a model of our data, anomalies will be mapped into low-density regions of the latent space as well, making it possible to distinguish them from normal samples.

However, there are no theoretical guarantees since the models are only trained on normal data. Therefore, it needs to be empirically verified for each dataset. In cases where it does not work, one would have to resort to ELBO-based or reconstruction-based anomaly detection approaches.

We propose two distinct methods to derive an anomaly score using latent space representations:

- One-class classifier which do not rely on anomalous data for training and are only trained on V_{N2}.
- Classifier trained on V_2 which require more data for training but can find better decision boundaries.

Instead of directly using the classification results of these algorithms, we use the output of the classifiers decision function as the anomaly score.

We also found that the latent space approaches work better if we use the mean across time for the latent states as input instead of concatenating them into a single feature vector. This is not necessary for the EncDec model as its latent state has a fixed size.

Possible one-class classifier are *SVM*, *Isolation Forests* [39], and the *Local Outlier Factor* [8]. Classification algorithms include *KNN*, *Decision Trees*, and *Random Forests*. Preliminary experiments have shown that *Local Outlier Factor* and *KNN* generally achieve good performance on our datasets which is why they are further investigated in this thesis. For further details on the algorithms, we refer to Bishop [6] and the respective papers [8, 39].

As for reconstruction-based methods, using the modes of the latent distributions improves the anomaly detection performance for latent space approaches. Therefore, subsequent experiments are all conducted using the mode during inference.

5.2.2 *Evaluation*

Anomaly detection can be seen as a binary classification problem. A common method used to assess their performance is the confusion matrix which splits predictions into True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Using this terminology, several other evaluation metrics can be derived, such as accuracy

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{5.3}$$

and precision and recall

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5.4}$$

To express the anomaly detection performance as a single number, precision and recall can be combined to form the F1-score:

$$\text{F1} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5.5}$$

Accuracy and F1-score only consider the anomaly detection result using a specific threshold on the anomaly scores. This threshold encodes a preference on the trade-off between TPR and FPR.

The ROC curve plots the TPR against the FPR. The Area Under the Receiver Operating Characteristic Curve (ROC-AUC) is thus a measure of performance across all possible thresholds. It represents the probability that a random anomalous sample has a higher anomaly score than a random normal sample. Because of this threshold-invariance we use the ROC-AUC to measure anomaly detection performance in the main body of this thesis on the Paderborn and CMAPSS dataset, while also reporting the other performance metrics in the Appendix.

5.3 PROGNOSTICS

After the hyperparameter search, trained sequential latent variable models can also be applied for prognostics. In the following section, we describe a novel method utilizing latent space representations to predict the RUL and how it is evaluated on the CMAPSS dataset.

5.3.1 *Methods*

Prognostics using sequential latent variable models is inspired by latent space anomaly detection methods. Instead of applying a classifier we use a regression model on the latent space representation to predict the RUL for each sequence.

Similar to anomaly detection on the latent space, we base our approach on the assumption that once we have trained a model, normal data is mapped onto the prior and as the engine starts degrading, the latent space representations will move further away from it. Again, there are no theoretical guarantees as our model is only trained on normal data. Thus, it needs to be confirmed empirically for each dataset whether this assumption holds.

The input to a regression model is a TD-dimensional vector where T is the length of the sequence and D is the dimensionality of the latent space. The output is the RUL for the last time step of the sequence. Similar to other researchers, we decided to model the RUL as a piece-wise linear function for training as it prevents the model from

overestimating it [3, 21]. Heimes [21] empirically found that a steady state of 130 leads to good results:

$$\tilde{\text{RUL}} = \min(\text{RUL}, 130) \tag{5.6}$$

Generally, any regression model can be used to create a mapping between latent state representations and RUL. We decided to use *Linear Regression* because of its simplicity and *Random Forests* which are able to exploit non-linear relationships, but are also more prone to overfitting. For further details on both models, we refer to Bishop [6].

5.3.2 *Evaluation*

Prognostics is a regression problem. Thus, common metrics such as MSE and Root Mean Squared Error (RMSE) can be used to evaluate the performance, whereas the RMSE is most commonly used for the CMAPSS dataset:

$$\text{RMSE} = \sqrt{\sum_i \left(\hat{\text{RUL}}^{(i)} - \text{RUL}^{(i)}\right)^2} \tag{5.7}$$

The RMSE, however, does not take into account the common requirement of prognostics that late predictions should be penalized more than early predictions as they can have fatal effects, especially for turbofan engines. Nevertheless, early predictions should be penalized as well because of their negative economic impact. Therefore, Saxena et al. [55] designed the asymmetric timeliness score which fulfills this requirement and is commonly used to evaluate methods on the CMAPSS dataset:

$$E^{(i)} = \begin{cases} e^{-\frac{d^{(i)}}{13}} - 1 & \text{for } d^{(i)} < 0 \\ e^{\frac{d^{(i)}}{10}} - 1 & \text{for } d^{(i)} \geq 0, \end{cases} \tag{5.8}$$

$$S = \sum_i E^{(i)} \tag{5.9}$$

with $d^{(i)} = \hat{\text{RUL}}^{(i)} - \text{RUL}^{(i)}$.

Drawbacks of the timeliness score are that it depends on the dataset size and a single outlier can dominate the whole score because of its exponential nature. Therefore, we decided to report our results on the CMAPSS dataset using both the RMSE and the timeliness score to allow an easier comparison with existing research and to acknowledge the specific requirements in the area of prognostics.

Table 5.1: Correlation of the ELBO with the performance of different anomaly detection and prognostics methods on the CMAPSS FD001 dataset using STORN and DVBF.

Method	Correlation with ELBO
Global One-Step Pred. (ROC-AUC)	0.09
Local Outlier Factor (ROC-AUC)	−0.06
Lower Bound (ROC-AUC)	0.13
Random Forest (RMSE)	0.13

5.4 MODEL SELECTION

Using the ELBO or MSE for model selection during the hyperparameter search has the advantage that it is a task-independent measure of how well a model fits the data. However, preliminary experiments have shown that the anomaly detection and prognostics performance does not have a significant correlation with the ELBO (cf. Table 5.1).

Also, it is clear that using the RMSE for prognostics or the ROC-AUC for anomaly detection as a criterion during model selection will lead to overall better results on those tasks instead of relying on the loss as a proxy metric.

So far the V_{N1} dataset split which only contains normal data is used for model selection. However, anomalous data is needed as well in order to use these performance metrics. Therefore, we propose an additional split for anomalous data introducing the dataset V_{A1}. The dataset $V_1 = V_{A1} \cup V_{N1}$ is then utilized during the hyperparameter search which allows us to use the performance of any anomaly detection and prognostics method for model selection.

6 RESULTS

This chapter presents the results for anomaly detection on the Paderborn dataset in Section 6.1 and the results for both anomaly detection and prognostics on the CMAPSS dataset in Section 6.2.

6.1 PADERBORN DATASET

The Paderborn dataset is used to evaluate the anomaly detection performance of the methods described in Section 5.2. As described in the previous chapter, first a hyperparameter search on the Paderborn dataset is conducted. Then, the anomaly detection and prognostics performance is evaluated for the best model architectures. Detailed results on the Paderborn dataset can be found in Appendix B.1.

6.1.1 *Training*

The results for the best model architectures found during the hyperparameter search on the Paderborn dataset in terms of loss on the V_{N1} split are listed in Table 6.1. While DVBF has a significantly higher KL divergence, it achieves a lower reconstruction error and better ELBO compared to STORN. A direct comparison with the EncDec model is not possible as it uses a different loss function.

Figure 6.1 shows the results of the hyperparameter search with different hyperparameter configurations using STORN and DVBF. Almost all configurations achieve comparable ELBO values. In general, DVBF achieves lower reconstruction errors with higher KL divergence compared to STORN. Also, there exist several outliers for STORN that have both high reconstruction errors and a high KL divergence.

6.1.2 *Anomaly Detection*

After the hyperparameter search, the respective best model architecture for the EncDec model, STORN, and DVBF are trained on the full training dataset and applied to anomaly detection on the Paderborn dataset.

The ROC curve for each model in combination with the existing and newly introduced anomaly detection methods is depicted in Figure 6.2. It shows that STORN and DVBF significantly outperform the EncDec model using latent space methods whereas DVBF achieved the best results. Examining the first two principal components of the latent space representations depicted in Figure 6.3 gives insights into

Table 6.1: Training results on Paderborn dataset. For STORN and DVBF the ELBO is used as loss function and for the EncDec model the MSE.

