Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

Victoria Glott

# Condition Monitoring Of A Gas Turbine Using Bayesian Recurrent Neural Networks.

Master's thesis in Chemical Engineering
Supervisor: Johannes Jäschke, Timur Bikmukhametov
June 2020

NTNU
Norwegian University of
Science and Technology

Victoria Glott

# Condition Monitoring Of A Gas Turbine Using Bayesian Recurrent Neural Networks.

Master's thesis in Chemical Engineering
Supervisor: Johannes Jäschke, Timur Bikmukhametov
June 2020

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

NTNU
Norwegian University of
Science and Technology

# Preface

This master thesis was written in the spring of 2020. The thesis concludes the 5-year master's degree program Chemical Engineering and Biotechnology at the Norwegian University and Technology and is written at the Process Systems Engineering group.

I would like to thank my supervisor Associate Professor Johannes Jäschke for guidance, assistance, and valuable inputs. You have supervised me for my 4th-year project, specialization project, and master thesis and I could never ask for anyone better. I truly believe that every student who gets you as their supervisor in the future is very lucky. I would also like to thank my co-supervisor Timur Bikmukhametov. Your guidance has been invaluable and you have always taken your time to help me. I especially appreciate that you have shared all your knowledge on machine learning with me. I've learned so much from you during my last year and this thesis would never been realized without your help.

## Abstract

Degradation is a major issue for gas turbines and will result in reduced efficiency. Maintenance is, therefore, regularly performed to recover the performance. Monitoring the level of degradation and determining the optimal time of maintenance is vital to ensure cost-optimal operation. Data-driven approaches for determining the state of an asset are becoming an increasingly attractive approach and could provide several benefits for monitoring degradation in a gas turbine. This thesis presents a systematic methodology for measuring the level of degradation in a gas turbine with machine learning and suggests how to utilize the results to determine the optimal time of maintenance. Moreover, the uncertainty associated with the predictions will be evaluated using Bayesian recurrent neural networks. Through the proposed methods, two case studies were performed based on a dataset from an offshore gas turbine, courtesy of Equinor ASA. The first case study investigates the level of degradation in terms of power output, compressor discharge pressure, and fuel consumption for three operating intervals in the dataset. In the second case study, the optimal time of maintenance will be determined based on the reduced throughput of gas for pipeline transmission and increased fuel consumption due to performance degeneration, and maintenance downtime production loss. This thesis confirms that the proposed methodology can be used to determine the level of degradation and that the uncertainty associated with the predictions can provide valuable information to assess the correctness of the results. Finally, based on the level of degradation the optimal time of maintenance was determined.

## Sammendrag

Degradering er et problem for gassturbiner og vil resultere i redusert effektivitet. Vedlikehold blir derfor regelmessig utført for å gjenopprette effekten. For å sikre kostnadsoptimal drift kan det være gunstig å bestemme nivået av degradering. Datadrevne metoder blir stadig mer attraktive og kan tilby flere fordeler for å overvåke degradering i en gassturbin. Denne oppgaven vil derfor presentere en metode for å måle nivå av degradering i en gassturbin med maskinlæring samt beskrive hvordan resultatene kan benyttes for å bestemme det optimale vasketidspunktet. Usikkerheten knyttet til prediksjonene vil bli evaluert med Bayesian recurrent neural networks. Med den foreslåtte metoden ble det utført to casestudier basert på et datasett fra en offshore gassturbin. Fra datasettet ble det plukket ut tre operasjonsintervall. Den første casestudien undersøkte hvordan degradering påvirker kraft, utløpstrykk i kompressor og drivstofforbruk. I den andre casestudien ble det optimale vedlikeholdstidspunktet bestemt basert på degraderingsnivået. Prosjektet har bekreftet av den foreslåtte metoden kan brukes til å bestemme nivået av degradering og at usikkerheten knyttet til prediksjonene kan gi verdifull informasjon for å vurdere riktigheten av resultatene. Til slutt, basert på nivået av degradering, ble den optimale vedlikeholdstiden for gassturbinen bestemt.

# Contents

# List of Figures

# List of Tables

# List Of Symbols

Table 1: List of symbols.

| Symbol | Description |
|---|---|
| $\eta$ | Isentropic Efficiency |
| $h_t$ | Hidden state |
| $h_{t-1}$ | Previous hidden state |
| $c_t$ | Current cell state |
| $c_{t-1}$ | Previous cell state |
| $x_t$ | Input at timestep t in an RNN |
| $\underline{f}$ | Forget gate |
| $\underline{i_t}$ | Input gate |
| $\underline{g_t}$ | Input modulation gate |
| $\underline{o}$ | Output gate |
| $W_h$ | Weight matrix for hidden layer |
| $U_f$ | Forget gate weight matrix |
| $W_f$ | Forget gate weight matrix |
| $U_i$ | Input gate weight matrix |
| $W_i$ | Input gate weight matrix |
| $U_g$ | Input modulation gate weight matrix |
| $W_g$ | Input modulation gate weight matrix |
| $U_o$ | Output gate weight matrix |
| $W_o$ | Output gate weight matrix |
| $w$ | Weight function |
| $X$ | Input dataset |
| $Y$ | Target dataset |
| $x^*$ | Unseen input dataset |
| $y^*$ | Prediction |
| $q_\theta$ | Approximate distribution of $P(w \mid X, Y)$ |
| $\theta$ | Variational parameter |
| $p_{mc}$ | Dropout probability |
| $J$ | Objective function in machine learning context |
| $x^{*(i)}$ | MC sample of $x^*$ |
| $w^{(i)}$ | MC sample of $w$ |
| $E(y^*)$ | Expected value of $y^*$ |

| | |
|---|---|
| $Var(y^*)$ | Variance of $y^*$ |
| $W$ | Set of weights in an LSTM $W = \{W_i, U_i, W_f, U_f, W_o, U_o, W_g, U_g\}$ |
| $z_h$ | Dropout mask for hidden input |
| $z_x$ | Dropout mask for input |
| $\Theta$ | Set of hyperparameters |
| $\Theta^*$ | Optimal set of hyperparameters |
| $\mu$ | Mean |
| $k$ | Covariance |
| $J_{\text{fuel}}$ | Cost function fuel |
| $C_{\text{fuel}}$ | Cost fuel |
| $\dot{m}_{\text{fuel, true}}$ | Actual fuel consumption |
| $\dot{m}_{\text{fuel, expected}}$ | Expected fuel consumption |
| $J_{\text{gas}}$ | Cost function gas |
| $C_{\text{gas}}$ | Cost gas |
| $\dot{m}_{\text{gas, true}}$ | Actual gas throughput |
| $\dot{m}_{\text{gas, expected}}$ | Expected gas throughput |
| $J_{\text{wash}}$ | Cost function for gas turbine maintenance |
| $\overline{m}_{gas}$ | Average throughput of gas |
| $\bar{t}_{downtime}$ | Average maintenance downtime |

# List Of Abbreviations

Table 2: List of abbreviations.

| Acronym | Description |
|---|---|
| CM | Condition monitoring |
| CDP | Compressor discharge pressure |
| CDT | Compressor discharge temperature |
| EGT | Exhaust gas temperature |
| RNN | Recurrent neural network |
| LSTM | Long-short term memory |
| MC | Monte-Carlo |
| EI | Expected Improvement |
| Adam | Adaptive Moment Estimation |

# 1 Introduction

Offshore platforms are reliant on gas turbines for power generation and pipeline transmission of produced gas (Boyce, 2012). A major issue for gas turbines is degradation which results in performance deterioration. There are several factors that contribute to gas turbine degradation. Fouling, erosion, blade tip clearances, and object damage are some of the most common causes. Due to a loss in gas turbine efficiency, the inlet mass flow of air, compressor discharge pressure, and power output decreases (Haq and Saravanamuttoo, 1991). Increased fuel consumption is also a direct effect of gas turbine degradation and causes both increased costs and emissions. To ensure a desirable gas turbine performance, maintenance is performed to recover the losses caused by degradation. However, proper maintenance requires the gas turbine to be shut down and results in production loss. Detecting and monitoring the degradation is therefore important to ensure cost-effective operations and to determine the optimal time of maintenance.

Condition monitoring (CM) is a broad term that covers the methodology of utilizing measurements to determine the status of an asset during operation (Gonfalonieri, 2019). For gas turbines, condition monitoring involves measuring the level of degradation to determine how it subsequently affects performance. The rate of degradation depends on numerous factors such as load, pollution, humidity, and quality of fuel. Consequently, the degradation rate is unique for different operation intervals for the same gas turbine. Therefore, condition monitoring can provide valuable information regarding the specific state of the gas turbine. The most common approach in the industry is to follow the instructions provided by the manufacturer (Meher-Homji et al., 2019). These methods often involve using the measurements to directly inspect abnormalities which can be caused by degradation. However, since the performance of a gas turbine is affected by ambient conditions such as temperature, pressure, and humidity, this method is only valid near standard reference conditions (Krampf, 1992). Also, a common approach frequently involves simply staying alert of health symptoms such as the gas turbine failing to accelerate to full speed. More advanced prognostic methods have also been proposed and can be classified into two main categories: model-based and data-driven. Model-based prognostic is based on accurate mathematical models describing the system to evaluate the overall performance based on the measurements. However, the approach requires accurate models and can be time-consuming, expensive, and difficult to obtain for complex systems (Adamowicz and Zywica, 2018). Data-driven methods have recently received great attention due to promising potential and the ability to address the above-mentioned limitations as well as providing accurate diagnostic. Both degradation forecasting (Zagorowska et al. (2020), Kiakojoori and Khorasani (2016)) some forms of degradation monitoring has been investigated using data-driven methods (Batayev (2019), M. Raghavan et al. (2019)).

However, there is a lack of a systematic data-driven methodology for degradation monitoring in literature. Moreover, no attention has been brought to identifying and selecting suitable variables to monitor the performance deterioration from a machine learning perspective. There are also fundamental limitations of machine learning methods that should be taken into considerations. For instance, the performance is generally poor if the machine learning model is tested on a dataset that is outside the training range. As a result, these predictions are often associated with large uncertainties. Assessing the uncertainty is important because performance deterioration is not a measurable quantity. It is, therefore, no fundamental way to evaluate the correctness of the results. However, the uncertainty can provide information regarding the confidence of the predictions, and can, therefore, be used as a measure to evaluate the correctness. Based on the current challenges and knowledge gaps, this thesis aims to investigate the following research objectives.

- Propose a machine learning methodology for monitoring degradation using historical data.
- Implement the methodology to investigate the level of degradation for measured variables in an offshore gas turbine.
- Assess and evaluate the machine learning uncertainty linked to degradation monitoring.
- Apply the obtained degradation rates to determine the optimal time for maintenance.

# 2 Gas Turbine Background, Process Description, And Dataset

The gas turbine is a power plant that generates large amounts of energy. The compactness, low weight, and applications make it suitable for offshore platforms. For offshore applications, gas turbines are extensively used to drive compressors for natural gas pipelines and for power generation (Boyce, 2012). However, the performance of a gas turbine will deteriorate due to the ingestion of particles like sand, oil, and dust. Monitoring the degradation is therefore important to ensure stable operation and optimal maintenance. This section will present the basic working principle behind a gas turbine, degradation causes, and effects, maintenance procedures, and previously proposed data-driven methods for forecasting and monitoring performance deterioration. Finally, the process description and specifications of the dataset which are used in the case studies will be introduced.

## 2.1 Basic Principles

A gas turbine is a combustion engine that converts natural gas or other fuels into mechanical energy (Zohuri, 2015). The engine derives the power from burning fuel in a combustion chamber with high-pressure air. A simple gas turbine consists of three main parts: a compressor, combustor, and turbine. When compressed air is mixed with fuel and burned under constant pressure conditions the hot gas is expanded through the turbine to perform mechanical work. Intercooling between compressors can also be utilized to cool the air between stages, allowing for more fuel to be burned and thus generating more power. This is the most common design for industrial gas turbines and is referred to as a two-spool gas turbine. The process flow diagram for a two-spool gas turbine is shown in Figure 1.

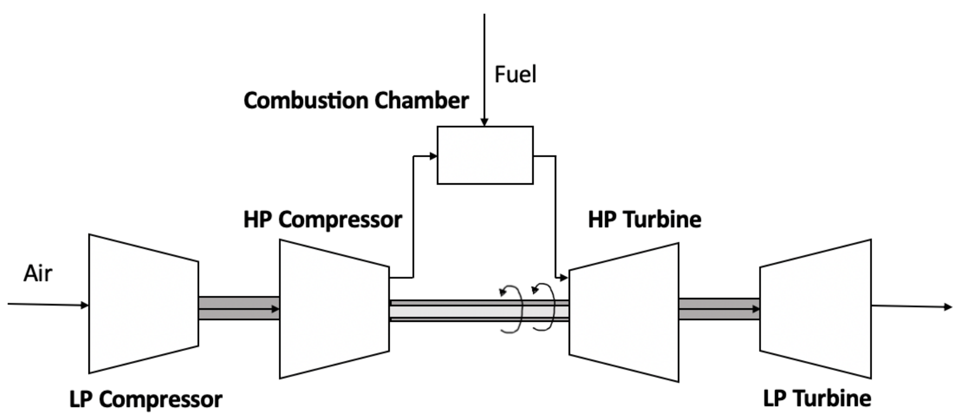



Figure 1: Process flow-sheet of a two-spool gas turbine.

As shown in Figure 1, the air is compressed in two stages, by the low pressure (LP) and high pressure (HP) compressor. The most common type of compressor for gas turbine applications with high power output requirements is the axial-flow compressor (Soares, 2008). In an axial-compressor, the working fluid is accelerated and diffused by a series of rotating and stationary blades, respectively. It is the diffusion process that converts the gained velocity to a pressure rise. An axial-flow compressor usually consists of many stages, with each stage raising the pressure slightly. Since each stage produces low-pressure increments, high compressor efficiencies can be achieved (Boyce, 2012).

After the air is compressed to the desired pressure, it will enter the combustion chamber where the fuel-air mixture ignites, releasing energy as heat. Most modern gas turbines run on either natural gas, diesel fuel, methane, or naphtha. Combustors play a critical role in the operating characteristics of a gas turbine, such as efficiency and level of emission (Breeze, 2016).

After the combustion chamber, the exhaust gas is expanded in the turbine, producing mechanical energy. For a two-spool engine, the HP and LP turbine drives the HP and LP axial compressor, respectively. Gas turbines used for industrial applications are also connected to a power turbine (PT). Power turbines are utilized for driving various loads. For the mechanical drive of a compressor train, the power output will depend on the load and process conditions of the produced gas (Razak, 2007). The power and efficiency characteristics of a gas turbine are, therefore, complex due to different turbomachinery and combustion system interactions (Kurz, 2005).

The performance of a gas turbine will be affected by atmospheric conditions such as ambient pressure, temperature, and relative humidity (El Hadik, 1990). For instance, the power consumed by the axial compressor is proportional to the inlet air temperature. Thus, the performance of a gas turbine increases as the ambient temperature decreases. A high relative humidity is also associated with increased power output. A small ambient pressure can also contribute to poorer performance. However, this effect is most notably for gas turbines installed at different altitudes. The load will also affect performance and the efficiency will drop quickly as the load reduces. However, the efficiency of a two-spool gas turbine is more robust for part-load operations (Smith, 2014).

## 2.2 Degradation

The performance of a gas turbine can deteriorate temporarily or permanently. The most common temporary degradation causes are fouling, erosion, and blade tip clearances, whereas object damage, airfoil, and platform distortion lead to permanent degradation (Fentaye et al., 2019a). These physical faults will change the gas turbine characteristics, causing a decreased performance. Temporarily degradation can, to some extent, be recovered by online and offline washing, or by major engine overhaul.

Fouling is caused by adherence of particles to the airfoils and annulus surfaces in an axial compressor and is categorized as a recoverable temporary degradation cause. Smoke, oil mists, carbon, and sea salts are examples of particles that can build up on the airfoils and result in increased surface roughness. If the gas turbine is not washed and maintained regularly, the build-up material can form a thick layer of deposits and change the shape of the airfoil (Kurz and Brun, 2011). Fouling is the major cause of performance deterioration, and is responsible for 70 to 85% of the total performance loss of a gas turbine (Diakunchak, 1992). Table 3 summarizes the performance change indicators and resulting changes in the gas turbine due to fouling.

Table 3: Summary of the degradation effects caused by fouling.

| Physical Fault | Performance Change Indicator | Result |
|---|---|---|
| Fouling | $\eta\downarrow$ CDP$\downarrow$, CDT$\uparrow$, Air Flow$\downarrow$ | Power output $\downarrow$, Fuel Rate $\uparrow$, EGT $\uparrow$ |

Due to loss in compressor efficiency, the mass flow capacity of air will decrease. Consequently, there will also be a reduction in compressor discharge pressure (CDP). Reduced efficiency and increased friction between the surface of the blades and the airflow will also result in an increased compressor discharge temperature (CDT). Combining these effects, the power output will, consequently, decrease (Haq and Saravanamuttoo, 1991). However, the power turbine will still try to reach the desired power output by increasing the fuel flow. Due to the increased fuel consumption, the exhaust gas temperature (EGT) will thus also increase.

The rate of fouling is site-specific due to different environmental conditions at each plant. Moreover, the deterioration rate is also affected by local changes in ambient and operating conditions. Ambient conditions include factors such as humidity, airborne salt, carbon particles, dust, and oil (Ogbonnaya, 2011). Deposits of sea salts are commonly found in coastal or off-shore locations, oil and grease are typically found in industrial areas while desert regions tend to attract more sand and dust particles (Kurz and Brun, 2012).

In general, particles up to 10 microns are responsible for fouling. Industrial gas turbines are usually equipped with an effective inlet air filtration system, which can restrain the rate of fouling. However, the efficiency of the air filtration system decreases with finer particles. Therefore, a significant amount of smaller particles will still enter the gas turbine and cause fouling (Kurz and Brun, 2012).