Model	KL	Rec. Error	ELBO	MSE
EncDec	-	-	-	0.84
STORN	0.75	0.32	−1.07	-
DVBF	3.18	−2.37	−0.80	-

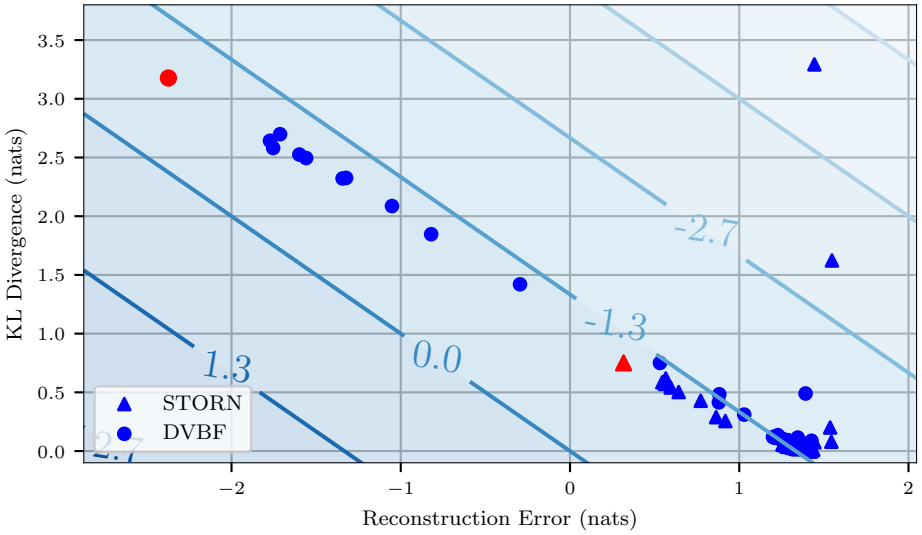


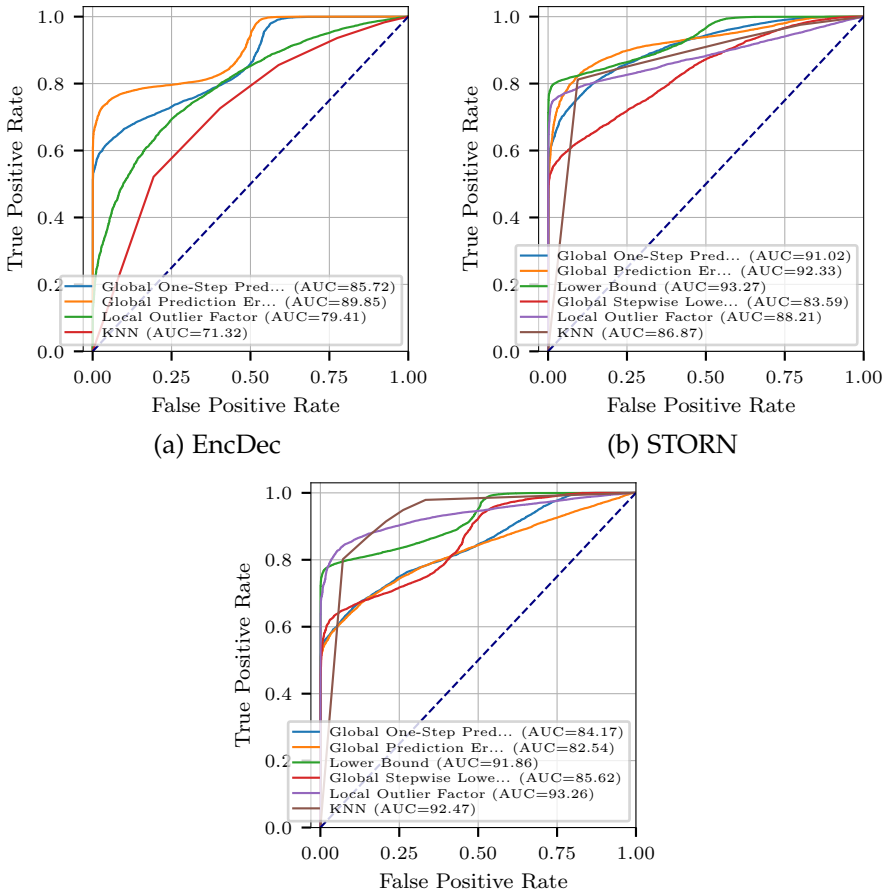
Figure 6.1: Hyperparameter search results of STORN and DVBF on the Paderborn dataset. Each point represents one hyperparameter configuration and the contour lines indicate the respective ELBO value. The best configuration for STORN and DVBF are marked in red.

why this is the case: while there is an overlap between normal and anomalous samples for all models, DVBF and STORN separate them significantly better than the EncDec model.

ELBO-based methods are close to latent space methods in terms of anomaly detection performance and for STORN, the *Lower Bound* even outperforms them and achieves the overall best results. In general, the *Global Stepwise Lower Bound Threshold* obtains a lower ROC-AUC compared to the *Lower Bound*.

For both the EncDec model and STORN, the reconstruction-based methods outperform latent space methods. Also, the *Global Prediction Error Threshold* achieves better results than the *Global One-Step Prediction Error Threshold* except for DVBF. Figure 6.4 shows the reconstructions of a normal and anomalous sequence using STORN. The reconstruction error for the anomalous sequence is significantly higher allowing us to distinguish it from a normal sample.

In summary, the anomaly detection results on the Paderborn dataset show that the performance for most anomaly detection methods differs significantly depending on the used sequential latent variable model.



(a) EncDec (b) STORN

(c) DVBF

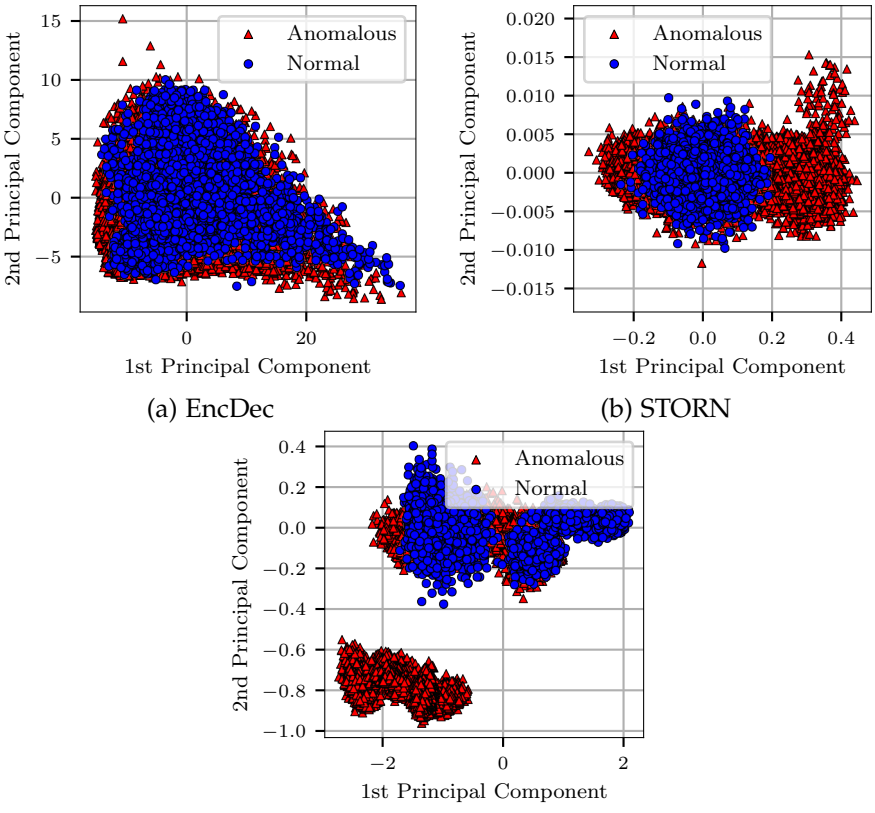
Figure 6.2: Anomaly detection ROC curves on the Paderborn dataset using the loss for model selection.

Therefore, when choosing an approach it is important to choose the right combination of anomaly detection method and sequential latent variable model. Furthermore, the results prove that anomaly detection on latent space representations is a feasible alternative to reconstruction-based and ELBO-based approaches even though it lacks the theoretical guarantees.

6.1.3 *Model Selection*

So far model selection was done based on the loss. As described in Section 5.4, the use of an additional split for anomalous samples allows us to evaluate the performance of anomaly detection methods directly during the hyperparameter search instead of relying on the loss as a proxy measure.

We choose the ROC-AUC obtained using the *Global One-Step Prediction Error Threshold* as a model selection criterion, while in general any method can be used.



(a) EncDec (b) STORN

(c) DVBF

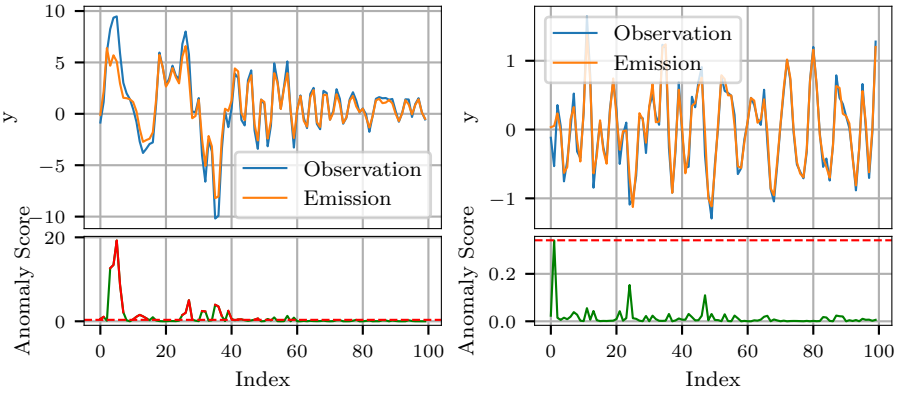
Figure 6.3: PCA projection of mean of latent space representations across time of normal (blue) and anomalous (red) samples for the EncDec model, STORN, and DVBF on the Paderborn dataset.

The results using the best models and both reconstruction-based anomaly detection methods are depicted in Figure 6.5. The performance of each method is either comparable or has improved compared to the earlier results using either the ELBO or MSE depending on the model for selection. Especially for DVBF in combination with the *Global Prediction Error Threshold* the performance has significantly improved.

This proves that selecting model architectures directly based on the anomaly detection metric can improve the results.

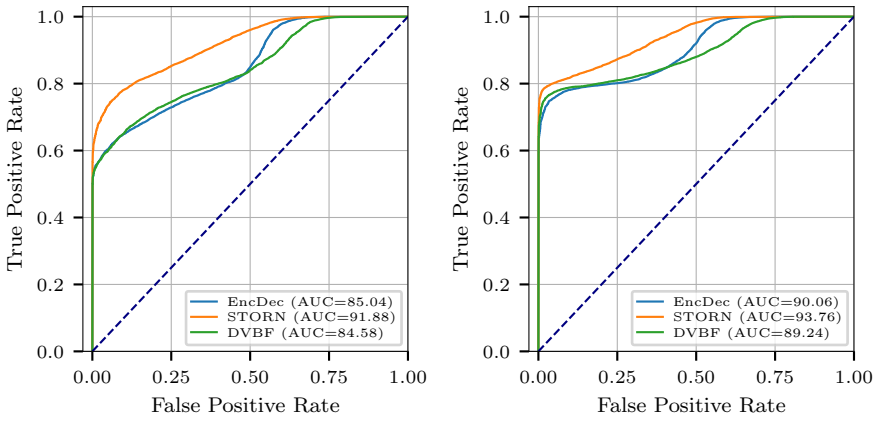
6.2 CMAPSS FD001 DATASET

The CMAPSS FD001 dataset is used to evaluate both the anomaly detection and prognostics performance using sequential latent variable models. As for the Paderborn dataset, first, a hyperparameter search on the CMAPSS FD001 dataset is conducted. Then, the anomaly detection and prognostics performance is evaluated for the best model architectures. Additional results on the CMAPSS datasets can be found in Appendix B.2.



(a) Anomalous (b) Normal

Figure 6.4: Reconstructions of first 100 time steps of sequence using STORN. Anomaly scores are derived using *Global One-Step Prediction Error*. The red dashed line marks the optimal threshold $\hat{\kappa}$ on anomaly scores.



(a) *Global One-Step Pred. Error* (b) *Global Prediction Error Threshold*

Figure 6.5: Anomaly detection ROC curves on the Paderborn dataset using the ROC-AUC of *Global One-Step Prediction Error Threshold* for model selection.

6.2.1 *Training*

Table 6.2 lists the results for the best model architectures found during the hyperparameter search on the CMAPSS FD001 dataset in terms of loss on the V_{N1} split. STORN achieves a lower KL divergence while DVBF has a better reconstruction error and overall a higher ELBO. Again, the EncDec model can not be compared to the others as it is trained using the MSE.

Figure 6.6 shows that STORN and DVBF achieved comparable results in terms of KL divergence and reconstruction error. In general, the choice of hyperparameters has a higher impact on the models' performance compared to the Paderborn dataset where almost all configurations achieved similar ELBO values. The models achieving

Table 6.2: Training results on CMAPSS FD001 dataset. For STORN and DVBF the ELBO is used as loss function and for the EncDec model the MSE.

Model	KL	Rec. Error	ELBO	MSE
EncDec	-	-	-	0.55e−2
STORN	1.12	−9.00	7.88	-
DVBF	1.60	−10.69	9.09	-

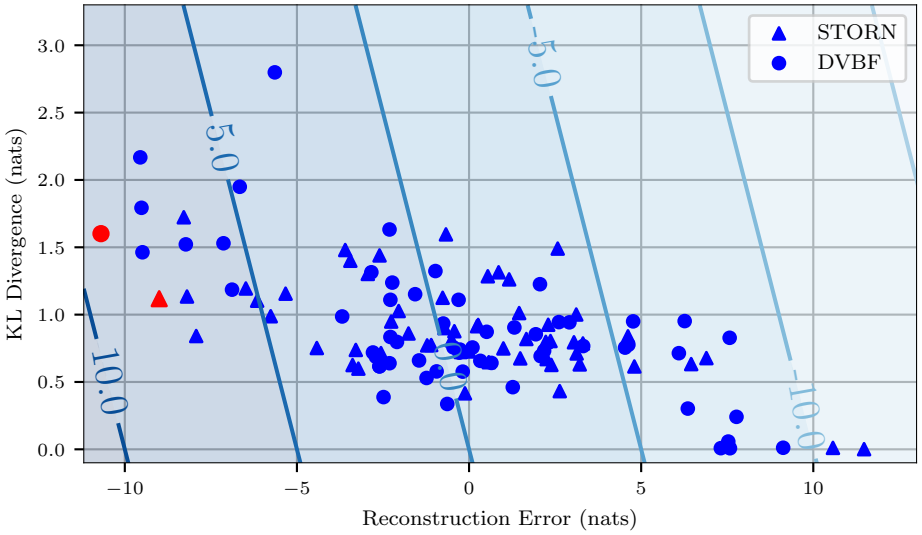


Figure 6.6: Hyperparameter search results of STORN and DVBF on the CMAPSS FD001 dataset. Each point represents one hyperparameter configuration and the contour lines indicate the respective ELBO value. The best configuration for STORN and DVBF are marked in red.

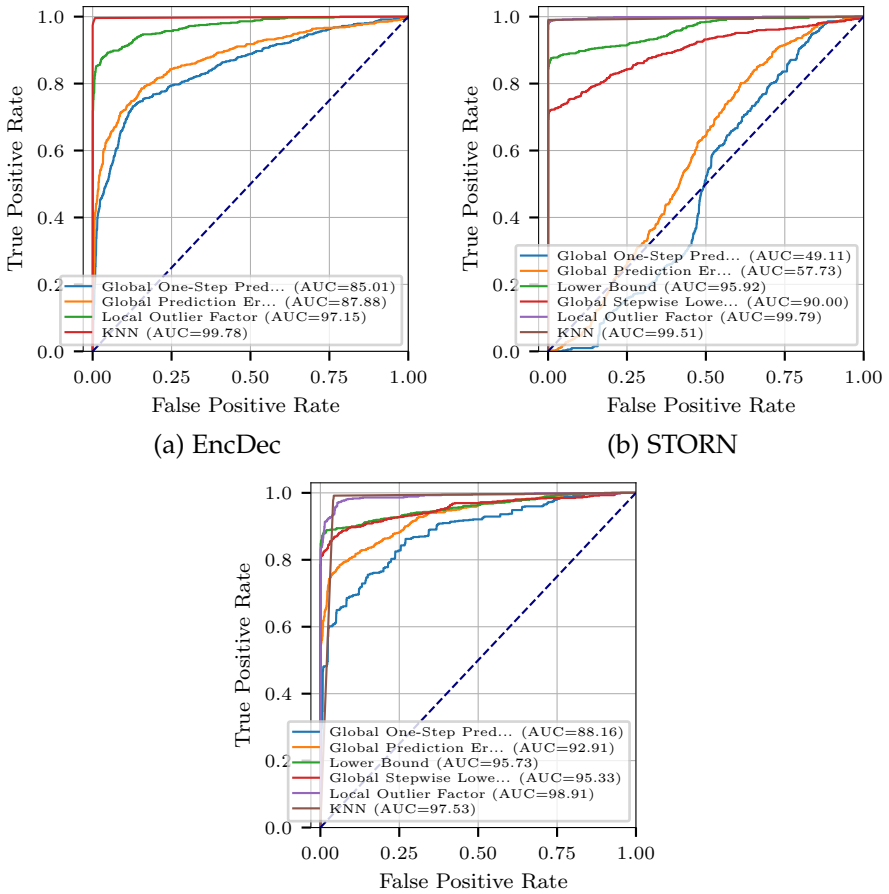
the highest ELBO generally have a lower reconstruction error and a comparably high KL divergence.