The main parameters that are affected by fouling are isentropic efficiency, the flow capacity of air, compressor discharge pressure, and power output. There is, however, no consensus on the magnitude of the extent of the deviation which is caused by compressor fouling (Fentaye et al., 2019a). As shown in Table 4, the effects of fouling on different variables have been studied. The research to date has tended to focus on the relative relationship between variables that are affected by fouling. Diakunchak

(1992) reported that with a 1.8% reduction in isentropic efficiency the flow capacity and power output decreased 5% and 7%, respectively. Similar findings have also been reported by Saravanamutto and Lakshminarasimha (1985) which reported a 5% reduction in flow capacity with a corresponding decrease of 2.5% in isentropic efficiency and 10% in power output. Moreover, Zwebek and Pilidis (2003) found that a 2.5% decrease in isentropic efficiency resulted in a 8% reduction in power output and a 5% reduction in flow capacity. Zaba (1980) suggested that a 1% decrease in mass flow rate corresponds to a 1.25% decrease in compressor efficiency. Meher-Homji et al. (2019) argued that, as a rule of thumb, a 0.138 Bar loss in compressor discharge pressure will result in a gross loss in power output of 1MW. The degradation in terms of power output has also been expressed as a function of operating time. Tarabrin et al. (1998) suggested that the degradation rate is most prominent during the initial operation before it decays exponentially.

Table 4: Summary of previous findings on how degradation affects the gas turbine.

| Reference | Effect of fouling |
|---|---|
| Diakunchak (1992) | $\eta$ 1.8%↓, Power output 7%↓, $\dot{m}$ 5%↓ |
| Saravanamutto and Lakshminarasimha (1985) | $\eta$ 2.5%↓, Power output 10%↓, $\dot{m}$ 5%↓ |
| Zwebek and Pilidis (2003) | $\eta$ 2.5%↓, Power output 8%↓, $\dot{m}$ 5%↓, Fuel Rate 2.5% ↑ |
| Zaba (1980) | $\Delta\dot{m}\downarrow = 1.25\ \Delta\dot{\eta}\downarrow$ |
| Meher-Homji et al. (2019) | CDP ↓ 0.138 Bar, Power output 1MW ↓ |
| Giampaolo (2013) | CDP ↓ 2%, CDT ↑ 0.5%, EGT ↑ 3%, Fuel Rate 3% ↑ |
| Tarabrin et al. (1998) | $\Delta$ Power (%) $= a(1 - e^{-bt})$ |

## 2.3 Maintenance

Since the axial-flow compressor can consume up to 60% of the work generated by the gas turbine it is important to eliminate the effect of fouling to ensure overall high performance (Enyia et al., 2015). A typical gas turbine can ingest large amounts of particles. For instance, it was reported that a 7.5MW gas turbine operating in an environment with a particle concentration of 1ppm can ingest 5 kg of dust and contaminants each day. Whereas if the gas turbine operates in a highly polluted area such as a mining or oil field, up to 40 kg each day can be ingested by the axial compressor (Enyia et al., 2015). Even though compressor fouling will not result in engine breakdown or fault, it will reduce the compressor power output and efficiency. The state-of-the-art method for removing fouling is washing and is achieved by injecting fluid at the front end of the gas turbine (Boyce and Gonzalez, 2005).

Washing can be categorized into online and offline. The former is performed under operation, while the latter requires the gas turbine to be shut and cooled down. Therefore, offline cleaning is more expensive in terms of production loss due to downtime, and labor cost. Scheduling the offline washing should, therefore, take into consideration the trade-off between loss due to deteriorated gas turbine performance and lost production cost (Boyce and Gonzalez, 2005). Due to the high cost associated with offline washing, online washing has become an increasingly attractive method since it is performed during operation. Online washing if often scheduled regularly, and is performed more frequently compared to offline washing. However, online washing is not as effective as offline washing and restores only partly the loss in performance (Enyia et al., 2015). Online washing will, therefore, not eliminate the need for offline washing, but rather increase the interval between each offline wash.

## 2.4 Predictive Maintenance

Degradation of important equipment and devices is a challenge several companies face and can lead to both failure and reduced performance. Since the consequences of failures can be comprehensive many companies use a precautionary approach referred to as preventative maintenance. The approach in preventative maintenance is regular maintenance cycles to avoid failure. Thus, the current condition of the asset does not affect the maintenance schedule (Meyer Zu Wickern, 2019). Since most companies deal pessimistically with maintenance, the components of an asset will be replaced in good time ahead of failures. Although regular maintenance is better than failures, this approach will often result in performing maintenance before it is required. Therefore, to determine the optimal time of maintenance the concept of Predictive Maintenance has becoming increasingly popular. The goal is to find patterns to forecast and measure the state of an asset to prevent failures and obtain optimal schedules for maintenance. Despite being widely used, Predictive Maintenance still faces challenges within reliability, robustness, and accuracy (Gonfalonieri, 2019).

### 2.4.1 Condition Monitoring

Condition monitoring (CM) is a subcategory of Predictive Maintenance and utilizes measurements to determine the status of an asset during operation. By continuously measuring the health status of an asset, the current state can be used for maintenance planning (Gonfalonieri, 2019). For gas turbines, the rate of performance deterioration will vary depending on operating conditions such as atmospheric conditions, pollution, load, and quality of fuel. The degradation rate will, therefore, be distinct for all operating intervals. Consequently, the main advantage of condition monitoring for a gas turbine is that it allows operators to better schedule maintenance for optimal economic operation (Fentaye et al., 2019a).

The ability to predict and evaluate the health of a component over time is the foundation of condition monitoring. The prognostic methods can be classified as data-driven, model-based, and knowledge-based. Model-based prognostic utilizes accurate physics-based mathematical models. These models can be used to evaluate the performance based on the measurement readings. The model-based output can be compared with the actual measurements to estimate the level of deterioration. However, this analytical approach is most effective for small-scale systems since developing an accurate model can be time consuming and expensive (Adamowicz and Zywica, 2018). Knowledge-based systems are based on human expert knowledge and are developed based on facts and rules to mimic the advice given by an experienced engineer. The prognosis system is often based on IF-THEN statements based on fault and performance deterioration symptoms. Using knowledge-based systems is preferred when the problem is well understood and the gas turbine operating conditions are stable and predictable. However, knowledge acquisition is often time-consuming and will only cover known faults under similar operating conditions (Fentaye et al., 2019b). The drawbacks of model and knowledge-based have resulted in an increased focus and data-driven methods. Data-driven methods utilize historical data to make statistical relationships to map input-output relationships. Thus, eliminating the need for developing detailed analytical models.

A considerable amount of literature has been published on fault detection and diagnostics using machine learning methods. However, relatively little attention has been paid to degradation monitoring using machine learning. Batayev (2019) proposed a method for estimating the degradation level based on fuel consumption. The predicted fuel consumption was obtained by training a neural network on available gas turbine measurements. The ratio between the calculated fuel consumption, from first-principle models, and the predicted fuel rate was used as a measure for the level of degradation. M. Raghavan et al. (2019) also used machine learning methods to measure degradation. The variables used for degradation monitoring was compressor discharge pressure and temperature, and mass flow of air. However, the model was trained on historical data which accommodated a normal rate of fouling. The machine learning model could, therefore, only detect abnormal degradation. Kiakojoori and Khorasani (2016) used neural networks to forecast fouling trends in an aircraft turbine. The neural network was trained on historical data which was produced synthetically using modeling software. The aim was to forecast the turbine temperature increase caused by fouling. Zagorowska et al. (2020) also proposed a method for degradation forecasting based on historical data from an offshore gas turbine. The aim was to forecast a degradation indicator, defined as the deviation between the efficiency of a healthy and a degraded gas turbine. The healthy efficiency was calculated based on the data from the manufacturer, and the degraded efficiency was obtained using the gas turbine measurements and simplified thermodynamic relationships. By assuming the degradation rate was a combination of constant, linear, and exponential terms, the algorithm was able to forecast the degradation indicator accurately for short to medium time intervals.

Despite several proposed techniques and methods in the literature, the most common practice in the industry for detecting fouling is based on simpler methods and operator intuition (Meher-Homji et al., 2019). For instance, being alert to gas turbine symptoms such as the engine failing to accelerate to full speed, high exhaust temperature and inability to run at full load. A common approach is also to plot variables such as power output and correct it for a given ambient temperature, corrected speed, or compressor mass flow rate and compare it with the compressor performance maps provided by the manufacturer. Yet, this method is often this method is only valid close to standard reference conditions (Krampf, 1992). Therefore, many operators claim that the only way of detecting fouling is by visual inspection (Meher-Homji et al., 2019). However, for most designs, a visual inspection will involve the gas turbine to be shut down.

### 2.4.2 Health Parameters

For condition monitoring, it is crucial to select suited variables for effective and accurate monitoring. Variables which are used for monitoring the state of an asset is often referred to as health parameters. There have been proposed several health parameters for monitoring the state of a gas turbine, such as:

- Compressor Discharge Pressure
- Compressor Discharge Temperature
- Compressor Isentropic Efficiency
- Compressor Air Flow
- Exhaust Gas Temperature
- Fuel Flow Rate
- Power Output

The requirements for a good health parameter are based on the methodology of visual inspecting of the measurements and correcting for ambient conditions and load (Haq and Saravanamuttoo, 1991). Generally, the health parameters should,

- provide an accurate and repeatable indication of compressor condition
- serve as an accurate indicator of the state of the gas turbine.
- not be affected by external variables and load
- be affected sufficiently to reduce effects of measurement noise.

Moreover, Haq and Saravanamuttoo (1991) also argued that health parameters that are calculated using monitoring software based on several measurements are associated with a larger uncertainty for degradation and can be unsuitable. For instance, the inlet compressor airflow is calculated using total pressure and temperature, and static pressure. Since each variable is associated with a corresponding measurement error, the calculated value will be uncertain. The same argument was also made for isentropic efficiency, where inlet and outlet pressures and temperatures are necessary. The power output is also calculated by two measurements, the torque and shaft speed. Haq and Saravanamuttoo (1991) argued that the compressor discharge pressure is the most suitable parameter for monitoring performance deterioration. Firstly, because the compressor discharge pressure is measured directly and, secondly, it exhibited a consistent downward trend due to degradation.

## 2.5 Optimization Of Washing Schedule For Economic Operation

The frequency between each shutdown to perform offline washing will depend on the accumulated level of degradation (Hanachi et al., 2018). Operating a gas turbine in a deteriorated state is associated with significant costs. Cost related to performance loss must address the value of the resulting shortfall in power output. How the reduced power output affects the revenue will depend on the specific application. For electrical power generation, the loss can be calculated through the shortfall in power output and electrical price. However, for offshore applications, the loss can be estimated as the reduction of the throughput of gas compressed and moved through the pipeline and the sales price of the gas. Moreover, to compensate for the loss in performance, the fuel consumption will also increase thus adding additional costs. On the other hand, there are costs associated with gas turbine washing. These costs include the loss in production caused by downtime, labor costs associated with washing and fixed costs (Hanachi et al., 2018).

## 2.6 Process Description And Dataset

This section will briefly cover the process that will be taken into consideration. There are two main parts of the process: the gas turbine and compressor train. The process flow-sheet for the system that will be considered in this thesis as well as a selection of the measured variables are shown in Figure 2.

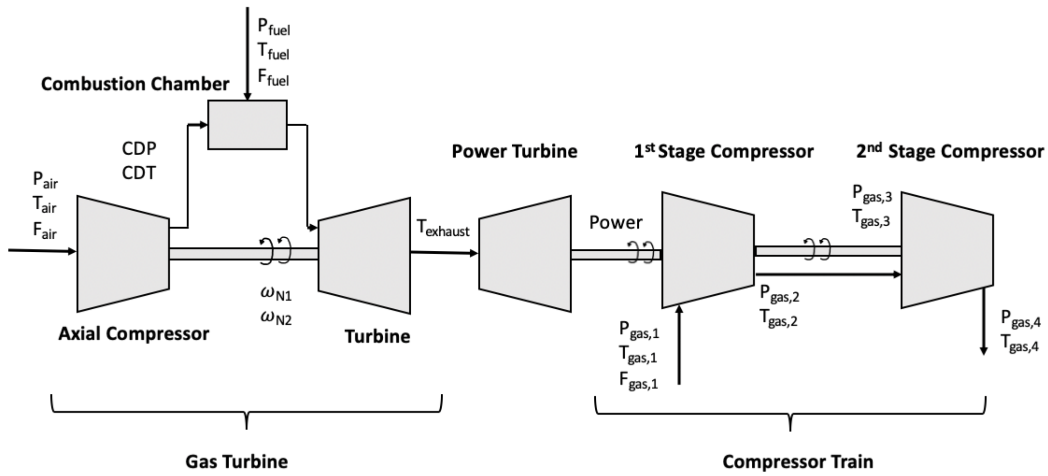



Figure 2: Flow-sheet of the process that will be considered in this thesis. The process consists of a gas turbine and a compressor train.

The gas turbine is a two-spool engine as previously introduced in Figure 1. However, since the measurements only incorporate the input and output for the first and second, respectively, for both compressor and turbine, the gas turbine will be considered as a single-spool engine, as shown in Figure 2. The gas turbine delivers power to two components; the axial compressor and the compressor train. The compressor train consists of two compressors, the 1$^{st}$ and 2$^{nd}$ stage compressor. The compressor train is responsible for compressing the produced gas and is considered as the bottleneck in the process.

The dataset is sampled during one year of production and origins from the Heidrun oil and gas field. The variables are sampled approximately every five minutes. The most important variables which are sampled regarding the gas turbine are shown in Figure 2. The dataset is characterized by regular shutdowns, which are either caused by offline washing or trips. The time of washing can be identified by an increased performance after shutdown, while the compressor trips in general yields the same performance after shutdown. It was identified in total four time series segments where the time of washing could be determined at the start and end of the operating interval. However, one of the operating intervals was disturbed with numerous trips and was therefore not chosen to investigate further.

# 3 Machine Learning Background

Machine learning is a branch of artificial intelligence that identifies patterns for regression analysis and makes decisions without human intervention (Mak et al., 2019). A popular machine learning algorithm for time series analysis is the recurrent neural network, abbreviated as RNN. Due to recurrent connections between each prediction step that mimics the concept of memory, RNNs have proven to be superior for data with sequential characteristics. However, machine learning models do have some fundamental limitations. For instance, poorly tuned models, noise and dissimilar training and test set distribution can decrease the performance (Gal, 2016). Evaluating prediction confidence is, therefore, becoming an increasingly attractive field of study. In other words, it is desirable to assess whether the predictions can be trusted. This section will present the necessary background on recurrent neural networks and how to obtain uncertainty through Bayesian methods. Moreover, machine learning uncertainty causes and interpretations will also be introduced. Finally, Bayesian hyperparameter optimization will be presented as a tuning technique. The following sections assume the reader to be familiar with common machine learning concepts such as feed-forward neural networks, hyperparameters, backpropagation, weight matrices, dropout, and activation functions. Otherwise, Appendix A will cover the necessary background.

## 3.1 Recurrent Neural Networks

RNNs or recurrent neural networks are a class of neural networks that are suited for sequential data. Feed-forward architectures can struggle with sequential data as there is no information flow between each pass of a data point. RNNs, on the other hand, allows for temporal dependencies that enable the network to account for time (Witten et al., 2017). Briefly explained, RNNs perform the same task for each data point in the sequence, where the output from a previous data point is passed on to the next computation. The information is fed through hidden layers between each step, which gives the RNNs the characteristics of memory. Figure 3 shows the architecture of one RNN cell.

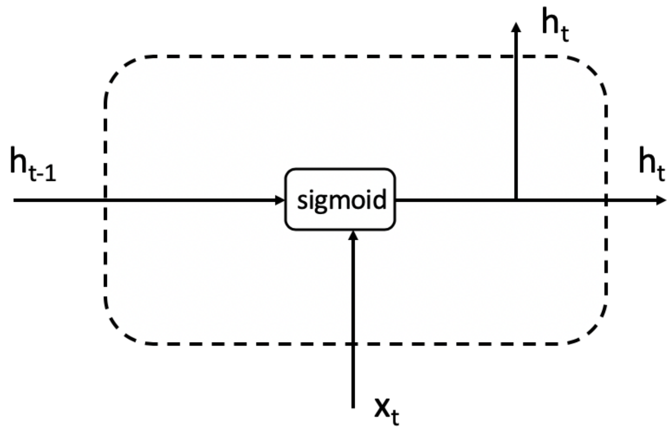



Figure 3: Illustration of an RNN cell and the information flow. For each time step, the RNN cell produce a new hidden state based on the previous hidden state and input.

Figure 3 shows how the input from a previous timestep and current input is feed into an activation function to produce the current hidden state. The calculation of the current hidden state can be expressed mathematically as,

$$h_t = \text{sigmoid}(x_t W_h + h_{t-1} U_h) \tag{1}$$

Where $x_t$ denotes the input, $h_{t-1}$ the previous hidden state, $h_t$ the current hidden state, $W_h$ and $U_h$ the weight matrices for the input and hidden states, respectively (Sherstinsky, 2018). However, RNNs tend to store short-term information. That is, if the input sequence is long, the information from earlier time steps can fade. RNNs also have issues with vanish gradients during backpropagation (Witten et al., 2017). Gradients are the values that are used to update the weights and have therefore a large impact on the training process. Therefore, a long sequence of multiplied gradients will result in the final gradients to, if not vanish, become too small to contribute with learning. To overcome the limitations of an RNN, Hochreiter and Schmidhuber (1997) suggested a recurrent algorithm named Long Short-Term Memory, commonly referred to as LSTMs, and were designed to retain long-term dependencies. With internal mechanisms called gates, the network can control and filter the important information. There are four gates in an LSTM: the input, forget, output, and input modulation gate. The gates decide which information to keep and forget during training. The remainder of this section will outline the working principles behind LSTMs based on the article published by Hochreiter and Schmidhuber (1997). However, the notation is simplified as presented by Phi (2018). Figure 4 shows the architecture of one LSTM cell.

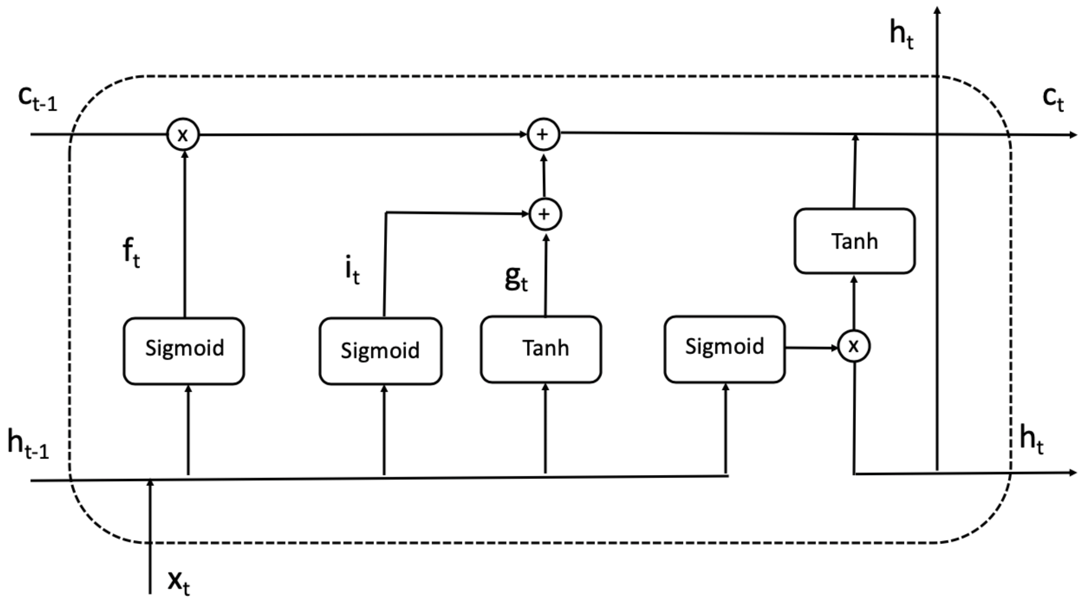



Figure 4: Illustration of an LSTM cell. The figure shows the information flow, operations, and activation functions for the four main gates: input, forget, output, and input modulation gate.