6.2.2 *Anomaly Detection*

After training is done, the EncDec model, STORN, and DVBF, with the respective best architecture found during the hyperparameter search, are used for anomaly detection to classify sequences of the CMAPSS dataset as either normal or anomalous.

Figure 6.7 depicts the ROC curve for each anomaly detection method. It shows that *KNN* and *Local Outlier Factor* achieved the best results for all models. Examining the first two principal components of the latent space depicted in Figure 6.8 shows a clear separation between normal and anomalous samples, which explains the good performance using classifiers on the latent space.

The ELBO-based methods achieve slightly worse results than the latent space methods. Using the *Lower Bound* led to better ROC-AUC



(a) EncDec (b) STORN

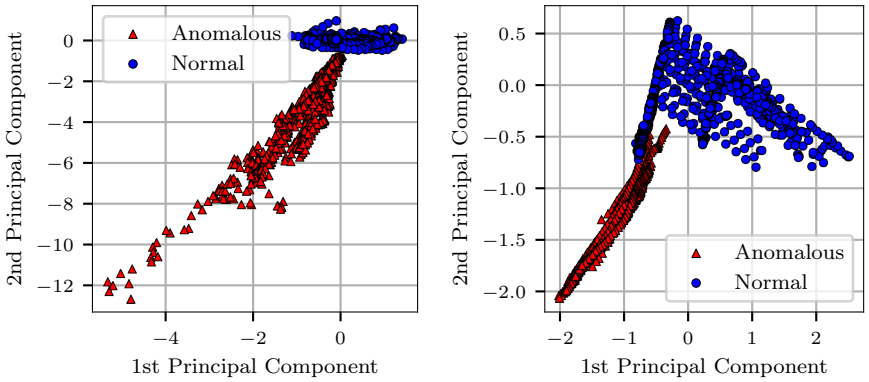
(c) DVBF

Figure 6.7: Anomaly detection ROC curves on the CMAPSS FD001 dataset using the loss for model selection.

scores compared to the *Global Stepwise Lower Bound Threshold* especially for STORN.

In comparison to the Paderborn datasets, the reconstruction-based methods perform considerably worse when compared to other methods and generally have the worst anomaly detection performance. Figure 6.9 shows the output of a temperature sensor and its reconstruction using DVBF. The emission variance is high and there are comparably big error residuals for the whole sequence. This makes it difficult to distinguish between normal and anomalous samples using anomaly scores derived with reconstruction-based methods. Nevertheless, the *Global Prediction Error Threshold* outperforms the *Global One-Step Prediction Error Threshold* proving that summing the error residuals instead of taking percentiles improves the anomaly detection capabilities.

In summary, the results on the CMAPSS FD001 dataset confirm most results gathered on the Paderborn datasets. For both datasets, the performance of an anomaly detector depends both on the anomaly detection method and the chosen model. Also, the *Global Prediction*



(a) STORN (b) DVBF

Figure 6.8: PCA projection of mean of latent space representations across time of normal (blue) and anomalous (red) samples for STORN and DVBF on the CMAPSS FD001 dataset.

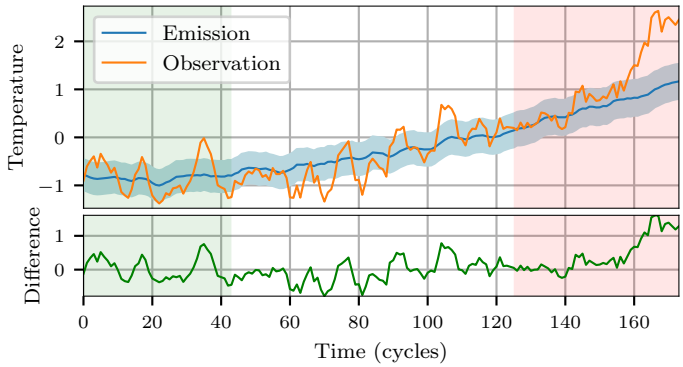


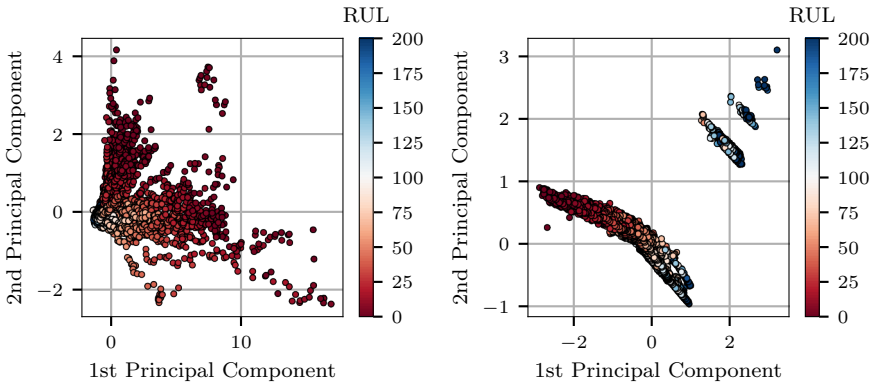
Figure 6.9: Reconstructions of a single sensor using DVBF. The blue shade indicates two standard deviations of the output distribution. Normal regions are marked in green and anomalous ones in red.

Error Threshold consistently outperforms the *Global One-Step Prediction Error Threshold*.

6.2.3 *Prognostics*

The same models with the respective best architecture found during the hyperparameter search are also utilized for prognostics. Figure 6.10 shows the latent space for each model. There is a clear gradient visible from healthy engines to damaged ones indicating that it is possible to infer the RUL from the latent space representation.

The results in Table 6.3 on the CMAPSS FD001 test dataset confirm this as our methods outperform several other approaches in literature (cf. Table 6.6). Among our methods, STORN in combination with *Random Forests* achieved the best RMSE, while DVBF in combination with *Linear Regression* has the best timeliness score. Except for STORN, *Linear Regression* obtains better results than *Random Forests*.



(a) STORN (b) DVBF

Figure 6.10: PCA projection of latent space representation for each time step of samples for STORN and DVBF on CMAPSS FD001 dataset.

Table 6.3: Prognostics results on the CMAPSS FD001 dataset using the loss for model selection.

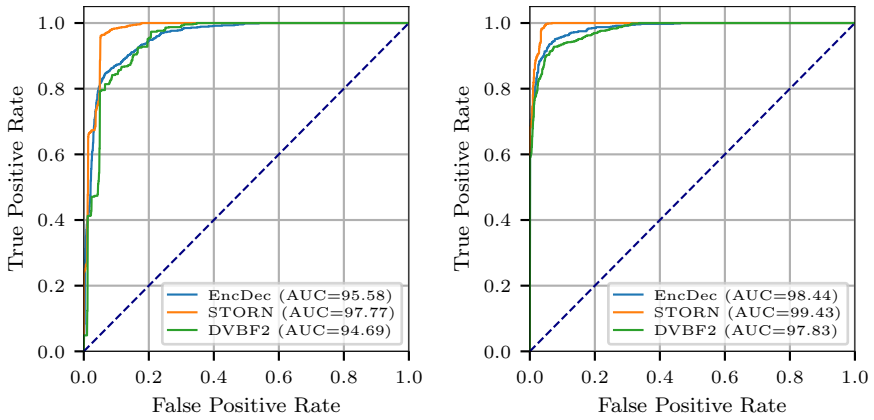
Model	Method	RMSE	Timeliness Score
EncDec	Linear Regression	18.32	1202.13
	Random Forest	20.51	1537.29
STORN	Linear Regression	18.21	997.11
	Random Forest	**18.16**	1252.72
DVBF	Linear Regression	18.51	**858.84**
	Random Forest	20.34	1242.52

Given these results, we conclude that even though the sequential latent variable models are only trained on normal samples it is possible to infer the RUL from the latent space.

6.2.4 *Model Selection*

As for the Paderborn dataset, we choose the ROC-AUC obtained using the *Global One-Step Prediction Error Threshold* as a second model selection criterion to verify whether the anomaly detection performance can be improved.

The results using the best models and both reconstruction-based anomaly detection methods are depicted in Figure 6.11. It shows that the performance has significantly improved. STORN in combination with the *Global Prediction Error Threshold* method achieved results similar to latent space methods which previously clearly outperformed reconstruction-based methods. Also, the relative improvement to previous results is considerably higher compared to the Paderborn dataset.



(a) *Global One-Step Pred. Error* (b) *Global Prediction Error Threshold*

Figure 6.11: Anomaly detection ROC curves on the CMAPSS FD001 dataset using the ROC-AUC of *Global One-Step Prediction Error Threshold* for model selection.

Table 6.4: Prognostics results on the CMAPSS FD001 dataset using the RMSE obtained by *Linear Regression* for model selection.

Model	Method	RMSE	Timeliness Score
EncDec	Linear Regression	19.89	1684.91
STORN	Linear Regression	**17.10**	**635.47**
DVBF	Linear Regression	17.77	879.08

We conducted a similar experiment for prognostics and used the performance of *Linear Regression* in terms of RMSE for model selection. Table 6.4 lists the results, which show that, except for the EncDec model, the performance using the RMSE of *Linear Regression* for model selection is better for both metrics.

This proves that the performance for both anomaly detection and prognostics can be improved significantly using different metrics instead of the loss for model selection.

6.3 CMAPSS DATASETS

While the CMAPSS FD001 dataset is the most popular dataset in literature [51], the other CMAPSS datasets can be used as well to evaluate both anomaly detection and prognostics methods. It is expected that our methods perform slightly worse on these datasets as they include more fault types and operating conditions which aggravates anomaly detection and prognostics. Additional results on the CMAPSS datasets can be found in Appendix B.2.

Table 6.5: Anomaly detection results on the CMAPSS FD001-FD004 datasets using the loss for model selection and ROC-AUC as evaluation metric.

Model	Method	FD001	FD002	FD003	FD004
EncDec	Global One-Step Pred.	85.01	95.10	96.54	95.08
	Global Prediction Error	87.88	96.29	97.67	96.73
	Local Outlier Factor	97.15	81.56	97.65	88.75
	KNN	99.78	83.59	**99.81**	88.41
STORN	Global One-Step Pred.	49.11	93.59	94.25	94.87
	Global Prediction Error	57.73	97.00	96.34	98.52
	Lower Bound	95.92	97.20	98.41	98.52
	Global Stepwise LB	90.00	97.82	96.63	99.16
	Local Outlier Factor	**99.79**	96.79	98.87	92.24
	KNN	99.51	96.57	98.89	95.52
DVBF	Global One-Step Pred.	88.16	97.42	87.51	79.45
	Global Prediction Error	92.91	98.30	92.71	86.51
	Lower Bound	95.73	**99.48**	98.47	98.45
	Global Stepwise LB	95.33	99.17	97.98	**99.80**
	Local Outlier Factor	98.91	50.81	98.96	98.89
	KNN	97.53	51.68	98.13	97.28

6.3.1 *Anomaly Detection*

We performed a hyperparameter search as described in Section 5.1 on each of the CMAPSS datasets and subsequently evaluated the anomaly detection performance of the best model architectures. Table 6.5 lists the ROC-AUC of all anomaly detection methods on each of the CMAPSS datasets.

In general, ELBO-based approaches achieved the best results for the FD002 and FD004 dataset with six operating conditions while for the FD001 and FD003 dataset with only one operating condition the latent space approaches outperformed them. Especially the EncDec model and DVBF have issues inferring a useful latent space for anomaly detection on the CMAPSS FD002 dataset. For both models, the performance of the latent space approaches drops significantly compared to other methods.

Again, the *Global Prediction Error Threshold* outperforms the *Global One-Step Prediction Error Threshold* for all models and datasets.

Given these results, we conclude that ELBO-based methods systematically outperform reconstruction based methods on all datasets while latent space methods achieved the best results on datasets with only

one operating condition. Furthermore, STORN and DVBF have a significant advantage over the EncDec model as ELBO-based methods are superior to the other methods on the FD002 and FD004 dataset. The results furthermore show that the performance of a specific anomaly detection method varies significantly between different sequential latent variable models.

6.3.2 *Prognostics*

We also conducted prognostics experiments on all CMAPSS datasets using the best model architectures found during the hyperparameter search. Since *Linear Regression* generally achieved better results on the CMAPSS FD001 dataset, we directly evaluate our prognostics approach using the RMSE achieved by *Linear Regression* as model selection criterion during the hyperparameter search. The results for using the loss as selection criterion can be found in Appendix B.2.

The RMSE obtained by our methods and other approaches in literature are listed in Table 6.6. Again, *Linear Regression* outperforms *Random Forests* in most cases.

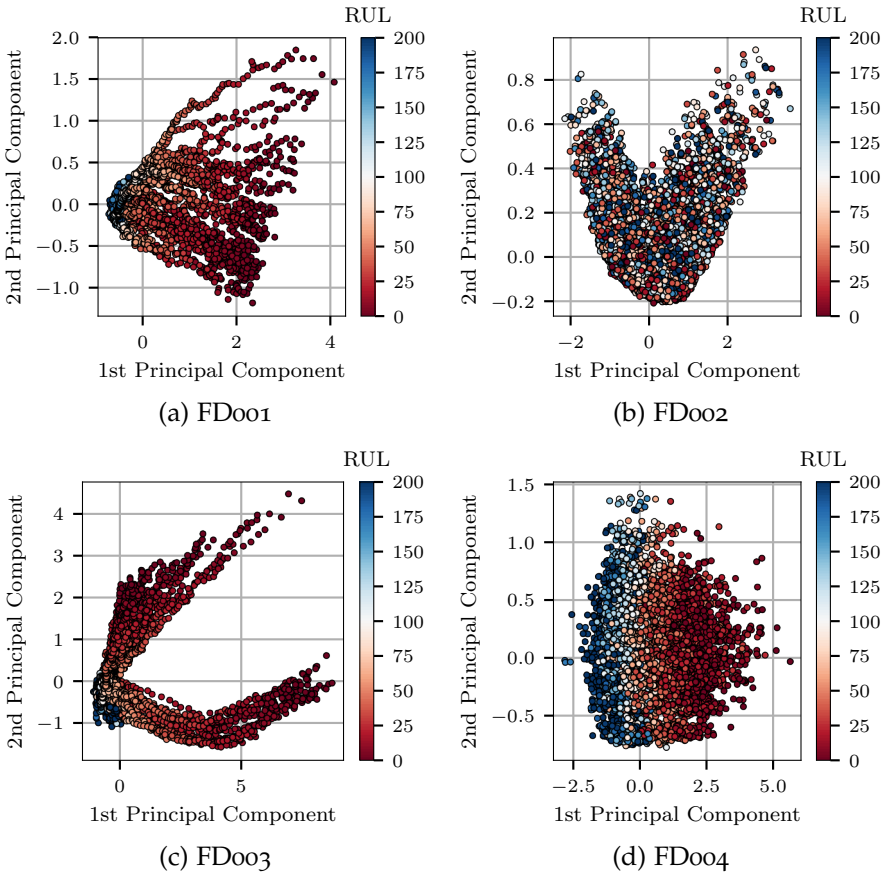
The performance on different datasets significantly differs, as was already the case for anomaly detection with latent space methods. Figure 6.12 shows the latent space of STORN on the four CMAPSS datasets. There is a significant difference in how well the latent space reflects the RUL depending on the number of operating conditions. For the FD001 and FD003 dataset, which only use one operating condition, there is a clear gradient visible in the latent space. However, for the other two datasets with six operating conditions and especially the FD002 dataset, there is no structure recognizable, which is also reflected in the prognostics performance.

The latent space for the FD003 dataset has two distinctive trajectories. As the dataset has two different types of faults (compared to the FD001 dataset with only one type), it is a reasonable assumption that the fault type is reflected in the latent space. However, this can not be verified as no ground truth is available.

Overall, our semi-supervised methods are superior to several supervised approaches such as Babu et al. [3] and Wu et al. [68]. However, there is still a significant gap between our results and the state-of-the-art results of Ellefsen et al. [15] and Gugulothu et al. [20].

Table 6.6: Prognostics RMSE results from methods proposed in literature and methods proposed in this thesis using the RMSE obtained by *Linear Regression* for model selection on the CMAPSS FD001-FD004 datasets.