The most distinct feature that separates the LSTM from an RNN is the cell state, denoted $c_t$ and $c_{t-1}$ in Figure 4. The cell state transfers relevant information throughout the network and makes sure that information from earlier time steps is accounted for. The cell state is thoroughly regulated by the gates by either adding or removing information such that the most important information from earlier steps is passed on. In this way, the LSTM is better equipped to hold on to long term information opposed to an RNN.

The forget gate, $\underline{f_t}$, decides what information should be preserved and rejected from *prior* steps. Information from the previous hidden state, $h_{t-1}$ and current input, $x_t$ is sent through the sigmoid activation function. The forget gate can be mathematically expressed as,

$$\underline{f_t} = \text{sigmoid}(h_{t-1}U_f + x_tW_f) \tag{2}$$

Where $U_f$ and $W_f$ are weight matrices for the hidden and input state in the forget gate, respectively. The forget gas outputs a vector with values between 0 and 1 which indicates the importance of the information from prior steps. Where 1 signifies that information should be remembered, and 0 means the information can be forgotten.

The input gate, $i_t$, decides whether any new information, based on the *current* input, is going to be saved in the cell state. Similar to the forget gate, the hidden state and input are passed through a sigmoid function which outputs a vector between 0 and 1 and indicates the importance. The hidden state and input are also passed by the input modulation layer, $g_t$, and is a tanh function which outputs values between -1 and 1. The output from the input gate and the input modulation gate is multiplied to determine what information to keep. The input gate can be expressed as,

$$\underline{i_t} = \text{sigmoid}(h_{t-1}U_i + x_t W_i) \tag{3}$$

Where $U_i$ and $W_i$ are the weight matrices for the hidden and input state in the input gate, respectively. The input modulation layer is given as,

$$\underline{g_t} = \tanh(h_{t-1}U_g + x_t W_g) \tag{4}$$

Where $U_g$ and $W_g$ are the weight matrices for the hidden and input state in the input modulation gate.

Based on the input, input modulation gate and forget gate the new cell state can be calculated. First, the previous cell state is pointwise multiplied ($\odot$) with the forget vector, $\underline{f_t}$. This ensures the cell state removes information that was decided to ignore based on the forget gate. Then, the pointwise multiplication of the output from the input and input modulation gate is also added to the cell state. In this way, the network can update the state to new values that it finds relevant. The calculation of the new cell state is as follows,

$$c_t = \underline{f_t} \odot c_{t-1} + \underline{i_t} \odot \underline{g_t} \tag{5}$$

Lastly, the output gate determines the next hidden state. In LSTMs hidden states are both used to transfer information from previous inputs and for making final predictions. To calculate the current hidden state, the previous hidden state and current input are passed through a sigmoid function,

$$\underline{o} = \text{sigmoid}(h_{t-1}U_o + x_t W_o) \tag{6}$$

Then, the hidden layer can be determined from the state of the output gate and the current cell state,

$$h_t = \underline{o} \odot \tanh(c_t) \tag{7}$$

To summarize, the most important gates and their functionalities are as follows,

- The forget gate determines relevant information from *prior* steps.
- The input gate determines relevant information to add based on the *current* step.
- The output gate determines the next hidden state.

LSTM cells can also be stacked to build a deeper network. The number of stacked LSTM cells is referred to as hidden layers. Figure 5 shows the architecture and information flow of an unrolled LSTM network with two hidden layers. At the end of each LSTM chain, the network outputs a prediction, $y^*$.

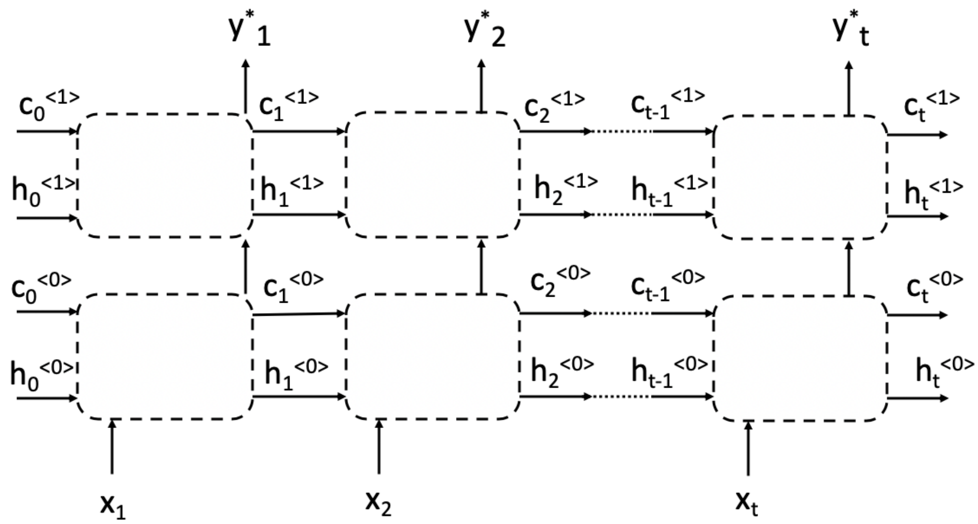



Figure 5: Illustration of an LSTM network with t unrolled steps and two hidden layers. The figure shows the information flow of the input layer, hidden layer, and cell state between each layer and time step.

### 3.1.1 Bayesian Neural Networks

During the last years, neural networks have gained large recognition for outstanding performance in many different tasks. Recently, assessing the uncertainty linked to the predictions has also gained attention (Gal, 2016). One way of obtaining uncertainty is by the use of Bayesian neural networks. Bayesian neural networks is a family of neural networks where the weights are represented by distributions instead of point estimates (Neal, 1996). Yarin Gal and Zoubin Ghahramani (2016) proposed a method to obtain model uncertainty using dropout, i.e. dropping random units in a neural network, and was able to quantify the prediction uncertainty of the neural network based on a Bayesian interpretation. The idea is to perform dropout on multiple passes during the training and test phase. This approach will then be equivalent to Monte-Carlo sampling over a Gaussian process posterior approximation. The derivation is outside the scope of this thesis, instead, and this section will briefly present the basic concept and how to obtain model uncertainty from a practical perspective. The approach is first presented in the context of feed-forward neural networks and the adaption for recurrent neural networks will be covered in Section 3.1.2.

To estimate the expected values and corresponding variance, the posterior distribution over all parameters is required. Applying Bayes Theorem, the posterior of the most likely model parameters, given the observed data, X and Y, can be expressed as,

$$P(w \mid X, Y) = \frac{P(Y \mid w, X)P(w)}{P(Y \mid X)} \tag{8}$$

Where $P(Y \mid w, X)$ is the likelihood of the occurrence of Y, given the model parameters $w$ and input $X$. $P(W)$ is the prior and $P(Y \mid X)$ is the normalizer. The idea is then to predict $y^*$ based on an unseen datapoint $x^*$. By integrating over all network weights, $w$, the expected output can be found through a process known as interference.

$$P(y^* \mid x^*, X, Y) = \int P(y^* \mid x^*, w)P(w \mid X, Y)dw \tag{9}$$

However, the true posterior $P(w \mid X, Y)$ cannot be evaluated analytically for more complicated models due to a complex form of the probability distribution. Based on the findings of Graves (2011), a variational scheme can be used to approximate the Bayesian posterior distribution. The idea is to introduce a simplified approximate distribution, $q$ that is easier to evaluate compared to the initial distribution. $q$ will then severe as a variational approximation, with corresponding variational parameter $\theta$.

$$q_\theta(w \mid X, Y) \approx P(w \mid X, Y) \tag{10}$$

To ensure that $q_\theta$ is a good approximation of the true posterior distribution, the Kullback–Leibler divergence between the two functions can be minimized with respect to $\theta$. However, introducing the Kullback–Leibler divergence results in an intractable integral. Monte-Carlo (MC) integration can then be performed to estimate this quantity and uses random sampling of the function to numerically approximate the integral.

By selecting $q$ as a variational distribution that resembles dropout and sampling $w$ from $q(w)$ with MC integration, the following objective function can be obtained.

$$J(X, W, p_{mc}) = -\frac{1}{N}\sum_{i}^{N} \log P(x^{*(i)} \mid \hat{w}^{(i)}) + \frac{1 - p_{mc}}{2N}\|W\|^2 \tag{11}$$

Where J denotes the objective function, $W$ the weights, $p_{mc}$ the dropout probability and $\hat{w}^{(i)}$ the weight sample drawn from the variational approximation $q_\theta$. In the test phase, two quantities are of interest, the expected model output - the predictive mean, and the confidence associated with the predictions - the predictive variance. By performing T passes over the neural network during the test phase with new dropout masks for each pass, the empirical samples can be used for finding the predictive mean and variance,

$$\mathrm{E}(y^*) \approx \frac{1}{T}\sum_{t=1}^{T} P(x^* \mid \hat{w}^i) \tag{12}$$

$$\mathrm{Var}(y^*) \approx \sigma^2 + \frac{1}{T}\sum_{t=1}^{T} P(x^* \mid \hat{w}^{(i)}) - \mathrm{E}(y^*)^T \mathrm{E}(y^*) \tag{13}$$

Where $\mathrm{E}(y)$ is the expected model output and $\mathrm{Var}(y)$ is the predictive variance. $\sigma$ the noise associated with the data. This is an uncertainty that cannot be reduced by including more data.

### 3.1.2 Bayesian Recurrent Neural Networks

MC dropout has proven to be a good approach for obtaining the uncertainty in neural networks. However, applying the technique in recurrent neural networks has proven to be challenging. Studies have shown that adding dropout in recurrent layers can add noise and drown the signal between the recurrent connections (Zaremba et al., 2014). Therefore, it was concluded that dropout should only be applied to the input and output connections. Gal (2015) proposed variational dropout with *repeating* dropout masks for each input, output, and recurrent layer for each time step. By following the same approach briefly described in the previous section, and inserting for the recurrent gate equations described in Section 3.1, the reasoning for the same dropout mask at each time step can be proved. This method is different from previous techniques where all time steps have distinct dropout masks. The proof for this approach is outside this thesis scope. Instead, the practical aspects and how to obtain model uncertainty will be covered.

As described in Section 3.1, the four main gates of an LSTM is the input gate, forget gate, output gate, and input modulation gate. Gal (2015) suggested a re-parameterisation of LSTM model to based on Equation 2, 3, 4 and 6,

$$\begin{pmatrix} \underline{\mathbf{i}} \\ \underline{\mathbf{f}} \\ \underline{\mathbf{o}} \\ \underline{\mathbf{g}} \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \tanh \end{pmatrix} \left( \begin{pmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{pmatrix} \mathbf{W} \right) \tag{14}$$

Where $\mathbf{W}$ is the set of weights introduced in Equation 2, 3, 4 and 6, and is defined as $\mathbf{W} = \{W_i, U_i, W_f, U_f\ W_o, U_o, W_g, U_g\}$. There are some differences between taking basis in the parametrisation presented in the above-mentioned equations and Equation 15. For the gate parametrisation described by the equations presented in Section 3.1, the dropout mask is placed over the weight matrices rather than the input. Whereas for the parametrisation in Equation 15 the dropout masks is applied over the actual input and hidden layers. The advantage of the latter approach is a faster forward pass, but with slightly diminished results. The dropout variant can therefore be written as,

$$\begin{pmatrix} \underline{\mathbf{i}} \\ \underline{\mathbf{f}} \\ \underline{\mathbf{o}} \\ \underline{\mathbf{g}} \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \tanh \end{pmatrix} \left( \begin{pmatrix} \mathbf{x}_t \odot z_x \\ \mathbf{h}_{t-1} \odot z_h \end{pmatrix} \mathbf{W} \right) \tag{15}$$

Where $z_x$ and $z_h$ denotes the dropout masks. Conceptually, since both feed-forward and recurrent neural network use MC dropout to approximate the posterior distributions, the expected values and variance can be derived from the posterior distributions shown in Equation 12 and 13. Figure 6 shows a illustration of the proposed dropout technique in an LSTM where the colored connections represents dropout masks.

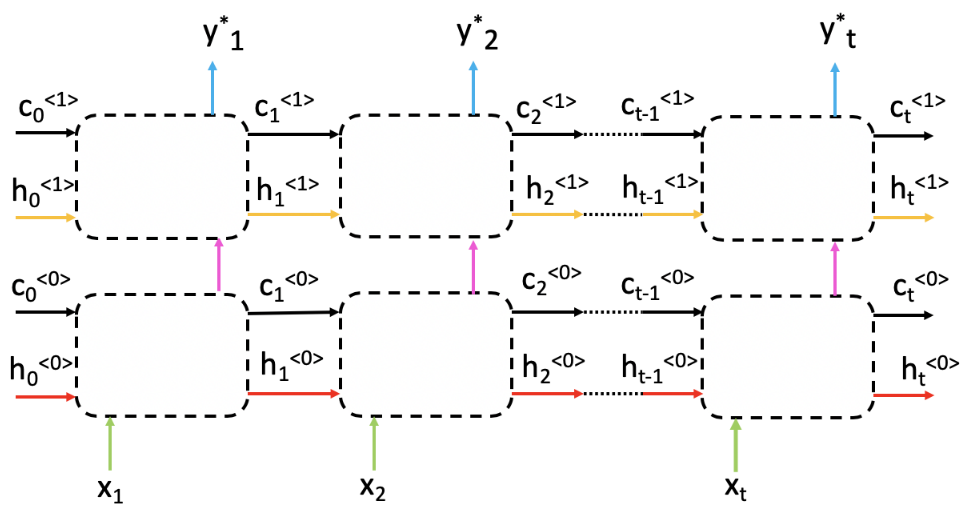



Figure 6: Illustration of a variational LSTM with the proposed dropout technique proposed by Gal (2015), where each timestep have the same dropout mask. The colored lines shows the different dropout masks for the input, cell state, and output connections.

## 3.2 Machine Learning Uncertainty

An important consideration when evaluating predictions should be whether the machine learning model is certain. It is commonly accepted that predictions are unreliable if the model is tested outside the distribution it was trained upon and machine learning models are therefore often considered as ”a prisoner of its training data set” (Minns and Hall, 1996). This fact is often neglected when deploying machine learning models into the real world (Varoonchotikul, 2003). There are different sources of uncertainty associated with the predictions from a machine learning model.

An unrepresentative training set is one of the most common sources of uncertainty. As mentioned above, the uncertainty tends to be large if the test set is outside the trained scope. The absence of representative training data results in epistemic uncertainty (Gal, 2016). An intuitive explanation for epistemic uncertainty is the level of consistency in the predictions. It is, therefore, a measure of how well models agree on the outcome. For instance, if multiple models produce deviating predictions, the data it was trained upon did not succeed in capture the full range of the underlying relationships. This is also apparent by inspecting the second term in Equation 13. If the sampled predictions deviate, the variance will also, consequently, increase. However, epistemic uncertainty can be reduced by increasing the training set and could theoretically be eliminated.

Epistemic uncertainty also includes model uncertainty and can be further classified into uncertainty in the model parameters and structure. Structure uncertainty includes the selection of an appropriate machine learning model such as a feed-forward, recurrent, or convolutional neural network (Gal, 2016). In other words, structure uncertainty describes how well the model explains the given data. Choosing a model that is not suited to explain the type of data, whether it is a time series, images, or sentences, can, therefore, lead to uncertainty. Uncertainty in model parameters reflects the uncertainty in the weights and biases and is described by the Bayesian posterior $P(w \mid X, Y)$. There will often be several models and parameters that can be used for explaining a dataset. The issue is, however, how to generalize the parameters in the model such that it can be useful for an unseen dataset. When training a machine learning model details are often discarded in order to generalize the underlying relationships in the dataset (Brownlee, 2019). As such, the model will make errors on both the training and test set thus causing uncertainty associated with the generalization.

Aleatoric uncertainty arises due to the stochastic variability in data sampling, for instance, due to an imperfect sensor. Every aspect of collecting the measurements that is unknown and introduces randomness is categorized as aleatoric uncertainty. Aleatoric uncertainty is a irreducible uncertainty. In other words, it cannot be reduced by including more training data or improving the model (Gal, 2016).

## 3.3 Bayesian Hyperparameter Optimization

Machine learning models frequently involve tuning of model hyperparameters. If the set of hyperparameters is selected poorly, the performance can be significantly affected. Conceptually, the objective of hyperparameter tuning is to minimize the validation error as a function of the hyperparameters,

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} f(\Theta) \tag{16}$$

Where $\Theta^*$ denotes the optimal set of hyperparameters and $\Theta$ the set of hyperparameters in the search space. The challenge is, however, that $f(\Theta)$ is an expensive function to evaluate. For each time the function is evaluated a new model must be trained and validated.

There are four main ways of finding the optimal set of hyperparameters, hand-tuning, grid-search, random-search, and Bayesian hyperparameter optimization. Hand-tuning the hyperparameters can be time-consuming. Besides, the optimal set of hyperparameters can be counter-intuitive. Grid-search can also find the optimal set of hyperparameters by iterating through every combination. This brute-force approach can also be time-consuming since the number of joint combinations grows exponentially. In random-search, the sets of hyperparameters are selected randomly. Random-search is similar to grid-search but has proven to yield slightly better results (Yakowitz and Lugosi, 1990).