Method	FD001	FD002	FD003	FD004
EncDec + Linear Regression	19.89	30.05	20.35	32.60
EncDec + Random Forest	20.77	31.36	22.95	35.23
STORN + Linear Regression	17.10	31.06	17.56	31.67
STORN + Random Forest	19.28	37.35	19.10	33.03
DVBF + Linear Regression	17.77	34.44	31.81	30.54
DVBF + Random Forest	21.93	43.15	23.82	29.69
MLP [3]	37.56	80.03	37.39	77.37
SVR [3]	20.96	42.00	21.05	45.35
CNN [3]	18.45	30.29	19.82	29.16
SVM [41]	29.82	-	-	-
DLSTM [68]	18.33	-	19.78	-
LSTM [70]	16.14	24.49	16.18	28.17
LSTM [24]	16.74	29.43	18.07	28.40
CNN [38]	12.61	**22.36**	12.64	23.31
Hybrid CNN and LSTM [1]	22.13	36.35	23.79	-
RBM and LSTM [15]	12.56	22.73	**12.10**	**22.66**
EncDec [20]	**12.45**	-	-	-



(a) FD001 (b) FD002

(c) FD003 (d) FD004

Figure 6.12: PCA projection of latent space representation for each time step of samples for STORN on all CMAPSS datasets.

7 CONCLUSION

In this thesis, we proposed a novel method using sequential latent variable models such as STORN to perform the two PHM tasks of anomaly detection and prognostics on sequential data. In contrast to existing approaches that only focused on either anomaly detection or prognostics, our approach can be universally applied to both problems. It is semi-supervised and the model is only trained on data from healthy components, which is significantly easier to record compared to anomalous data. Also, since our method solely relies on the output of the sequential latent variable models for both anomaly detection and prognostics, it can be easily adapted to different datasets as no domain knowledge is required.

In a previous study, Soelch et al. [61] used ELBO- and reconstruction-based methods in combination with STORN for anomaly detection. We extended this methodology by introducing latent space methods that detect anomalies based on latent space representations. They achieved superior results on the CMAPSS FD001 dataset and performed only marginally worse than the best method on the Paderborn dataset. Furthermore, we introduced the *Global Prediction Error Threshold* as an improvement to the *Global One-Step Prediction Error Threshold* which outperformed the latter in all but one of our benchmarks.

Our results proved that anomaly detection and prognostics using sequential latent variable models are not restricted to STORN but also applicable with DVBF. Furthermore, sequential latent variable models generally outperformed the deterministic EncDec model which, for several cases, can be attributed to the availability of ELBO-based methods.

We also found that the loss is not an optimal criterion for model selection and showed that other metrics can lead to significant performance gains for both anomaly detection and prognostics.

There has already been research by Soelch et al. [61] into applying sequential latent variable models for anomaly detection. Using them for prognostics is a less investigated problem. Nevertheless, the methods used in this thesis outperformed several supervised NN-based approaches, but they still fall short of the state-of-the-art methods in the field.

Therefore, we envision the transfer of proven methods in the area of prognostics to sequential latent variable models. Similar to anomaly detection, the use of ELBO- or reconstruction-based approaches for prognostics can be investigated and compared to latent space methods. Another promising approach [20] that led to state-of-the-art results

maps the latent space representation of an EncDec model to a health index. The index is then used to find the closest existing trajectory using similarity matching instead of directly predicting the RUL. This approach can be easily adapted to sequential latent variable models as it only requires meaningful latent space representations.

In general, any sequential latent variable model can be used for both PHM tasks using our framework. Thus, another future research direction would be to compare other models such as the Deep Kalman Filter [32], the Stochastic RNN [16] or the Variational RNN [13] to the models used in this thesis.

Finally, the presented approaches can be extended to fault diagnostics which is also part of the PHM discipline. Latent space methods are an especially promising approach since different types of faults are likely to cluster in the latent space as was already shown on the CMAPSS FD003 dataset. Adding fault diagnostics to our method would then lead to a fully universal approach, as it would allow us to detect a fault, diagnose it, and finally predict a systems RUL.

A

HYPERPARAMETER SEARCH SPACE

This section lists all hyperparameters and their range of values for each model used during hyperparameter search as described in Section 5.1.

For detailed information on the weight initialization schemes, namely Orthogonal initialization [54] and Xavier initialization [17], we refer to the respective papers.

Table A.1: Hyperparameter search space for the EncDec model

Hyperparameter	Range of Values
Latent Dimensionality	32, 64, 128, 256, 512
Activation Function	softsign, softplus, ReLU
Recurrent Cell Type	LSTM, GRU
Weight Initialization	Xavier, Orthogonal
Optimization Algorithm	Adam
Learning Rate	1e−2, 1e−3, 1e−4, 1e−5
β_1	0.5, 0.8, 0.9, 1.0

Table A.2: Hyperparameter search space for STORN

Hyperparameter	Range of Values
Latent Dimensionality	1, 4, 16, 32, 64
Dropout Rate	0, 0.1, 0.2, 0.3, 0.4, 0.5
Activation Function	softsign, softplus, ReLU
Recurrent Cell Type	LSTM, GRU
Weight Initialization	Xavier, Orthogonal
Bidirectional Recognition Model	
Hidden Dimensionality	16, 32, 64, 128
Number of Layers	1, 2, 3
Unidirectional Generating Model	
Hidden Dimensionality	16, 32, 64, 128
Number of Layers	1, 2, 3
Optimization Algorithm	Adam
Learning Rate	1e−2, 1e−3, 1e−4, 1e−5
β_1	0.5, 0.8, 0.9, 1.0

Table A.3: Hyperparameter search space for DVBF

Hyperparameter	Range of Values
Latent Dimensionality	1, 4, 16, 32, 64
Activation Function	softsign, softplus, ReLU
Weight Initialization	Xavier, Orthogonal
Initial Sequence Length	4, 8, 16, 32
Inverse Measurement Model	
Hidden Dimensionality	16, 32, 64, 128
Number of Layers	1, 2, 3
Emission Model	
Hidden Dimensionality	16, 32, 64, 128
Number of Layers	1, 2, 3
Transition Model	
Hidden Dimensionality	16, 32, 64, 128
Number of Layers	1, 2, 3
Initial Inference Model	
Hidden Dimensionality	16, 32, 64, 128
Number of Layers	1, 2, 3
Initial Transition Model	
Hidden Dimensionality	16, 32, 64, 128
Number of Layers	1, 2, 3
Optimization Algorithm	Adam
Learning Rate	1e−2, 1e−3, 1e−4, 1e−5
β_1	0.5, 0.8, 0.9, 1.0

B

ADDITIONAL RESULTS

Additional results for both the Paderborn and CMAPSS dataset are presented in this chapter to facilitate comparison with new algorithms.

B.1 PADERBORN DATASET

This section lists the results of the models with the respective best architecture on the Paderborn dataset for anomaly detection using the loss as model selection criterion during the hyperparameter search.

Table B.1: Anomaly detection results on the Paderborn dataset using the loss for model selection.

Model	Method	ROC AUC	Acc.	F1	Rec.	Prec.
EncDec	Global One-Step Pred.	85.72	0.92	0.96	0.98	0.94
	Global Pred. Error	89.85	0.92	0.95	0.97	0.94
	Local Outlier Factor	79.41	0.73	0.83	0.74	0.95
	KNN	71.32	0.55	0.68	0.52	0.96
STORN	Global One-Step Pred.	91.02	0.89	0.94	0.92	0.95
	Global Pred. Error	92.33	0.89	0.93	0.90	0.97
	Lower Bound	93.27	0.86	0.92	0.87	0.96
	Global Stepwise LB	83.59	0.69	0.79	0.67	0.97
	Local Outlier Factor	88.21	0.81	0.89	0.82	0.97
	KNN	86.87	0.82	0.89	0.81	0.99
DVBF	Global One-Step Pred.	84.17	0.82	0.89	0.86	0.93
	Global Pred. Error	82.54	0.79	0.88	0.82	0.94
	Lower Bound	91.86	0.91	0.95	0.96	0.94
	Global Stepwise LB	85.62	0.70	0.80	0.68	0.98
	Local Outlier Factor	93.26	0.88	0.93	0.88	0.98
	KNN	92.47	0.82	0.89	0.80	0.99

B.2 CMAPSS DATASETS

This section lists the results of the models with the respective best architecture found during the hyperparameter search on each CMAPSS dataset for anomaly detection and prognostics using the loss as model selection criterion.

Table B.2: Anomaly detection results on the CMAPSS FD001 dataset using the loss for model selection.