Bayesian hyper-parameter optimization has proven to be a superior alternative to the above-mentioned methods. Rather than testing every combination in the search space, the Bayesian approach offers a more intelligent procedure by selecting the next set of hyperparameters to evaluate based on prior knowledge (Lévesque et al., 2016). There are two main concepts in Bayesian hyperparameter optimization: the surrogate model and the acquisition function. The surrogate function is an approximating of the objective function, $f(\Theta)$, and is computationally cheaper to evaluate. Due to the flexibility and tractability, a Gaussian process prior is often selected as the surrogate function for $f(\Theta)$ (Snoek et al., 2012).

$$f(\Theta) \sim GP(\mu(\Theta), k(\Theta, \Theta')) \tag{17}$$

Where $\mu(\Theta)$ denotes the mean, and $k(\Theta, \Theta')$ the covariance. For each sample, the surrogate will map the mean and variance for the search space (Snoek et al., 2012). The purpose of the acquisition function is to propose a new set of hyperparameters that are most likely to improve accuracy. The next set of hyperparameters proposed by the acquisition function is based on the posterior distribution from the surrogate model. A widely used acquisition is the expected improvement and is defined by Equation 18 (Lévesque et al., 2016).

$$\mathrm{EI}(\Theta) = \mathbb{E}[max(0, f' - f(\Theta))] \tag{18}$$

Where $f'$ denotes the best accuracy for $f(\Theta)$ so far. The expected improvement is generally large for unexplored regions of $f(\Theta)$. Figure 7 illustrates one step of Bayesian hyperparameter optimization (B. Shahriari et al., 2016). Even not usually known, the black dashed line illustrates the true objective. By sampling different sets of hyperparameters the surrogate function, shown in black, maps the corresponding mean and variances. Based on this information the acquisition function, shown in green, determines the next sampling point based on the expected improvement. The expected improvement is generally large for points that are expected to be good and low for previously sampled observations. An important feature of the acquisition function is the trade-off between exploitation and exploration. In other words, if the acquisition function should explore larger parts of the search space of hyperparameters or exploit smaller regions.

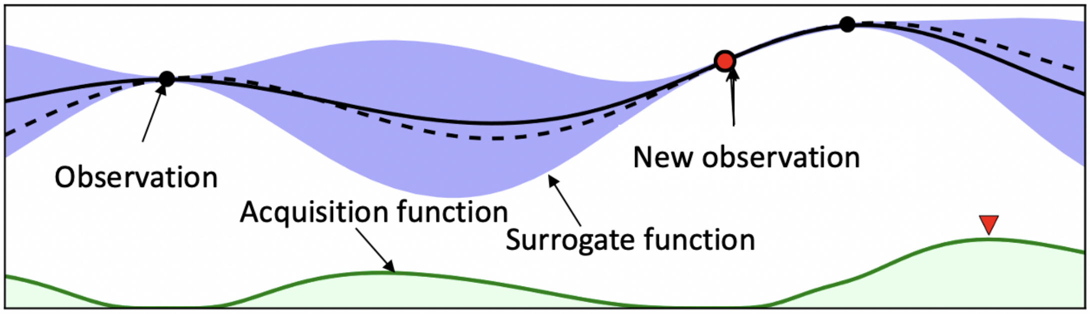



Figure 7: Illustration of one step with Bayesian hyperparameter optimization. The figure shows how new sets of hyperparameters are sampled, how the surrogate model approximates the true objective, and how the acquisition maps the corresponding expected improvement (B. Shahriari et al., 2016)

# 4 Proposed Methodology

Set aside the general advantages of using data-driven methods for condition monitoring, such as eliminating the need for complex analytical models, using machine learning methods to estimate the level of degradation for a gas turbine has several benefits. Firstly, machine learning methods can handle varying ambient conditions. As briefly explained, the performance of a gas turbine will change under different atmospheric operating conditions. Ambient variables can be included as input features thus making the model able to differentiate between performance changes in ambient conditions and degradation. The gas turbine and compressor train loads can also easily be incorporated into the machine learning model such that the relative increase or decrease in output caused by degradation can also be distinguished from changes due to load.

Moreover, the rate of degradation will be dependent on local operating conditions such as humidity, rain, and pollution. Thus, it is expected that the performance deterioration is unique for all operating intervals. What makes machine learning a superior method for these problems is that it can find the degradation rate without any knowledge about external conditions. In this way, the degradation rate can be customized for each specific operating interval. However, due to the black-box characteristics of machine learning methods, information about what is causing the degradation is not obtainable. Therefore, the extent of performance loss that is recoverable by maintenance cannot be determined.

To date, there has been little agreement on a general methodology for monitoring the level of degradation in a gas turbine with machine learning. This section will, therefore, present a systematic approach to several aspects such as choosing training, validation, and test data set, feature selection, and preprocessing.

## 4.1 Key Concept

One of the main objectives of the thesis is to measure the gas turbine degradation with machine learning. However, since degradation is an immeasurable variable, a common approach is to use health parameters that will serve as a unit of measure for degradation. That is, the health parameters will serve as a replacement for measuring the degradation directly and will be used as an indicator for determining the level of gas turbine deterioration.

The strategy is to train and validate the machine learning method on data that are sampled under ”new and clean” conditions. In other words, the machine learning model will be taught what outputs which, in this case, is the health parameters, corresponds to certain inputs under the assumption that the gas turbine is new and clean. In the test set, the input-output correlations are assumed to be different due to degradation. The machine learning model will, therefore, predict the output which is expected if the gas turbine was new and clean on the test set. The *bias* between the predicted expected output and the true measurement can, therefore, be used as a measurement for the level of degradation. Based on a physical interpretation, there are two sources that cause bias in the test set.

- For the same inputs, the gas turbine will not yield the same output in a deteriorated state.
- To achieve the same output in a deteriorated state, the gas turbine must manipulate the input variables.

The first case describes a gas turbine working either at full-load or close to an operating constraint. At full load, the gas turbine generates maximum output thus system variables cannot be altered to increase the capacity. Thus, if the gas turbine deteriorates the output will exhibit a constant downwards trend. Operating constraints, such as maximum exhaust gas temperature, may also cause the gas turbine to limit its capacity. For instance, increasing fuel flow will also increase the exhaust gas temperature. This will, therefore, limit the gas turbine to produce the desired output in a deteriorated state.

The latter case describes a gas turbine working part-load and far off from the operating constraints. Then, as the gas turbine deteriorates, manipulated variables in the system can be adjusted to compensate for the loss in performance. However, under normal operating conditions, these inputs would correspond to a different output. For instance, if the compressor deteriorates and the fuel flow is increased to maintain constant power output, the increased fuel flow would, under ”new and clean” conditions, actually correspond to a larger power output.

Whether the gas turbine operates full-load, part-load, or close to an operating constraint is conceptually irrelevant for the machine learning model. Since the machine learning model only maps the input-output correlations for a new and clean gas turbine, it will capture changes due to deterioration in both the inputs and outputs. Figure 8 shows the main concept for measuring the performance deterioration in the training and test phase.

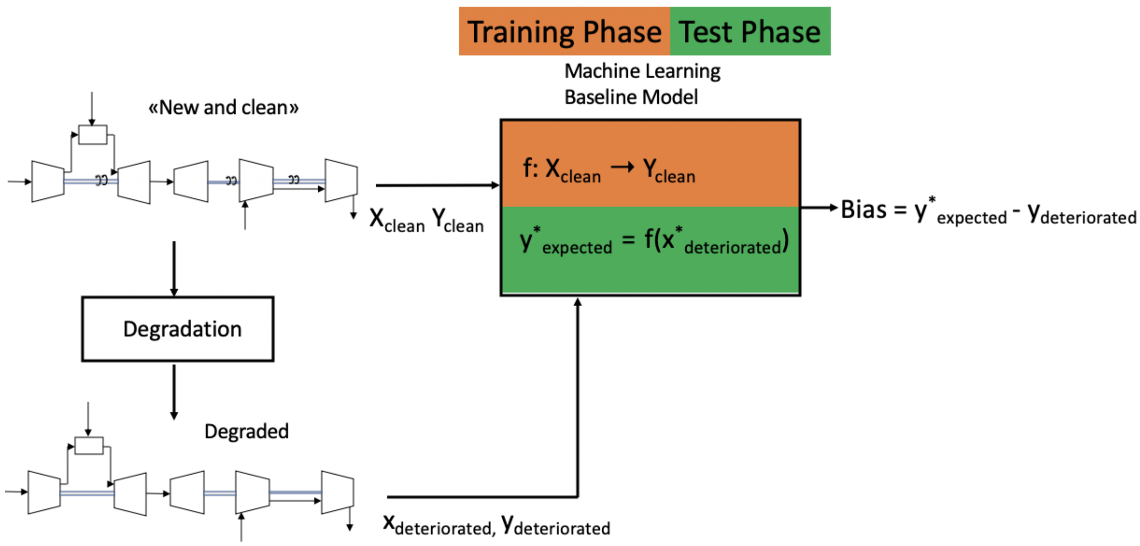



Figure 8: Scheme of the proposed methodology. Based on the dataset ($X_{clean}$,$Y_{clean}$) the model is trained (f: $X_{clean} \rightarrow Y_{clean}$). The model is tested on a dataset ($x_{deteoriated}$,$y_{deteoriated}$) from a deteoriated gas turbine from where the expected output, $y^{*}_{expected}$ can be obtained. The bias between the predictions and true output ($y^{*}_{expected}$-$y_{deteoriated}$) will serve as a measure for the level of degradation.

In Section 3.1.1 the term "expected" referred to mean of the predictions generated by multiple passes in a Bayesian neural network. In the context of this methodology, "expected" is also the mean found by a Bayesian neural network. However, the term is emphasized because it refers to the prediction that is generated by the baseline machine learning model. The term "expected" is conceptually used because the machine learning will make predictions under the assumption that the gas turbine is working under new and clean operating conditions. That is, the machine learning model will predict the output which is *expected* if the gas turbine did not undergo performance deterioration.

It is also important to emphasize that the method differs from what is generally thought of as traditionally machine learning objectives. Typically, in machine learning, it is desirable to minimize the error between the true and predicted output. In contrast, the objective for this task is to predict variables which are expected if the gas turbine is clean, and a deviation between the predictions and true values are therefore both expected and the key to measure the degradation.

## 4.2 Training, Validation, And Test Sets

Splitting the dataset into a training, validation, and test set is a common practice in machine learning and is important to ensure good generalization, prevent overfitting, and provide high accuracy on unseen data. The training set is the dataset that is used to fit the data in the model. In other words, it is the data the machine learning model sees and learns from. The validation data is the dataset that is used to prevent overfitting and tuning the hyperparameters. That is, the validation set ensures the generalizability and improves the ability to predict on datasets which the model has not been trained upon. The test set is used to provide an unbiased evaluation of the final model. The machine learning model will therefore never see or adjusted according to the test set (Molin, 2019).

The challenge when choosing appropriate training, validation, and test sets are that the objective is to train the machine learning model under the assumption that the gas turbine is working under ”new and clean” operating conditions. Two difficulties will, therefore, arise when choosing appropriate training, testing, and validation set for this specific case. First, it must be determined when the gas turbine is assumed to be new and clean. Secondly, it will reduce the flexibility of choosing a training set of sufficient size. If the model is trained at a section where the gas turbine already is degraded, the model will, consequently, learn the input-output relations for a deteriorated gas turbine. Hence, the predictions of the expected output in the test set can be underestimated. Moreover, the input distributions of the training set must also be representative for the test set. If not, the model will easier overfit and the uncertainty will increase. Even though many papers suggest that fouling tends to be more severe during initial operation, manufacturers and vendors will usually define the gas turbine to be ”new and clean” for the first 100-250 hours after initial site commissioning (Noordermeer and Eng, 2019). After considerable operation hours, the closest a gas turbine can get to ”new and clean” is immediate after washing and major overhaul.

Based on the ”new and clean” definition of 100-250 hours and taken into consideration the training set must be of sufficient size, the first seven and three days (240 hours) of initial operating after washing and overhaul will be used for the training and validation set, respectively. In this way, the model will both be trained and validated on a data set that can, to a reasonable extent, be considered clean. The remainder of the time series segment will then be considered degraded and will be used as test set. To increase the size of the training set, previous training and validation sets will be merged into the current training set. However, training and validation sets from time series segments that are sampled after the current time series segment will not be merged into the training set. Even though the data are from the same gas turbine under similar operation conditions, making predictions with a model that is trained on future data can be misleading. The partitioning of training, validation, and test set for time series segments 1, 2, and 3 are illustrated in Figure 9.

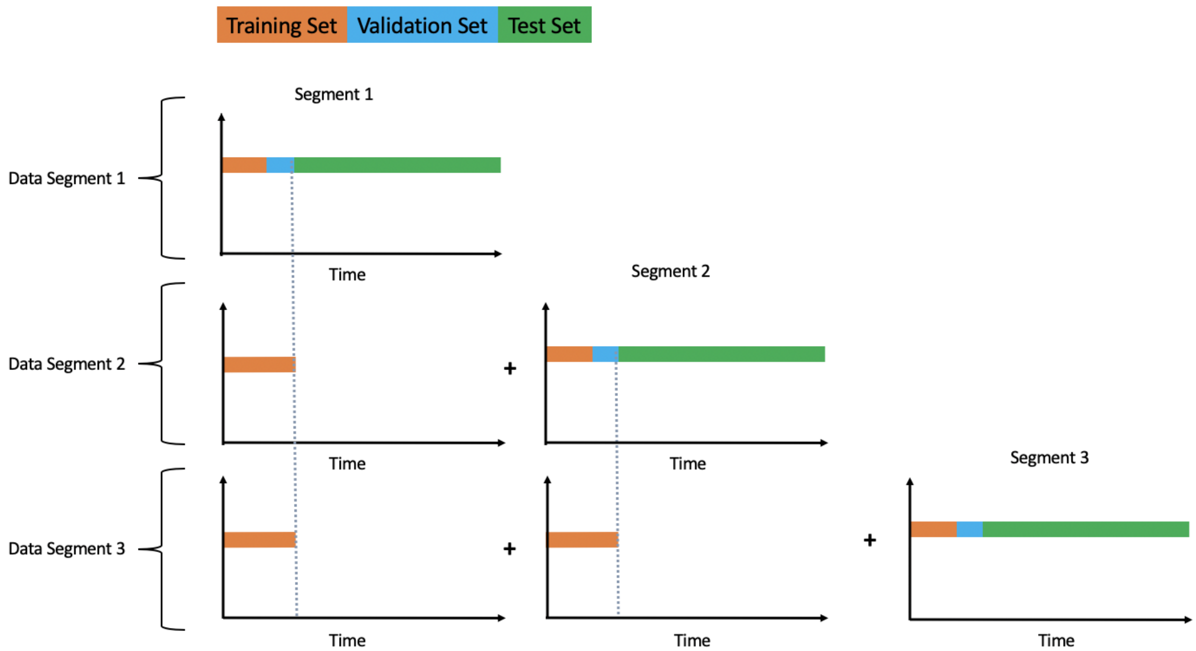



Figure 9: Selection of training, validation, and test set for time series segments 1, 2, and 3. Training and validation data sets from *previous* operation intervals will be merged into the current time series segment training set.

## 4.3 Evaluating The Uncertainty

An important part of the proposed methodology is to assess the uncertainty associated with the predictions. For condition monitoring of a gas turbine, evaluating the uncertainty is crucial due to two reasons. Evaluating the correctness and performance of a machine learning model is usually straightforward. For regression problems, the accuracy can be measured by how well the predicted values fit the true data. However, since the degradation is not directly measurable and is found from the bias, there is no fundamental way to describe the accuracy of the results. Therefore, uncertainty can severe as an indicator of whether the results can be trusted. Thus a large uncertainty would imply that human inspection is needed to revise the results. Secondly, evaluating the uncertainty is important because a system undergoing degradation will result in different training and test set distributions. For instance, if the power output in a gas turbine decreases due to degradation, this will also cause the rest of the system, such as throughput of gas, to drift. A gas turbine will, therefore, most likely enter different operating conditions in a degraded state. For simplicity, only the epistemic uncertainty will be assessed for this thesis.

## 4.4 Optimal Time Of Washing

Through the proposed method described in Section 4.1, the level of degradation expressed in terms of the health parameters can be obtained. Before the proposed method for determining the optimal time of washing is presented, the cost associated with maintenance and degradation which will be considered in this thesis will be covered. There are in general two factors that contribute to the economic loss caused by gas turbine performance deterioration. The first is the increased fuel costs. As the compressor deteriorates the gas turbine will compensate by increasing the fuel flow to reach the desired power output. The increased fuel flow caused by gas turbine degradation can be expressed as,

$$J_{\text{fuel}} = C_{\text{fuel}}(\dot{m}_{\text{fuel, true}} - \dot{m}_{\text{fuel, expected}}) \tag{19}$$

Where $C_{\text{fuel}}$ denotes the cost of fuel, $\dot{m}_{\text{fuel, true}}$ - the actual fuel rate and $\dot{m}_{\text{fuel, expected}}$ - the expected fuel rate if the gas turbine was new and clean.

Gas turbine deterioration loss must also address the value of the resulting decrease in power output. For offshore applications, the power shortfall translates to a decreased flow of gas moved through the pipeline. However, this assumes that the flow of gas through the compressor trains is the bottleneck. That is, the flow of gas in the compressor train can, to a reasonable extent, be increased if the power output from the gas turbine is increased. The cost associated with the reduced throughput of gas can be expressed as,

$$J_{\text{gas}} = C_{\text{gas}}(\dot{m}_{\text{gas, true}} - \dot{m}_{\text{gas, expected}}) \tag{20}$$

Where $C_{\text{gas}}$ denotes the cost of the produced gas, $\dot{m}_{\text{gas, true}}$ the actual gas rate and $\dot{m}_{\text{gas, expected}}$ the expected throughput of gas if the gas turbine was new and clean.

The maintenance cost must also be included and incorporates the fixed and variable costs, such as labor expenses and cleaning supplies, and cost due to downtime. Since the fixed and variable costs of washing are not available for this project, only the loss associated with downtime production loss will be considered. The average maintenance costs due to downtime can there be expressed as,

$$J_{\text{wash}} = \bar{m}_{\text{gas}} C_{\text{gas}} \bar{t}_{\text{downtime}} \tag{21}$$

Where $\bar{m}_{\text{gas}}$ denoted the average production of gas and $\bar{t}_{\text{downtime}}$ the average maintenance downtime.

It is assumed that the optimal time of washing can be found as the intersection between the total loss in fuel consumption and throughput of gas, and the cost associated with downtime. The optimal time of maintenance is therefore when,

$$J_{\text{gas}} + J_{\text{fuel}} = J_{\text{wash}} \tag{22}$$

Since the fuel and produced gas costs are not available for this thesis, they are assumed to be equal. Thus, the optimal time of washing can be simplified to,

$$\bar{m}_{\text{gas}}\bar{t}_{\text{downtime}} = (\dot{m}_{\text{fuel, true}} - \dot{m}_{\text{fuel, expected}}) + (\dot{m}_{\text{gas, true}} - \dot{m}_{\text{gas, expected}}) \tag{23}$$

Based Equation 23, there are two key terms that must be determined: the expected fuel consumption $\dot{m}_{\text{fuel, expected}}$ and the expected throughput of gas in the compressor train, $\dot{m}_{\text{gas, expected}}$. The expected fuel consumption can be obtained by the proposed method from Section 4.1. Conceptually, the same method could be used for finding the expected throughput of gas. However, it is desirable to use an approach that makes the effect of performance deterioration on gas throughput obvious for the machine learning model. Therefore, the proposed method involves utilizing the expected power output which can also be found by the method in Section 4.1. The idea is to train a machine learning model with the throughput of gas as the target variable. As shown in Figure 10, the machine learning model will be trained based on the actual power output. Then, to find the expected throughput of gas, the expected power output will replace the actual power output as an input to the machine learning model. The goal is then to predict the throughput of gas if the power output was not affected by degradation. The bias between the actual and expected throughput of gas can then be used as a measure for the lost production of gas.

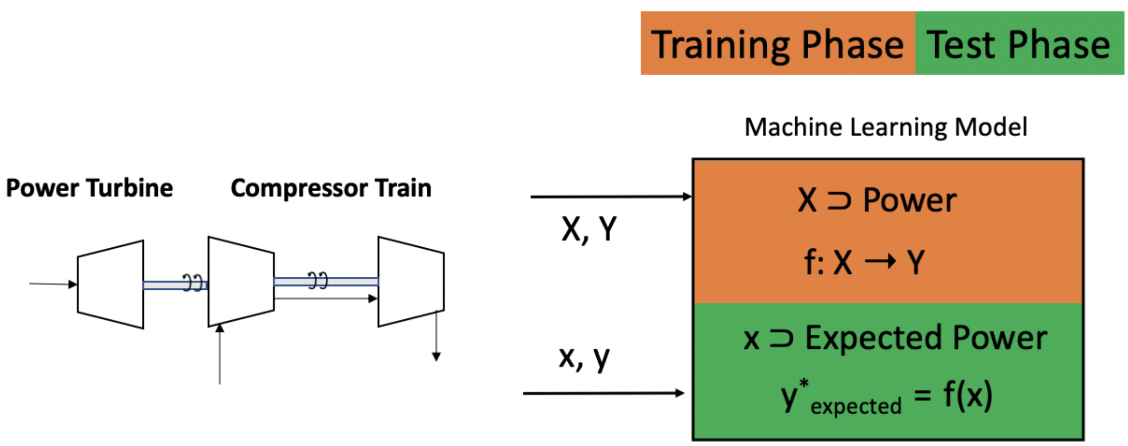



Figure 10: Scheme of the proposed methodology. Based on the dataset (X,Y) where the actual power output is a subset of the dataset (X ⊃ power), the model is trained (f: X → Y). The model is tested on a dataset (x,y) with the expected power output as a subset of the dataset (x ⊃ expected power).

## 4.5 Machine Learning Model, Preprocessing, And Tuning Pipeline

The machine learning model which will be used for this thesis is the Bayesian LSTM as introduced in Section 3.1 and is tuned with the pipeline described below. The implementation of Bayesian LSTM can be found in Appendix D. A machine learning pipeline describes the steps in order to train a model. The pipeline proposed by Bikmukhametov and Jäschke (2020) is used for tuning the model and is presented in Figure 11. First, the training, validation, and test set are selected. Then the data is preprocessed to transform the raw data into useful information which the machine learning model can interpret. Preprocessing will comprise handling missing values, smoothing, and scaling the data. Background for preprocessing is covered in Appendix B. Then, the hyperparameters are tuned. In this thesis, the hyperparameters are tuned with Bayesian hyperparameter optimization as described in Section 3.3. The learning rate, hidden layers, and dimension of the hidden state are tuned for the machine learning model. The dropout probability was set to 0.1. Initially, the sequence length for each forward pass was also tuned. However, since the optimal sequence length always produced the same result is was discarded as a tuned hyperparameter. When the optimal set of hyperparameters is obtained, the machine learning model is reinitialized and trained with early stopping to avoid overfitting.

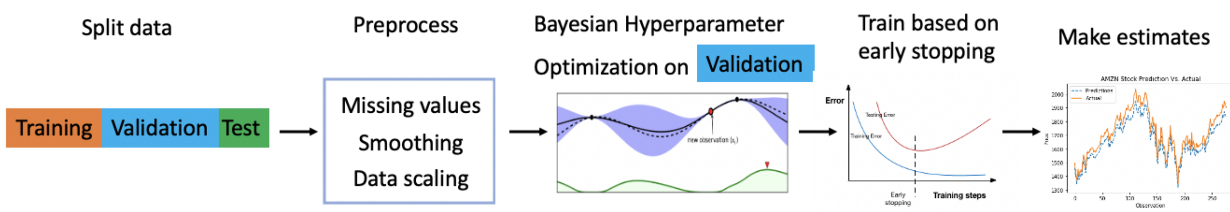



Figure 11: Machine learning preprocessing and tuning pipeline proposed by Bikmukhametov and Jäschke (2020).

# 5 Case Study - Condition Monitoring

This case study will investigate the effect of degradation based on three health parameters using the proposed method. The health parameters which will be considered are the power output, compressor discharge pressure, and fuel consumption. Both the measurements and the time axis presented in the case study are scaled. The time axis is scaled to similar magnitudes as the original unit and will still, for simplicity, be referred to in terms of the unit day. For all case studies, the results from the test set will be shown. Since the *actual* size of the training and validation sets are given, the time axis for the results will start at zero. The effect of degradation will be studied for all three operating intervals. Each section will include the reasoning for the input features, results, and discussion. The case study will be structured as described below.

The case study will be initiated with a **preliminary discussion** on machine learning considerations of selecting health parameters. It will be introduced desirable aspects regarding the bias and uncertainty. Then, it will be discussed dataset characteristics and health parameters that can fulfill these aspects.

The first case will consider **power output** as a health parameter. The preliminary discussion will be taken into consideration for evaluating the results. The uncertainty for all operating segments will also be discussed in terms of training and test set distributions.

The second case will examine **compressor discharge pressure** as a health parameter. Again, the results will be considered in terms of the preliminary discussion. Since the input features as similar to the power output, the uncertainty in terms of distributions will not be addressed. Instead, the section will focus on how the compressor discharge pressure compares to the power output as a health parameter. Moreover, consistency between the degradation levels for the expected power output and discharge pressure will also be considered.

The last case will consider the **fuel rate** as a health parameter. Similar to the other cases, the results will also be presented in light of the preliminary discussion. Since the input features are different from two latter-cases, the resulting effect on uncertainty will be discussed.

After the case study, the results will briefly be summarized and compared with other research. Possible weaknesses in the methodology will also be covered. Finally, a concluding discussion on the comparison of the health parameters and operating intervals will be presented.

## 5.1 Machine Learning Considerations For Degradation Monitoring

As outlined in Section 2.4.2, several health parameters have been proposed for performance monitoring. Moreover, certain criteria that are desirable in a health parameter and specific variables in a gas turbine that fulfilled these requirements were presented. For machine learning purposes, these criteria will, in general, also be relevant. The exception is, however, the effect of the load and ambient conditions on the health parameter. With machine learning methods, ambient and load measurements can easily be included as input features thus making the model able to differentiate between external fluctuations and degradation. However, machine learning methods do, fundamentally, have limitations that should be taken into consideration when selecting health parameters and evaluating the results. Until now, little importance has been given to the selection of health parameters from a machine learning aspect. This section will, therefore, aim to discuss the desired characteristics of predictions in terms of bias and uncertainty.

Based on the proposed method, three trends are desirable for the machine learning predictions and biases for performance monitoring,

- The predicted values should follow the governing dynamics of the system.
- The bias of the predicted values should follow a physically meaningful trend.
- The predicted expected values should be associated with a small uncertainty.

The expected values should follow the governing dynamics of the system. That is, even though the predicted values are biased due to performance deterioration, it should follow the same trends as the true values. To achieve this behavior, the target variable should be well correlated with the input features. Therefore, changes in the health parameter must also be reflected in other parts of the system. For this reason, variables that can be considered as disturbances in a system are unsuitable as health parameters. Even though these disturbances will be accounted for by other variables it will not be reflected in the overall dynamics.

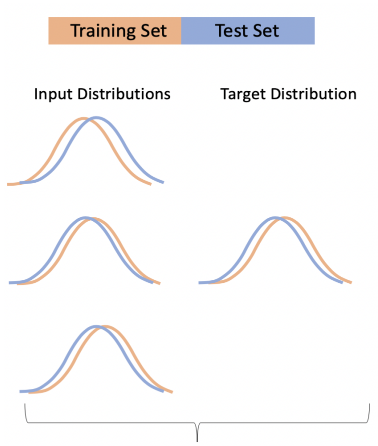
Since the bias conceptually describes the degradation, it should also follow a physically meaningful trend. Ideally, the bias should either increase or remain steady during the operating interval. A steady bias would imply a stabilized level of degradation and an increasing bias would correspond to an increasing level of degradation. An undesired trend a fluctuating bias which, in a physical sense, implies the deterioration to reverse.

An important requirement is that the predictions should be associated with small uncertainty. In terms of health monitoring, a small uncertainty is essential since the residual between the predicted and true values can be marginal. Therefore, it is important to measure this bias with as high certainty as possible. Also, since the bias can be marginal, even a small uncertainty in the predicted values could cause the confidence interval to intersect the true values. This is a problem because even though the average predictions suggest the presence of performance deterioration, it is not possible to determine with total confidence whether there is degradation in the system at all. Moreover, since the costs associated with performance deterioration is large, it is desirable to estimate the level of degradation accurately to ensure cost-optimal maintenance decisions.

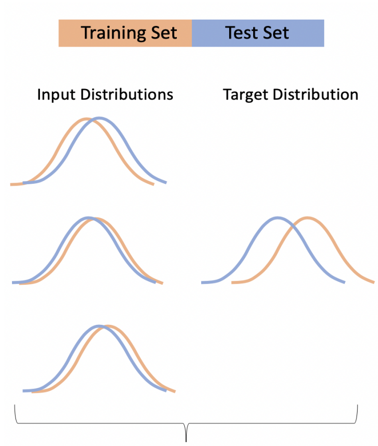
As presented in Section 3.2, three requirements should be fulfilled to reduce uncertainty.

- Each input feature for the training and test data set should be mutually overlapping.
- The data should be characterized by a low level of noise.
- Suitable selection of model parameters and structure.

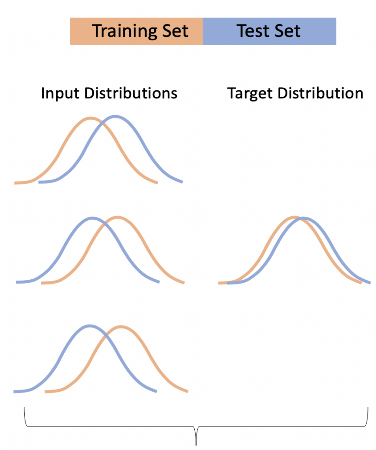
An important requirement in machine learning is that the distribution of the training set should be representative of the test set. If the condition is not fulfilled, the predictions will fluctuate thus resulting in large uncertainties. However, the challenge is that machine learning-based degradation monitoring is a fundamental uncertain problem in terms of operating conditions. For instance, consider the power output as a target variable, and fuel flow rate as one of the input features. When the level of gas turbine deterioration increases, the system will counteract by increasing the overall fuel flow consumption. Since the fuel flow is overall increased, the test set distribution, where the gas turbine is assumed to be deteriorated, will be different from the training set, where the gas turbine is assumed to clean. In terms of degradation monitoring, the training and test distributions for input and target variables can, conceptually, be divided into three scenarios as illustrated in Figure 12a, 12b and 12c.




(a) Scenario 1: mutually overlapping training and test set distributions.




(b) Scenario 2: mutually overlapping training and test set input distributions.




(c) Scenario 3: mutually overlapping training and test set output distributions.

Figure 12: Three possible configurations of the input and output training and test set distributions.

Figure 12a represents, in general, the most desirable input and output distributions for machine learning tasks. The input distributions of the training and test set are mutually overlapping, which means the model will be tested on a similar distribution as it was trained upon. In terms of degradation, however, this would imply no performance deterioration since the same input corresponds to the same output in both training and test set.

On the contrary, both Figure 12b and 12c suggests a relative shift between the training and test sets. Figure 12b represents a system that for the same set of input corresponds to a shifted output in the training and test set. As explained in Section 4.1 this could correspond to a gas turbine working full-load or close to an operating constraint. From a machine learning perspective, this would be the optimal set of distributions for detecting degradation. The input distributions are overlapping thus resulting in a small uncertainty while the target variables are shifted thus resulting in a measurable bias. For gas turbines operating at part-load, it could also be possible to achieve this distribution by selecting a manipulated variable as the target variable, whereas the input variables should be controlled variables. By transferring variables that tend to drift, thus yielding dissimilar training and test distributions, from the input feature set to the target variable, could reduce the uncertainty. For gas turbines, the manipulated variables will usually be the fuel flow rate.

In terms of degradation, Figure 12c represents a case where the input variables in the test set are adjusted to keep the target variable constant. As outlined in Section 4.1 this would correspond to a gas turbine working part-load and far from the operating constraints. For machine learning purposes this would result in a higher uncertainty compared to Figure 12b. In this case, the target variable in the machine learning model will correspond to a controlled variable. For gas turbines, this could for instance be the power output.

Since the performance of a gas turbine usually is a result of several interactions, the distributions of real datasets will be a combination of both Figure 12b and 12c. In addition, the effect of degradation is also relatively small, and it can, therefore, be difficult to pinpoint the shift in the distribution that is caused by degradation. Moreover, the behavior will also be dependent on ambient and external conditions. Consequently, skewed training and test set distributions are also caused by other factors than degradation. The training and test set distributions presented in the following case studies will, therefore, not be analyzed in terms of the shifts caused by degradation. Instead, it will be used for identifying different sources of uncertainty and for suggesting different approaches to possibly reduce the uncertainty.

## 5.2 Level Of Degradation Based On Power Output

The power output is one of the most researched health parameters. In the context of health monitoring, it is usually the power output from the turbine to the axial compressor which is studied. As this measurement is not available, the power output from the power turbine will be considered instead. Figure 23 shows the machine learning input features, in black, and the target variable, in green. Moreover, the

1 st and 2 nd stage compressors inlet and outlet variables will also be included as input features since the power output will depend on the throughput, pressure and temperature conditions of the gas. This will also contribute to improving the machine learning model's ability to differentiate between whether the variables of the system are manipulated to compensate for degradation or due to a different load in the compressor train. In other words, the measurements from the 1 st and 2 nd stage compressors must be included to determine the *relative* decrease in power output.

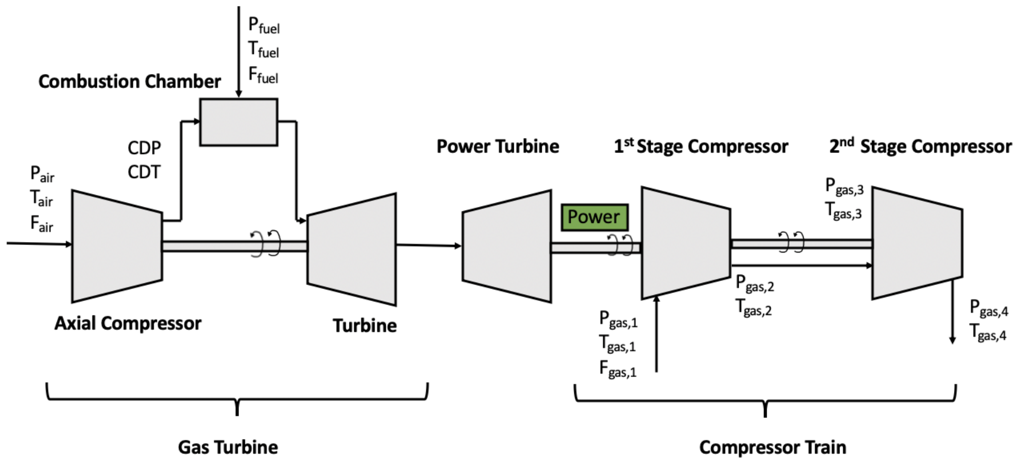



Figure 13: Flow-sheet of the process. The features used as input in the machine learning model are shown in black. Power output is the target variable, as indicated in green.

### 5.2.1 Segment 1

To find the expected power output for time series segment 1, the machine learning model was trained and validated on the first seven and three days, respectively. Figure 14 shows the results for the test set. The above plot shows the true and expected power output as well as the confidence interval. The lowermost plot shows the residual, given as percentage change, between the expected and true power output. Linear regression was used to find the best-fitting straight line for the residual to indicate the overall degradation rate.