Model	Method	ROC AUC	Acc.	F1	Rec.	Prec.
EncDec	Global One-Step Pred.	85.01	0.82	0.74	0.73	0.74
	Global Pred. Error	87.88	0.83	0.75	0.75	0.75
	Local Outlier Factor	97.15	0.90	0.86	0.91	0.81
	KNN	99.78	0.99	0.99	0.98	1.00
STORN	Global One-Step Pred.	49.11	0.50	0.38	0.45	0.32
	Global Pred. Error	57.73	0.56	0.47	0.59	0.39
	Lower Bound	95.92	0.95	0.92	0.88	0.96
	Global Stepwise LB	90.00	0.87	0.80	0.74	0.86
	Local Outlier Factor	99.79	0.99	0.99	0.98	1.00
	KNN	99.51	1.00	0.99	0.99	1.00
DVBF	Global One-Step Pred.	88.16	0.82	0.73	0.72	0.74
	Global Pred. Error	92.91	0.88	0.82	0.79	0.84
	Lower Bound	95.73	0.95	0.92	0.88	0.96
	Global Stepwise LB	95.33	0.91	0.87	0.89	0.85
	Local Outlier Factor	98.91	0.95	0.93	0.96	0.90
	KNN	97.53	0.97	0.95	0.98	0.92

Table B.3: Prognostics results on the CMAPSS FD001 dataset using the loss for model selection.

Model	Method	RMSE	Timeliness Score
EncDec	Linear Regression	18.32	1202.13
	Random Forest	20.51	1537.29
STORN	Linear Regression	18.21	997.11
	Random Forest	18.16	1252.72
DVBF	Linear Regression	18.51	858.84
	Random Forest	20.34	1242.52

Table B.4: Anomaly detection results on the CMAPSS FD002 dataset using the loss for model selection.

Model	Method	ROC AUC	Acc.	F1	Rec.	Prec.
EncDec	Global One-Step Pred.	95.10	0.89	0.85	0.88	0.81
	Global Pred. Error	96.29	0.91	0.87	0.89	0.86
	Local Outlier Factor	81.56	0.74	0.65	0.73	0.59
	KNN	83.59	0.81	0.70	0.66	0.75
STORN	Global One-Step Pred.	93.59	0.88	0.82	0.81	0.83
	Global Pred. Error	97.00	0.93	0.89	0.91	0.88
	Lower Bound	97.20	0.92	0.89	0.90	0.88
	Global Stepwise LB	97.82	0.94	0.91	0.92	0.90
	Local Outlier Factor	96.79	0.88	0.84	0.93	0.76
	KNN	96.57	0.95	0.93	0.92	0.93
DVBF	Global One-Step Pred.	97.42	0.92	0.88	0.85	0.90
	Global Pred. Error	98.30	0.94	0.91	0.94	0.88
	Lower Bound	99.48	0.97	0.96	0.95	0.97
	Global Stepwise LB	99.17	0.98	0.97	0.96	0.98
	Local Outlier Factor	50.81	0.46	0.43	0.61	0.34
	KNN	51.68	0.50	0.42	0.53	0.34

Table B.5: Prognostics results on the CMAPSS FD002 dataset using the loss for model selection.

Model	Method	RMSE	Timeliness Score
EncDec	Linear Regression	30.14	18 386.69
	Random Forest	30.51	16 933.04
STORN	Linear Regression	32.05	22 686.59
	Random Forest	32.28	36 686.80
DVBF	Linear Regression	53.49	1 267 592.35
	Random Forest	44.40	181 316.43

Table B.6: Anomaly detection results on the CMAPSS FD003 dataset using the loss for model selection.

Model	Method	ROC AUC	Acc.	F1	Rec.	Prec.
EncDec	Global One-Step Pred.	96.54	0.88	0.81	0.88	0.76
	Global Pred. Error	97.67	0.90	0.84	0.90	0.78
	Local Outlier Factor	97.65	0.89	0.84	0.95	0.75
	KNN	99.81	0.99	0.98	0.99	0.96
STORN	Global One-Step Pred.	94.25	0.88	0.81	0.86	0.77
	Global Pred. Error	96.34	0.91	0.85	0.90	0.81
	Lower Bound	98.41	0.97	0.95	0.94	0.96
	Global Stepwise LB	96.63	0.97	0.95	0.91	0.99
	Local Outlier Factor	98.87	0.93	0.89	0.97	0.82
	KNN	98.89	0.99	0.98	0.98	0.99
DVBF	Global One-Step Pred.	87.51	0.75	0.67	0.84	0.56
	Global Pred. Error	92.71	0.82	0.75	0.89	0.64
	Lower Bound	98.47	0.98	0.96	0.96	0.97
	Global Stepwise LB	97.98	0.93	0.89	0.94	0.85
	Local Outlier Factor	98.96	0.94	0.90	0.97	0.84
	KNN	98.13	0.97	0.95	0.99	0.91

Table B.7: Prognostics results on the CMAPSS FD003 dataset using the loss for model selection.

Model	Method	RMSE	Timeliness Score
EncDec	Linear Regression	19.56	1033.05
	Random Forest	21.04	2773.08
STORN	Linear Regression	21.60	1597.40
	Random Forest	21.69	2108.74
DVBF	Linear Regression	19.29	814.20
	Random Forest	17.69	668.40

Table B.8: Anomaly detection results on the CMAPSS FD004 dataset using the loss for model selection.

Model	Method	ROC AUC	Acc.	F1	Rec.	Prec.
EncDec	Global One-Step Pred.	95.08	0.89	0.80	0.88	0.73
	Global Pred. Error	96.73	0.92	0.85	0.90	0.80
	Local Outlier Factor	88.75	0.82	0.68	0.80	0.59
	KNN	88.41	0.90	0.76	0.69	0.86
STORN	Global One-Step Pred.	94.87	0.90	0.81	0.87	0.75
	Global Pred. Error	98.52	0.95	0.90	0.93	0.87
	Lower Bound	98.52	0.95	0.90	0.96	0.85
	Global Stepwise LB	99.16	0.95	0.91	0.97	0.86
	Local Outlier Factor	92.24	0.83	0.72	0.86	0.61
	KNN	95.52	0.93	0.85	0.90	0.81
DVBF	Global One-Step Pred.	79.45	0.71	0.60	0.90	0.45
	Global Pred. Error	86.51	0.81	0.69	0.85	0.57
	Lower Bound	98.45	0.94	0.88	0.94	0.84
	Global Stepwise LB	99.80	0.98	0.96	0.98	0.94
	Local Outlier Factor	98.89	0.92	0.85	0.97	0.75
	KNN	97.28	0.97	0.93	0.92	0.95

Table B.9: Prognostics results on the CMAPSS FD004 dataset using the loss for model selection.

Model	Method	RMSE	Timeliness Score
EncDec	Linear Regression	31.90	9424.69
	Random Forest	33.23	16 218.05
STORN	Linear Regression	32.47	10 482.11
	Random Forest	33.99	13 696.99
DVBF	Linear Regression	30.06	8123.90
	Random Forest	31.16	9458.96

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

[1] Khaled Akkad and David He. "A Hybrid Deep Learning Based Approach for Remaining Useful Life Estimation." In: *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2019, pp. 1–6.

[2] Jinwon An and Sungzoon Cho. "Variational autoencoder based anomaly detection using reconstruction probability." In: *Special Lecture on IE* 2.1 (2015).

[3] Giduthuri Sateesh Babu, Peilin Zhao, and Xiao-Li Li. "Deep convolutional neural network based regression approach for estimation of remaining useful life." In: *International conference on database systems for advanced applications*. Springer. 2016, pp. 214–228.

[4] Justin Bayer and Christian Osendorfer. "Learning Stochastic Recurrent Networks." In: *NIPS 2014 Workshop on Advances in Variational Inference*. 2014.

[5] E Bechhoefer. *Condition based maintenance fault database for testing of diagnostic and prognostics algorithms*. 2013.

[6] Christopher M Bishop. "Machine learning and pattern recognition." In: *Information science and statistics. Springer, Heidelberg* (2006).

[7] Hervé Bourlard and Yves Kamp. "Auto-association by multilayer perceptrons and singular value decomposition." In: *Biological cybernetics* 59.4-5 (1988), pp. 291–294.

[8] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. "LOF: identifying density-based local outliers." In: *ACM sigmod record*. Vol. 29. 2. ACM. 2000, pp. 93–104.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[10] Sucheta Chauhan and Lovekesh Vig. "Anomaly detection in ECG time signals via deep long short-term memory networks." In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2015, pp. 1–7.

[11] Min Cheng, Qian Xu, Jianming Lv, Wenyin Liu, Qing Li, and Jianping Wang. "MS-LSTM: A multi-scale LSTM model for BGP anomaly detection." In: *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE. 2016, pp. 1–6.

[12] Kyunghyun Cho, B van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. "On the properties of neural machine translation: Encoder-decoder approaches." In: *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8), 2014*. 2014.

[13] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. "A recurrent latent variable model for sequential data." In: *Advances in neural information processing systems*. 2015, pp. 2980–2988.

[14] Ömer Faruk Eker, Faith Camci, and Ian K Jennions. *Major challenges in prognostics: study on benchmarking prognostic datasets*. 2012.