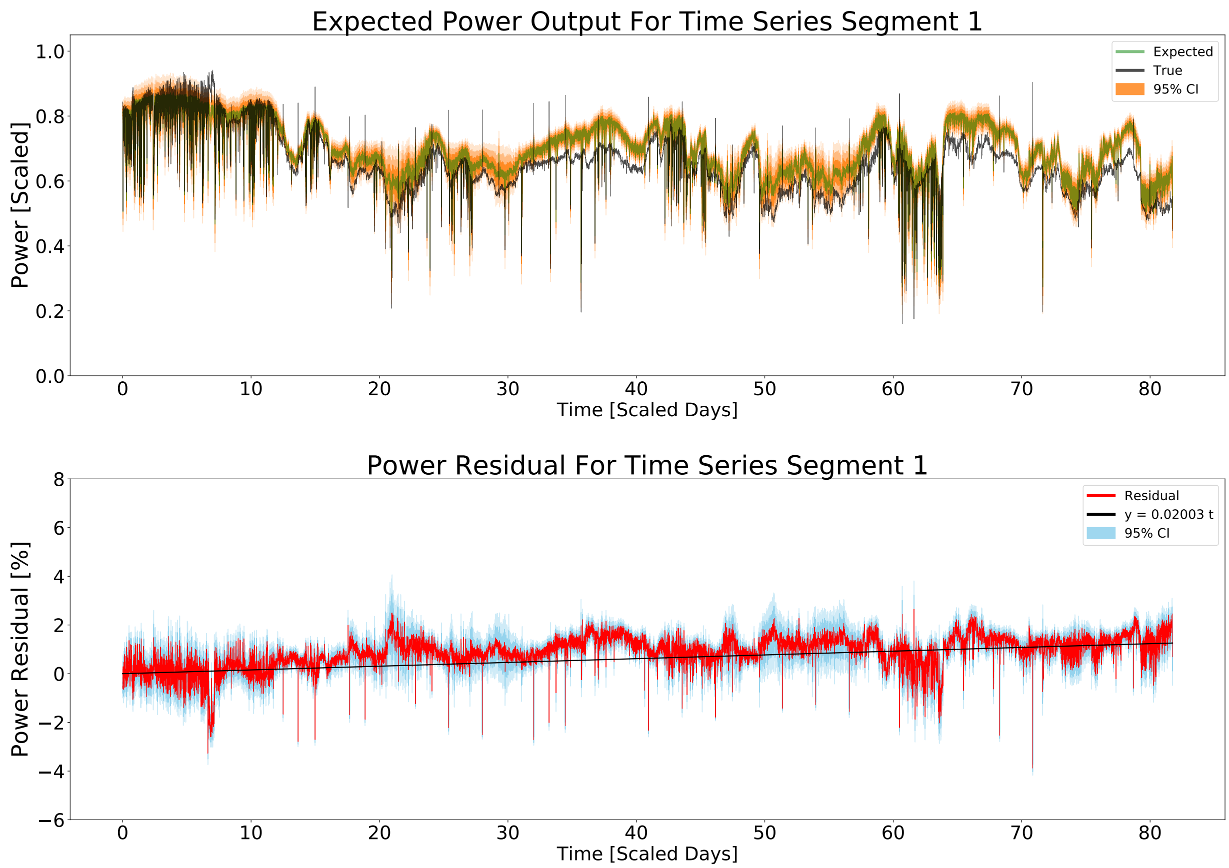
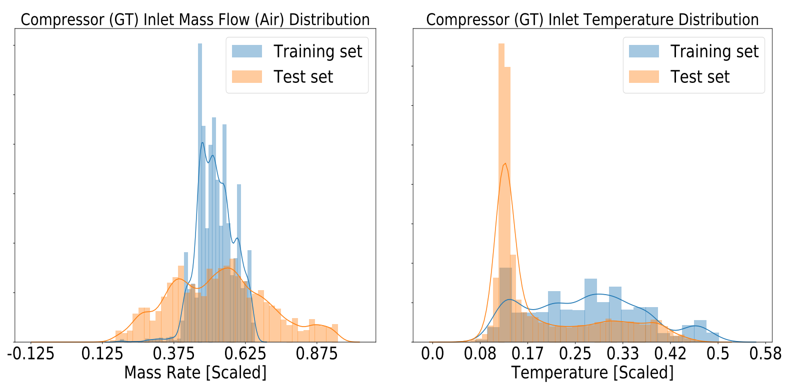



Figure 14: Level of degradation for time series segment 1 based on the power output from the power turbine. The actual and expected power output is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected power output.

The predictions capture the governing dynamical trends of the system. This indicates that the power output is well-correlated with the process and thus also the machine learning input features. Moreover, the expected power output is larger compared to the power output. In other words, if no degradation was present, the power output would be higher. The bias also shows a somewhat increasing trend which suggests an increasing degradation rate. Based on the residual function, the degradation will, on average, cause a drop in power output by 0.020% each day. Overall, the total decrease in power output caused by degradation can, therefore, be estimated to 1.635%.

For certain sections of the time series segment, the confidence interval is wider, and the expected power output tends to intersect the true power output. This is an issue since the intersection implies that there is a probability of no degradation at all. In other words, the machine learning model can in some cases predict the expected power output to be as good as the true output. On the other hand, a large uncertainty could also indicate that the degradation is more severe. Areas with large confidence intervals should, therefore, be interpreted with caution. However, during the last days (day 70-80), the uncertainty decreases compared to the rest of the operating interval. The evidence suggests that the degradation at the end of the operating time can be determined with larger confidence, and should be emphasized.

To explore the source of the uncertainties caused by a non-representative training set, the *input* distributions will be examined. Due to a large number of input features, only a selection of the features will be presented. Figure 15 shows the training and test set distributions of the inlet mass flow of air and ambient temperature. The range of the test set for the distributions tends to extend from the training set. In terms of the ambient temperature of the air, shown in Figure 15b, the fluctuations are due to natural variations in atmospheric conditions, and deviations are, therefore, expected. A possible solution to reduce the uncertainty is to discard input features that tend to drift. However, since the performance of the gas turbine depends on the ambient conditions, these features should always be included. In other words, the uncertainty caused by fluctuating ambient conditions must be accepted. The same argument can be made for the inlet mass flow of air, as shown in Figure 15a. Even though the distributions for the training and test set are dissimilar, the inlet mass flow of air is an important variable since it determines the efficiency of the gas turbine.

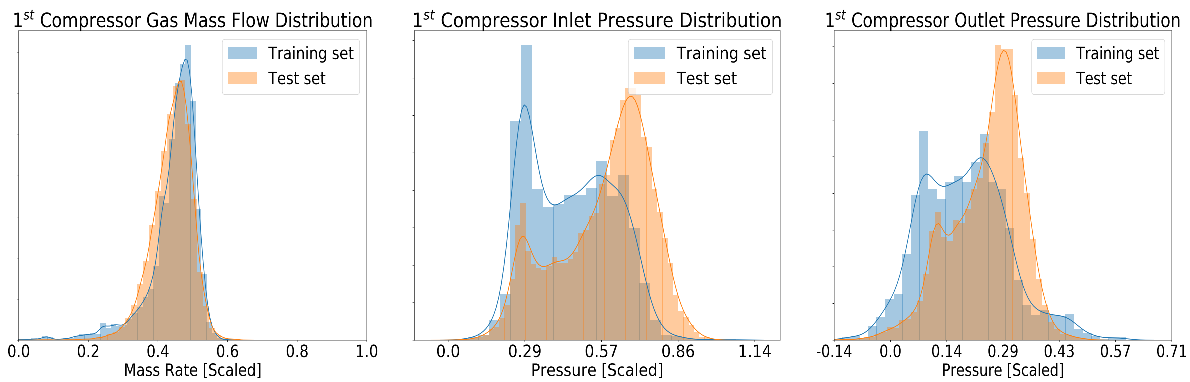



(a) Input distributions of inlet mass flow of air for time series segment 1.

(b) Input distributions of ambient temperature for time series segment 1.

Figure 15: A selection of training and test set distributions for the gas turbine in time series segment 1.

The training and test set distributions for the 1 st and 2 nd stage compressor are shown in Figure 16 and it is evident that the training set represents the test set well. Especially, is the inlet mass flow rate of the gas, as shown in Figure 16a, stable for the operating interval. Since the power output, and thus also other variables, in the gas turbine is dependent on the throughput of gas, this stability will be reflected in the other variables as well. Since the training set is small, a steady throughput of gas in the compressor train will, therefore, be a major advantage to achieve high prediction confidence. For the inlet and outlet pressure distributions, as shown in Figure 16b and 16c, the training set is also representative of the test set. Even not presented, the 2 $^{\text{nd}}$ stage compressor inlet and outlet conditions follow the same trend. Overall, a steady operation interval for time series segment 1 resulted in mutually overlapping distributions and is reflected as a relatively small uncertainty.




(a) Input distributions of inlet mass flow of gas for time series segment 1.

(b) Input distributions of inlet pressure of gas for time series segment 1.

(c) Input distributions of outlet pressure of gas for time series segment 1.

Figure 16: A selection of training and test set distributions for the compressor train in time series segment 1.

### 5.2.2 Segment 2

To find the expected power output for time series segment 2, the machine learning model was trained based on the proposed method in Section 4.2. The results are given in Figure 17. The first plot shows the true and expected power output and the confidence interval linked to the predictions. The lowermost shows the residual, given as percentage change, between the expected and true power output. Linear regression was used to find the best-fitting straight line for the residual to provide a rough indication of the degradation rate.

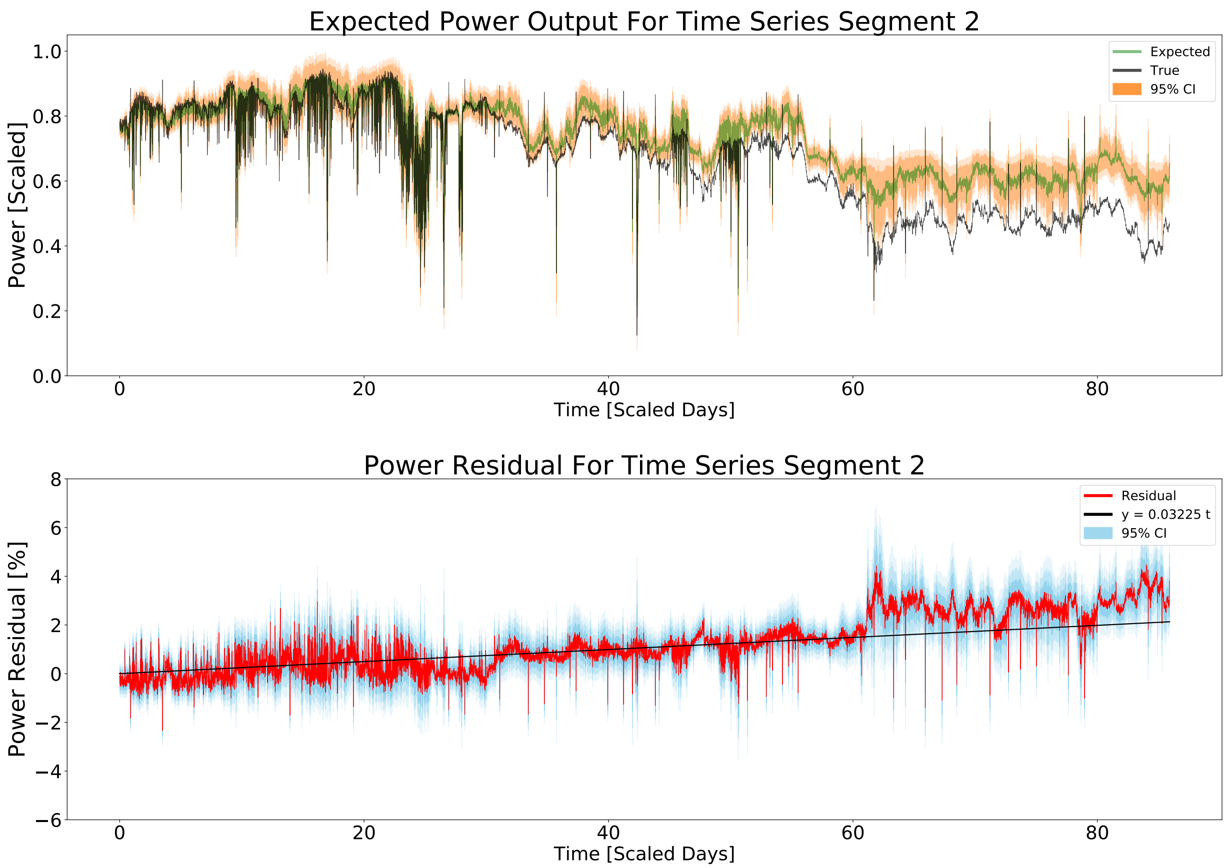
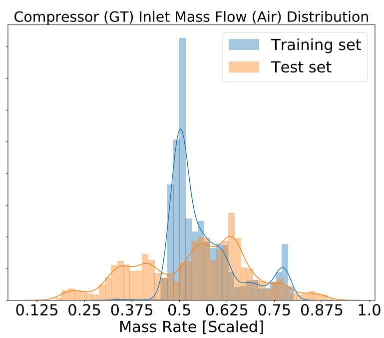


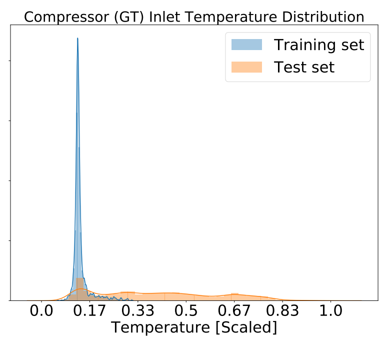

Figure 17: Level of degradation for time series segment 2 based on the power output from the power turbine. The actual and expected power output is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected power output.

Similar to the expected power output for time series segment 1, the predictions follow the governing dynamics of the system. In contrast, the bias of the expected power output also follows a more explicit trend, that is, the residual increases during the operating interval. Based on a physical interpretation, the degradation rate steadily increases. From the residual function, the degradation will, on average, cause a drop in power output by 0.032% each day. This is significantly higher compared to the first time segment. Overall, the total decrease in power output caused by degradation can, therefore, be estimated as 2.769%.

Compared to the rest of the operating interval, the uncertainty at the end is larger. However, the confidence interval does not intersect the true power output during the intervals with large uncertainty. Therefore, it can nevertheless be concluded that the system is deteriorating. When taken into consideration a low level of noise, which can contribute to increasing uncertainty, it can be argued that the uncertainty is due to an unrepresentative training set. Figure 18 and 19 presents a selection of input variables distributions that might contribute with increasing the uncertainty. Figure 18 shows some of the training and test set distributions linked to the gas turbine compressor. The mass flow of air, in Figure 18a shows similar characteristics as time series segment 1. The training and test set distributions for the ambient temperature in Figure 18b are very distinct. This will cause the uncertainties to increase. However, as previously discussed, the ambient temperature is important to include since the performance of the gas turbine is affected by atmospheric conditions.



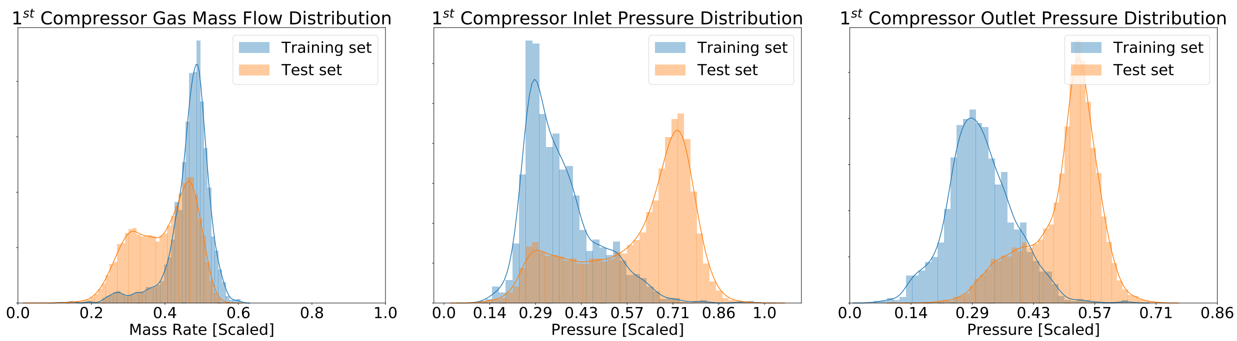

(a) Input distributions of inlet mass flow of air for time series segment 2




(b) Input distributions of ambient temperature of air for time series segment 2 set

Figure 18: A selection of training and test set distributions for the gas turbine in time series segment 2.

Figure 19 shows a selection of distributions related to the compressor train. Compared to time series segment 1, the distributions for the compressor train for time series segment 2, as shown in Figure 19, as less overlapping. To reduce the uncertainty, feature engineering should be considered for the 1 $^{st}$ and 2 $^{nd}$ stage compressor variables. As seen in Figure 19b and 19c, the test set distributions are shifted evenly compared to the training set. By combining the raw features into, for example, the pressure ratio would, therefore, push the training and test set distributions closer. Moreover, the power requirement in the compressor train is also proportional to the pressure ratio. Therefore, it can be argued that including physical meaningful features can represent the underlying pattern better and allow for the neural network to exploit relationships the raw data cannot directly provide. As a proof of concept, feature engineering for the expected power output for time series segment 2 will be covered in Appendix C.




(a) Input distributions of inlet mass flow of gas for time series segment 2

(b) Input distributions of inlet pressure of gas for time series segment 2.

(c) nput distributions of outlet pressure of gas for time series segment 1

Figure 19: A selection of training and test set distributions for the compressor train in time series segment 2.

### 5.2.3 Segment 3

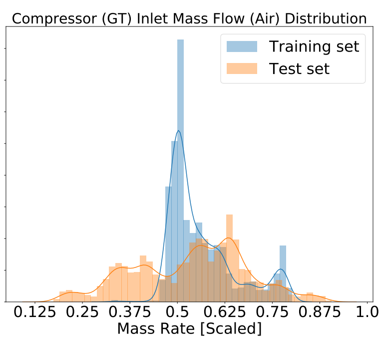
To find the expected power output for time series segment 3, the machine learning model was trained according to the proposed method in Section 4.2. Figure 20 presents the results. The above plot shows the true and expected power output as well as the confidence interval. The lowermost shows the residual, given as percentage change, between the expected and true power output. Linear regression was used to find the best-fitting straight line for the residual to indicate the overall degradation rate.


Figure 20: Level of degradation for time series segment 3 based on the power output from the power turbine. The actual and expected power output is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected power output.

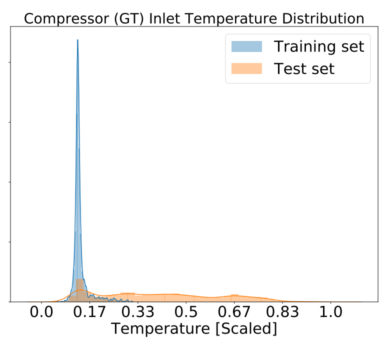
The expected values follow, to some extent, the dynamics of the system. However, compared to time series segments 1 and 2, the trend is not as obvious. It can be argued that the degradation is increasing since the expected power output is overall larger than the true rate. This is also evident by looking at the residual where the linear regression is defined by a positive inclination. The inclination is, however, not as large compared to the linear regressions for time series segments 1 and 2. This implies that the overall degradation rate is smaller. From the residual function, the degradation will, on average, cause a drop in power output by 0.015% each day. Overall, the total decrease in power output caused by degradation can, therefore, be estimated as 1.413%.

The bias is not consistent and will, consequently, lead to a fluctuating residual function. In terms of degradation, this result is not physically meaningful since it is highly unlikely that the degradation is improving during the operating interval. Moreover, under certain intervals, the expected power output also intersects the actual power output. It can, therefore, be argued that the results for time series segment 3 are less reliable. There are several possible explanations for these inconsistencies. Compared to both time series segments 1 and 2, the dynamics for time series segment 3 are more fluctuating. Since the machine learning model is trained on a small dataset, these fluctuating trends may not be captured in the training set. Since LSTMs use information from previous time steps, it could be argued that the *trends* in the training set are not representative of the *trends* of the test set.

Considering the confidence interval, the uncertainty does not follow an obvious trend. However, to some extent, the uncertainty tends to increase when the predictions deviate more from the true output. Figure 21 and 22 shows a selection of distributions of the training and test sets. Compared to both time segments 1 and 2, the distributions for the gas turbines, shown in Figure 21a and 21b, are more mutually overlapping. The same trends are also apparent for the variables in the compressor train, shown in Figure 21. Therefore, in terms of distribution coverage, the model should have a better basis for predicting the output. In addition, time-series segment 3, also has the largest training set. Therefore, a non-representative training set, in terms of distributions, can be excluded as the cause for poor results.

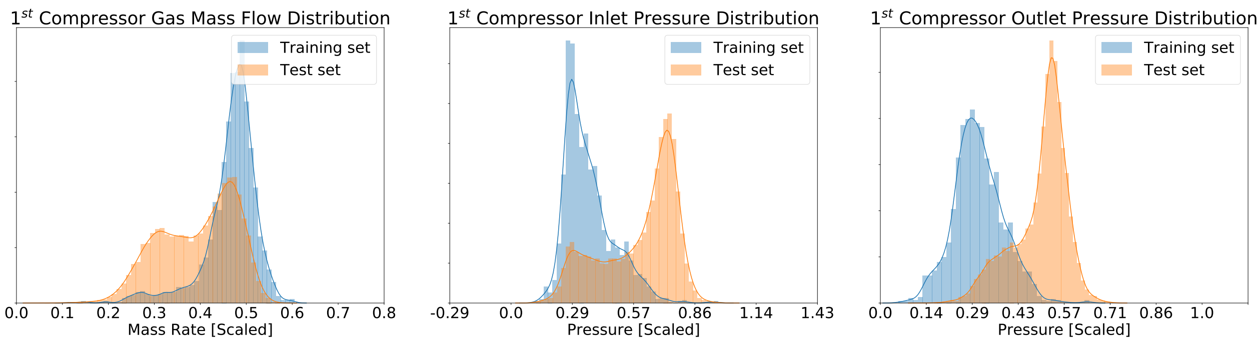



(a) Input distributions of inlet mass flow of air for time series segment 3




(b) Input distributions of ambient temperature of air for time series segment 3 set

Figure 21: A selection of training and test set distributions for the gas turbine in time series segment 3.




(a) Input distributions of inlet mass flow of gas for time series segment 3.

(b) Input distributions of inlet pressure of gas for time series segment 3.

(c) Input distributions of outlet pressure of gas for time series segment 1

Figure 22: A selection of training and test set distributions for the compressor train in time series segment 3.

Given that the model has good preconditions, improving the dynamical trends and uncertainty can also be handled by seeking a better machine learning model. An important question emerging from these findings is whether LSTMs are suited. Including long-term dependencies can be unnecessary if the dynamics are irregular. Therefore, a possible approach can be to use a different machine learning model, for instance, as a feed-forward network.

## 5.3 Level Of Degradation Based On Compressor Discharge Pressure

Compressor fouling will lead to a decreased compressor discharge pressure due to a reduction in stage efficiency (Giampaolo, 2013). The reduced stage efficiency is caused by increased frictional losses and internal recirculation. To counteract this effect, the power input to the axial compressor must also be increased. Thus, manipulated variables such as fuel flow and inlet mass flow will be adjusted accordingly to ensure the desired discharge pressure.

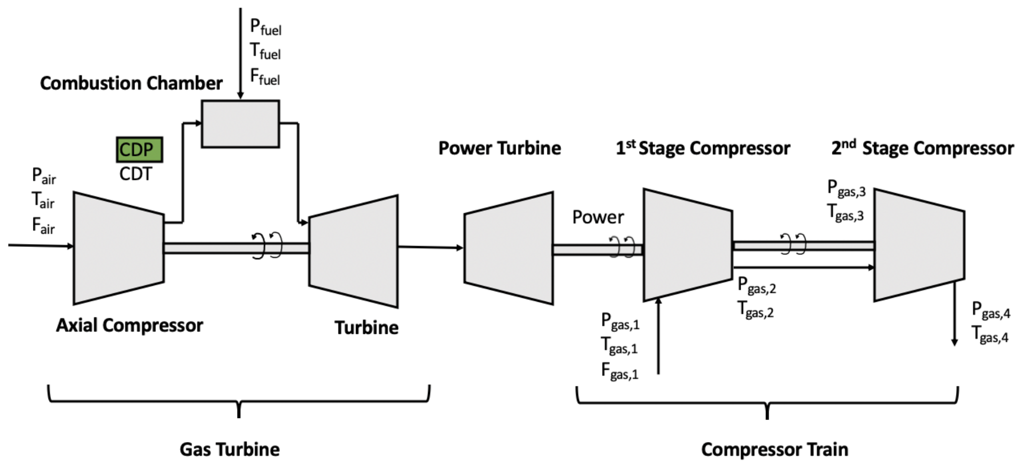



Figure 23: Flow-sheet of the process. The features used as input in the machine learning model are shown in black. Compressor discharge pressure is the target variable, as indicated in green.

### 5.3.1 Segment 1

To find the expected compressor discharge pressure for time series segment 1, the machine learning model was trained and validated on the first seven and three days, respectively. Figure 24 shows the results for the test set. The above plot shows the true and expected power output as well as the confidence interval. The lowermost plot shows the residual, given as percentage change, between the expected and true power output. Linear regression was used to find the best-fitting straight line for the residual to indicate the overall degradation rate.

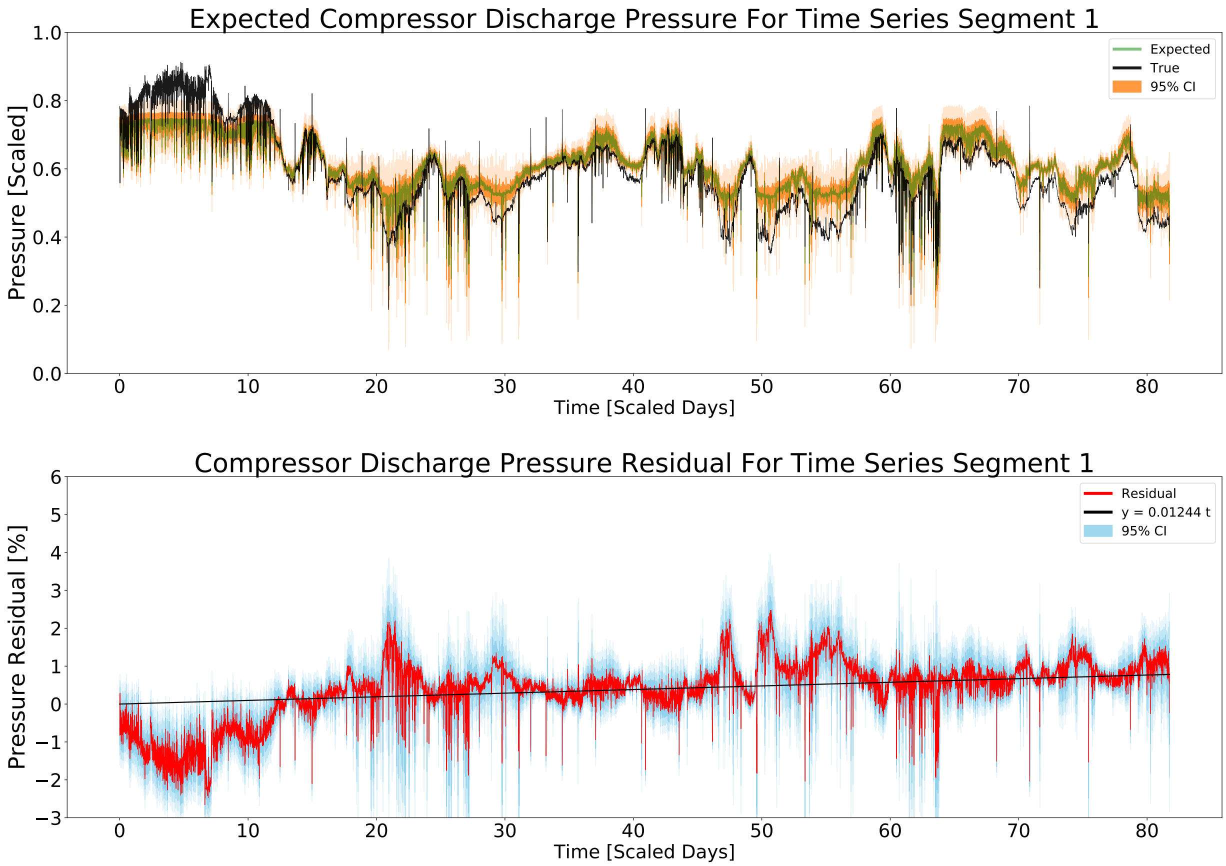



Figure 24: Level of degradation for time series segment 1 based on the compressor discharge pressure. The actual and expected compressor discharge pressure is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected compressor discharge pressure.

The compressor discharge pressure shows clear signs of performance deterioration. This bias of the predictions suggests, as expected, that the compressor discharge pressure should be larger under new and clean operating conditions. The expected compressor discharge pressure also follows the governing dynamics of the system. From the residual function, the degradation will, on average, cause a drop in compressor discharge pressure by 0.012% each day. Thus, at the end of the operating interval, the total loss in compressor discharge pressure can be estimated to 1.068%.

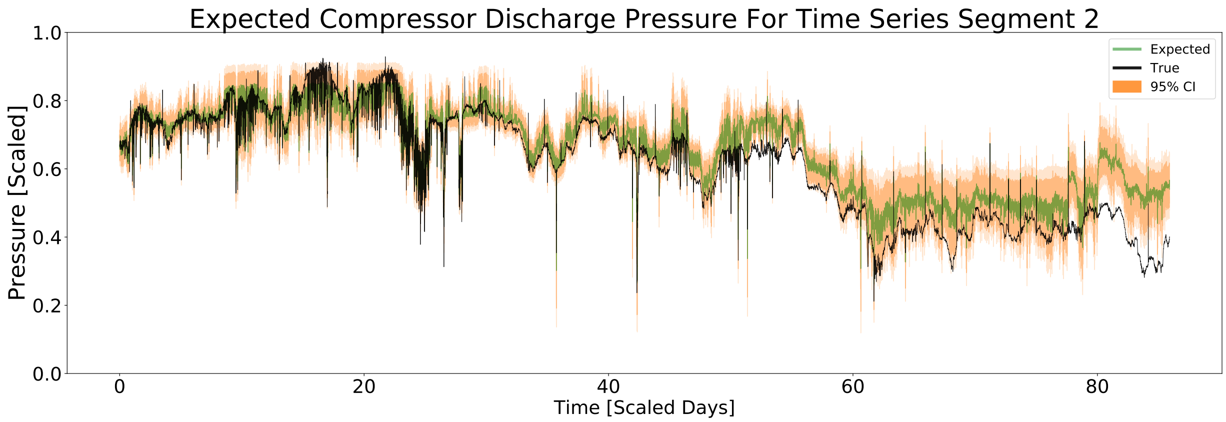
Dissimilar to the majority of the operating interval, the expected power outputs during the first period (day 0-10) are lower than the true output. This is also evident as a negative power residual. However, it is not physically meaningful that the gas turbine performance will improve relative to the first ten days. Because the performance of recurrent machine learning algorithms tends to decrease with highly fluctuating data, the inconsistency may be due to noisy data (Giles et al., 2001). Besides, if the data is noisy, the recurrent neural networks tend to emphasize short term-dependencies and can fall into the naive solution of always predicting the most common output which also seems to occur here.

Overall, the trends of the expected discharge pressure and uncertainties are consistent with the expected power output. Consistency between the health parameters is essential to determine the severity of degradation. Analyzing multiple health parameters simultaneously reduces the risk of, for instance, interpreting measurement drifts as degradation. Therefore, by observing the same trends in the expected discharge pressure and power output, the evidence of performance deterioration is stronger.

Similar to previous findings introduced in Table 4, performance deterioration has a smaller impact on the relative change in compressor discharge pressure compared to the power output. However, since the relative change is smaller, the bias is also more sensitive to measurement errors. Also, compared to the power output, the confidence interval has a larger tendency to cross the true discharge pressure. Thus, as discussed, the overall certainty for degradation in terms of discharge pressure decreases.

### 5.3.2 Segment 2

Figure 25 shows the results for the performance deterioration in terms of the compressor discharge pressure. The uppermost plot shows the true and expected compressor discharge pressure output as well as the confidence interval. The lowermost plot shows the residual, given as percentage change, between the expected and true power output. Linear regression was used to find the best-fitting straight line for the residual.



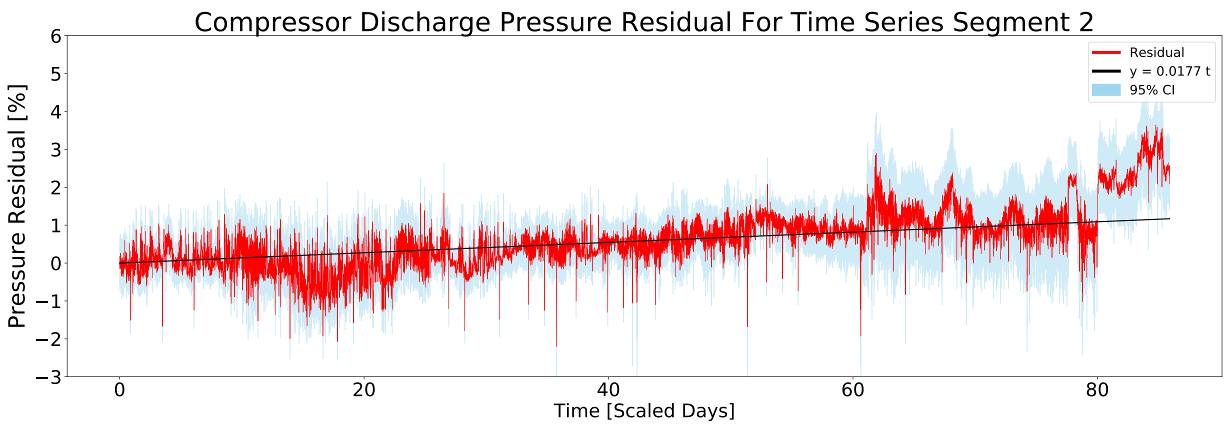




Figure 25: Level of degradation for time series segment 2 based on the compressor discharge pressure. The actual and expected compressor discharge pressure is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected compressor discharge pressure.

The expected compressor discharge pressure follows the dynamics of the true discharge pressure well. The bias is also consistent with the trend for the expected power output for time series segment 2. From the residual function, the relative loss due to degradation in compressor discharge pressure is 0.018% each day. Thus, at the end of the operating interval, the compressor discharge pressure is reduced by 1.520%.

The residual is larger compared to time series segment 1, which is consistent with the findings for the expected power output. Thus, both health parameters suggest that the level of degradation is more severe for time series segment 2. Since the input features are almost identical to the case with power output as a health parameter, the uncertainty for the expected discharge pressure is also, consequently, similar. However, since the effect of degradation is smaller on the compressor discharge pressure, the confidence interval shows a larger tendency to cross the true discharge pressure.

### 5.3.3 Segment 3

Figure 24 shows the results for the test set. The above plot shows the true and expected power output as well as the confidence interval. The lowermost plot shows the residual, given as percentage change, between the expected and true power output. Linear regression was used to find the best-fitting straight line for the residual to indicate the overall degradation rate.

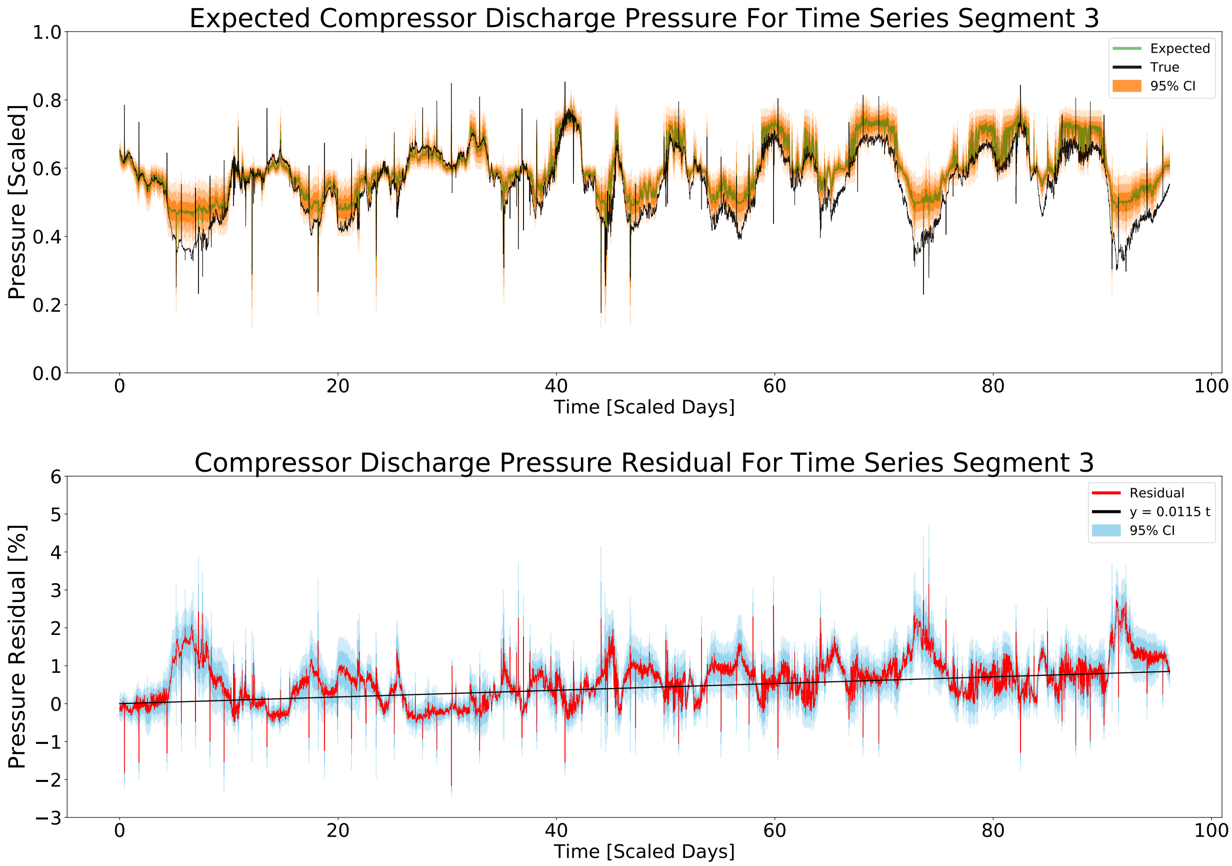



Figure 26: Level of degradation for time series segment 3 based on the compressor discharge pressure. The actual and expected compressor discharge pressure is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected compressor discharge pressure.