[15] André Listou Ellefsen, Emil Bjørlykhaug, Vilmar Æsøy, Sergey Ushakov, and Houxiang Zhang. "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture." In: *Reliability Engineering & System Safety* 183 (2019), pp. 240–251.

[16] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. "Sequential neural models with stochastic layers." In: *Advances in neural information processing systems*. 2016, pp. 2199–2207.

[17] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010, pp. 249–256.

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

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[20] Narendhar Gugulothu, Vishnu TV, Pankaj Malhotra, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. "Predicting remaining useful life using time series embeddings based on recurrent neural networks." In: *arXiv preprint arXiv:1709.01073* (2017).

[21] Felix O Heimes. "Recurrent neural networks for remaining useful life estimation." In: *2008 international conference on prognostics and health management*. IEEE. 2008, pp. 1–6.

[22] Geoffrey E Hinton and Ruslan R Salakhutdinov. "Reducing the dimensionality of data with neural networks." In: *science* 313.5786 (2006), pp. 504–507.

[23] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[24] Che-Sheng Hsu and Jehn-Ruey Jiang. "Remaining useful life estimation using long short-term memory deep learning." In: *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE. 2018, pp. 58–61.

[25] IS ISO13372. *Condition monitoring and diagnostics of machines-Vocabular*. 2004.

[26] Georgios Karatzinis, Yiannis S Boutalis, and Yannis L Karnavas. "Motor Fault Detection and Diagnosis Using Fuzzy Cognitive Networks with Functional Weights." In: *2018 26th Mediterranean Conference on Control and Automation (MED)*. IEEE. 2018, pp. 709–714.

[27] Maximilian Karl, Philip Becker-Ehmck, Maximilian Soelch, Djalel Benbouzid, Patrick van der Smagt, and Justin Bayer. "Unsupervised real-time control through variational empowerment." In: *International Symposium on Robotics Research* (2019).

[28] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. "Deep variational bayes filters: Unsupervised learning of state space models from raw data." In: *International Conference on Learning Representations* (2017).

[29] Jihyun Kim, Jaehyun Kim, Huong Le Thi Thu, and Howon Kim. "Long short term memory recurrent neural network classifier for intrusion detection." In: *2016 International Conference on Platform Technology and Service (PlatCon)*. IEEE. 2016, pp. 1–5.

[30] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014).

[31] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes." In: *arXiv preprint arXiv:1312.6114* (2013).

[32] Rahul G. Krishnan, Uri Shalit, and David Sontag. "Deep Kalman Filters." In: *arXiv preprint arXiv:1511.05121* (2015).

[33] Alexander Lavin and Subutai Ahmad. "Evaluating Real-Time Anomaly Detection Algorithms–The Numenta Anomaly Benchmark." In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2015, pp. 38–44.

[34] J Lee, H Qiu, G Yu, and J Lin. *Bearing Data Set, IMS, University of Cincinnati*.

[35] Yaguo Lei, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. "Machinery health prognostics: A systematic review from data acquisition to RUL prediction." In: *Mechanical Systems and Signal Processing* 104 (2018), pp. 799–834.

[36] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. "Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification." In: *Proceedings of the European conference of the prognostics and health management society*. 2016, pp. 05–08.

[37] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. "Anomaly detection with generative adversarial networks for multivariate time series." In: *arXiv preprint arXiv:1809.04758* (2018).

[38] Xiang Li, Qian Ding, and Jian-Qiao Sun. "Remaining useful life estimation in prognostics using deep convolution neural networks." In: *Reliability Engineering & System Safety* 172 (2018), pp. 1–11.

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

[40] KA Loparo. *Case western reserve university bearing data center*. 2012.

[41] C Louen, SX Ding, and C Kandler. "A new framework for remaining useful life estimation using support vector machine classifier." In: *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE. 2013, pp. 228–233.

[42] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. "LSTM-based encoder-decoder for multi-sensor anomaly detection." In: *arXiv preprint arXiv:1607.00148* (2016).

[43] Pankaj Malhotra, Vishnu TV, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. "Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder." In: *arXiv preprint arXiv:1608.06154* (2016).

[44] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. "Long short term memory networks for anomaly detection in time series." In: *Proceedings*. Presses universitaires de Louvain. 2015, p. 89.

[45] Arjovsky Martin and Lon Bottou. "Towards principled methods for training generative adversarial networks." In: *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*. Vol. 2016. 2017.

[46] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[47] Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Noureddine Zerhouni, and Christophe Varnier. "PRONOSTIA: An experimental platform for bearings accelerated degradation tests." In: *IEEE International Conference on Prognostics and Health Management, PHM'12.* IEEE Catalog Number: CPF12PHM-CDR. 2012, pp. 1–8.

[48] Timothy J O'Shea, T Charles Clancy, and Robert W McGwier. "Recurrent neural radio anomaly detection." In: *arXiv preprint arXiv:1611.00301* (2016).

[49] Khary I Parker and Ten-Heui Guo. *Development of a turbofan engine simulation in a graphical simulation environment*. 2003.

[50] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. "A review of novelty detection." In: *Signal Processing* 99 (2014), pp. 215–249.

[51] Emmanuel Ramasso and Abhinav Saxena. *Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets.* 2014.

[52] Emmanuel Ramasso and Abhinav Saxena. *Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset*. 2014.

[53] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models." In: *International Conference on Machine Learning*. 2014, pp. 1278–1286.

[54] Andrew M Saxe, James L McClelland, and Surya Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks." In: *arXiv preprint arXiv:1312.6120* (2013).

[55] A Saxena and K Goebel. "Phm08 challenge data set." In: *NASA Ames Prognostics Data Repository*. NASA Ames Research Center, 2008.

[56] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. "Damage propagation modeling for aircraft engine run-to-failure simulation." In: *2008 international conference on prognostics and health management*. IEEE. 2008, pp. 1–9.

[57] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery." In: *International Conference on Information Processing in Medical Imaging*. Springer. 2017, pp. 146–157.

[58] Insun Shin, Junmin Lee, Jun Young Lee, Kyusung Jung, Daeil Kwon, Byeng D Youn, Hyun Soo Jang, and Joo-Ho Choi. "A framework for prognostics and health management applications toward smart manufacturing systems." In: *International Journal of Precision Engineering and Manufacturing-Green Technology* 5.4 (2018), pp. 535–554.

[59] Hava T Siegelmann and Eduardo D Sontag. "On the computational power of neural nets." In: *Proceedings of the fifth annual workshop on Computational learning theory*. ACM. 1992, pp. 440–449.

[60] Maximilian Soelch. *Detecting anomalies in robot time series data using stochastic recurrent networks*. 2015.

[61] Maximilian Sölch, Justin Bayer, Marvin Ludersdorfer, and Patrick van der Smagt. "Variational inference for on-line anomaly detection in high-dimensional time series." In: *International Conference on Learning Representations* (2016).

[62] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. "Classification of imbalanced data: A review." In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.04 (2009), pp. 687–719.

[63] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks." In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[64] Prasanna Tamilselvan and Pingfeng Wang. "Failure diagnosis using deep belief learning based health state classification." In: *Reliability Engineering & System Safety* 115 (2013), pp. 124–135.

[65] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and composing robust features with denoising autoencoders." In: *Proceedings of the 25th international conference on machine learning*. ACM. 2008, pp. 1096–1103.

[66] TV Vishnu, Narendhar Gugulothu, Pankaj Malhotra, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. "Bayesian networks for interpretable health monitoring of complex systems." In: *Workshop on AI for Internet of Things at IJCAI*. 2017.

[67] Cunsong Wang, Ningyun Lu, Yuehua Cheng, and Bin Jiang. "Deep forest based multivariate classification for diagnostic health monitoring." In: *2018 Chinese Control And Decision Conference (CCDC)*. IEEE. 2018, pp. 6233–6238.

[68] Jun Wu, Kui Hu, Yiwei Cheng, Haiping Zhu, Xinyu Shao, and Yuanhang Wang. "Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network." In: *ISA transactions* (2019).

[69] Andre S Yoon, Taehoon Lee, Yongsub Lim, Deokwoo Jung, Philgyun Kang, Dongwon Kim, Keuntae Park, and Yongjin Choi. "Semi-supervised learning with deep generative models for asset failure prediction." In: *arXiv preprint arXiv:1709.00845* (2017).

[70] Shuai Zheng, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta. "Long short-term memory network for remaining useful life estimation." In: *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2017, pp. 88–95.

[71] Zhiyu Zhu, Gaoliang Peng, Yuanhang Chen, and Huijun Gao. "A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis." In: *Neurocomputing* 323 (2019), pp. 62–75.