Similarly to the expected power output for time series segment 3, the machine learning model struggles with capturing the dynamics. However, the expected discharge pressure is overall larger compared to the true output and is evident by a positive slope for the linear regression. Based on the residual function, the compressor discharge pressure is, on average, reduced 0.012% each day. That is, at the end of the operation interval, the total loss in compressor discharge pressure compared to a new and clean gas turbine is approximately 1.071%.

Despite an inconsistent residual, there is consistency between the expected discharge pressure and power output. As found for the previous time segments, the loss in compressor discharge pressure is lower compared to the power output. Similarly, for time series segment 3, the residual function for the power output showed a reduction of 0.015% each day whereas the compressor discharge pressure is reduced 0.012% each day.

## 5.4 Level Of Degradation Based On Fuel Consumption

Several factors determine the overall fuel flow. Compressor fouling will result in a decreased airflow, which should decrease the overall demand for fuel. However, to compensate for the loss of compressor efficiency the engine will accelerate the shaft speed thus causing the fuel flow to increase. The input features used for predicting fuel consumption are shown in Figure 27. Unlike the previous case studies, the 1 st and 2 nd stage compressor variables will not be included as input features. Since the fuel consumption essentially will depend on the required power, including the 1 st and 2 nd stage compressor variables are not necessary. Fuel consumption as a health parameter can also add additional insights based on the preliminary discussion, in Section 5.1, concerning the selection of health parameters for data-driven condition monitoring. As briefly explained, from a machine learning perspective, monitoring manipulated variables with the proposed method can result in a lower uncertainty. From an economical perspective, determining increased fuel consumption can also be crucial for scheduling maintenance.

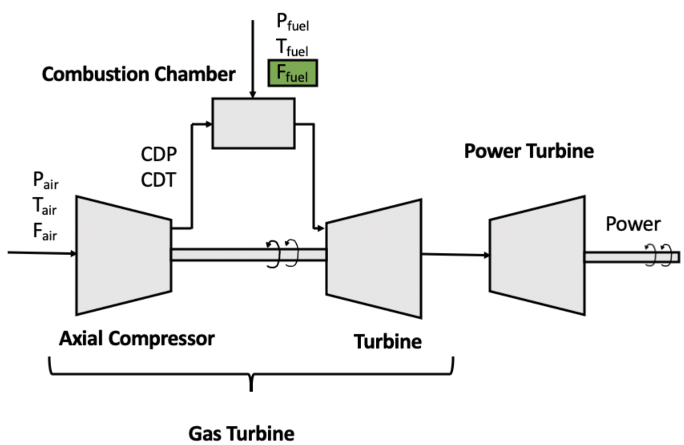



Figure 27: Flow-sheet of the process. The features used as input in the machine learning model are shown in black. The fuel rate is the target variable, as indicated in green.

### 5.4.1 Segment 1

To predict the expected fuel consumption for time series segment 1, the machine learning model was trained and validated on the first seven and three days, respectively. Figure 28 consists of two graphs.

The upper plot shows the true and expected fuel rate as well as the confidence interval. The lowermost shows the residual, given as percentage change, between the expected and true fuel rate. Linear regression was used to find the best-fitting line.

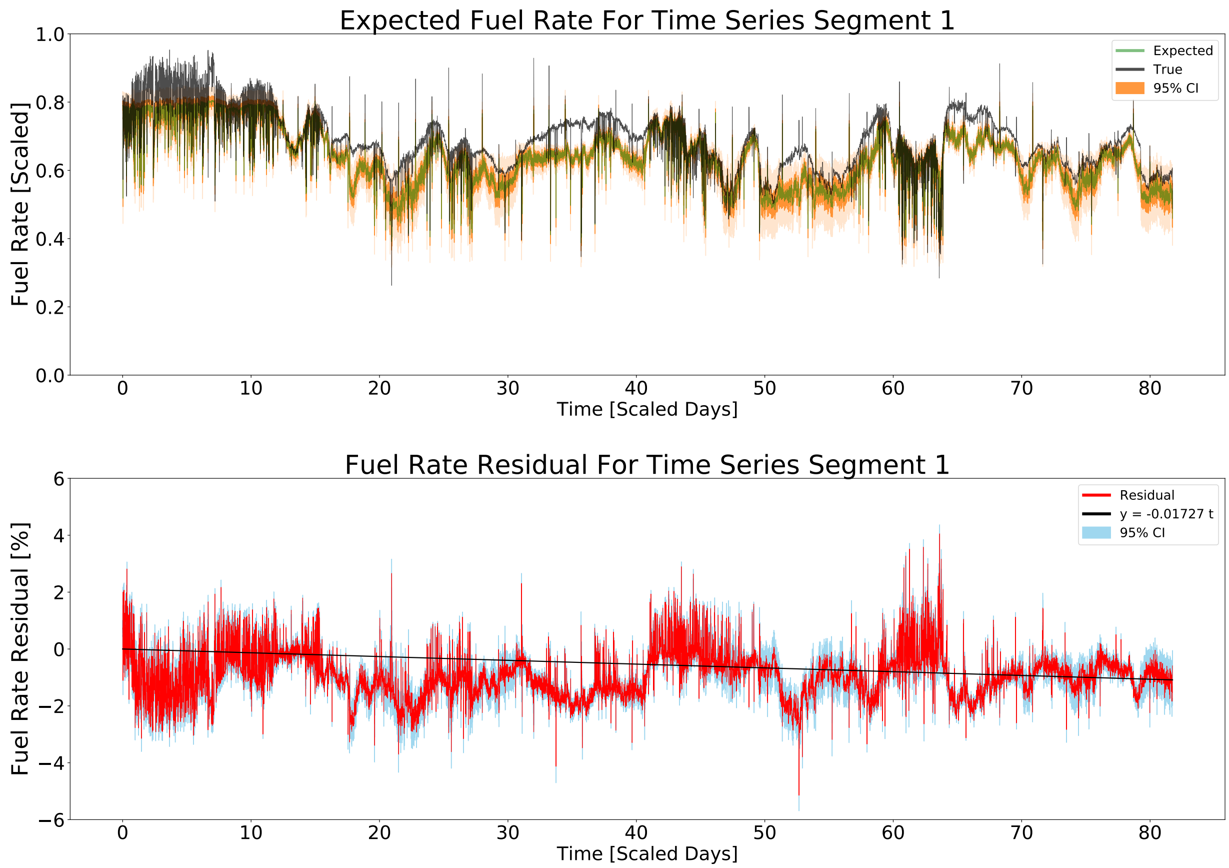



Figure 28: Level of degradation for time series segment 1 based on the fuel consumption. The actual and expected compressor discharge pressure is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected fuel consumption.

In contrast to the expected power output and discharge pressure, the expected fuel consumption is overall lower than the true rate. This indicates that the fuel rate needs to be increased to compensate for performance deterioration. The expected fuel consumption also follows the dynamical trends which suggest that even without the inclusion of the 1 $^{st}$ and 2 $^{nd}$ stage compressor variables, there is a good correlation between the input features and the fuel flow. The bias is, however, not as steady compared to cases with power output and compressor discharge pressure. This can be seen as a more unstable residual function. Considering the linear regression, the degradation will, on average, cause an increased fuel consumption of 0.017% each day. Thus, at the end of the operating interval, the total additional consumption of fuel can be estimated as 1.411%.

Compared to studies on time series segment 2, the uncertainty is overall lower. For instance, compared to the expected power output for time series segment 1, the confidence interval for expected fuel consumption rarely crosses the true rate. There are two factors with could contribute with an overall lower uncertainty. Firstly, the 1 $^{st}$ and 2 $^{nd}$ compressor variables are not included as inputs. As previously discussed, these variables were associated with dissimilar training and test set distribution. Therefore, excluding these features could provide an overall more mutually overlapping set of input features. However, this is not necessarily a generalizable advantage for the fuel rate as a health parameter since the distributions of the $1^{st}$ and 2 $^{nd}$ stage compressors will differ depending on the operating conditions. Secondly, as previously discussed in Section 5.1, using manipulated variables as the health parameters can, in terms of training and test set distributions, reduce the overall uncertainty.

### 5.4.2 Segment 2

Figure 29 presents the result for the expected fuel consumption for time series segment 2. The upper plot shows the true and expected fuel rate as well as the confidence interval. The lowermost shows the residual, given as percentage change, between the expected and true fuel rate. Linear regression was used to find the best-fitting straight line for the residual to indicate the overall degradation rate.

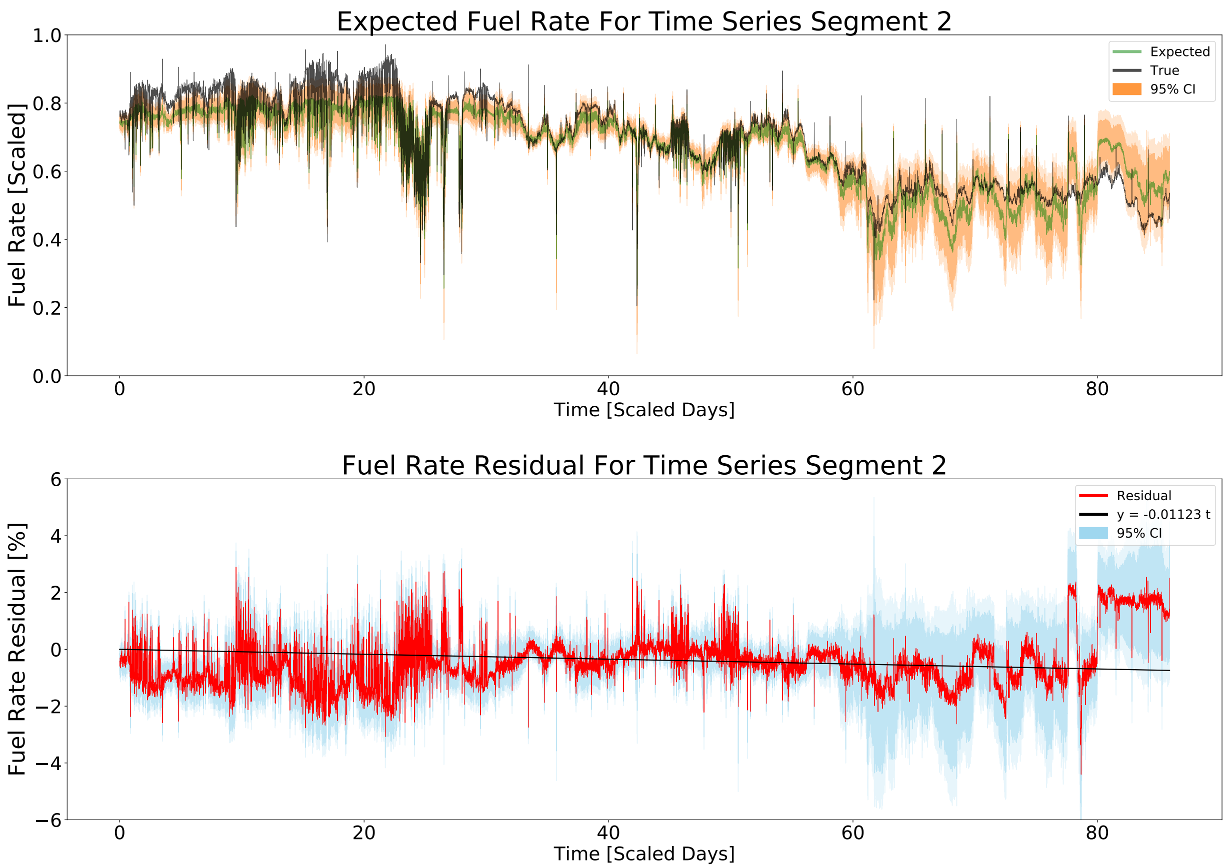



Figure 29: Level of degradation for time series segment 2 based on the fuel consumption. The actual and expected compressor discharge pressure is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected fuel consumption.

Overall, the expected fuel rate is lower than the true consumption. The trend of the residual is also consistent with the other health parameters for time series segment 2. That is, the bias is small at the start of the operating interval before it increases. Based on the linear regression of the residual plot, the relative increase in fuel consumption caused by performance deterioration is approximately 0.011% each day. Thus, at the end of the operating interval, the relative fuel consumption has increased to 0.964%.

In contrast to earlier findings, the level of degradation based on fuel consumption implies an overall lower deterioration compared to time series segment 1. This is particularly evident when considering the average decrease in power output for time series segments 1 and 2 of 0.020% and 0.032%, respectively. Whereas the average increase in fuel rate consumption for time series segments 1 and 2 are, 0.01727% and 0.011%, respectively. Ideally, the health parameters should yield consistent results for the level of degradation. This relative inconsistency between the health parameters for each time series may be due to several factors. First, the most apparent difference between the expected power output and compressor discharge pressure, and the expected fuel consumption for time series segment 2 is during the last part of the operating interval (day 60-80). For the expected power output and compressor discharge pressure, the end of the operating interval is generally characterized by a large bias. For the expected fuel rate consumption, however, the residual is relatively smaller at the end of the operating interval, thus reducing the overall level of degradation. As discussed, since the $1^{st}$ and $2^{nd}$ stage compressor variables are excluded, the model could have better preconditions in terms of the dataset. Consequently, it can be argued that the level of degradation based on the fuel rate perspective, is more accurate. Thus, the level of degradation measured in terms of power output and compressor discharge pressure can be overestimated. This also leads to the important conclusion that several health parameters should be taken into consideration when determining the overall level of performance deterioration.

### 5.4.3 Segment 3

Figure 30 presents the result for the expected fuel consumption for time series segment 2. The above plot shows the true and expected fuel rate as well as the confidence interval. The lowermost shows the residual, given as percentage change, between the expected and true fuel rate. Linear regression was used to find the best-fitting straight line for the residual to indicate the overall degradation rate.

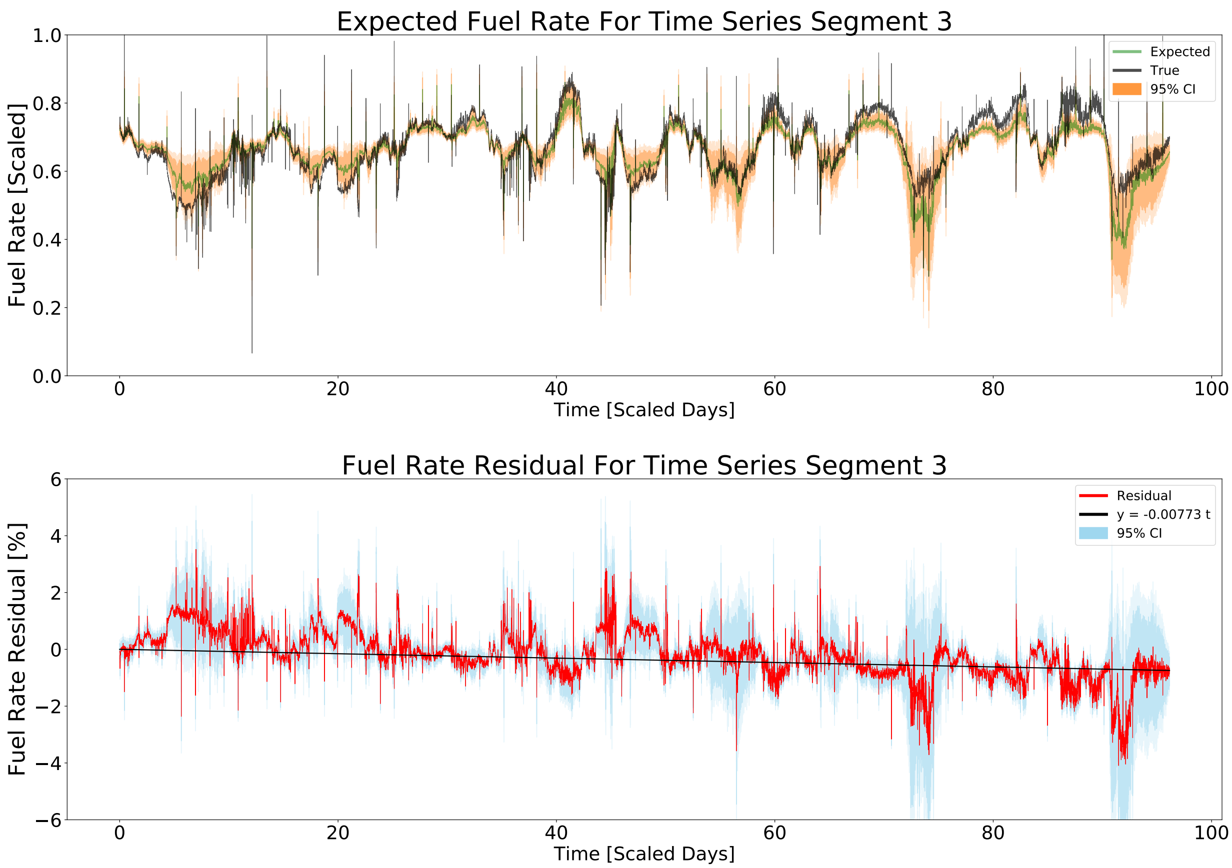



Figure 30: Level of degradation for time series segment 3 based on the fuel consumption. The actual and expected compressor discharge pressure is shown in black and green, respectively. A 95% confidence interval associated with the predictions are shown in orange. The lowermost plot shows the residual given as the percentage deviation between the true and expected fuel consumption.

Similar to previous findings for time series segment 3, the overall dynamics for the expected fuel rate tends to be inconsistent. However, by inspecting the residual, it is clear that the dynamics are slightly improved due to a smoother and less fluctuating function. Based on the linear regression, the fuel consumption is, on average, increased 0.077% each day to compensate for performance deterioration. Thus, the total additional fuel consumption at the end of the operating interval can be estimated to 0.743%.

The feature that differentiates the expected fuel rate from the expected compressor discharge pressure and power is the low uncertainty. As previously explained, the 1 $^{st}$ and 2 $^{nd}$ compressor variables are unnecessary for predicting the fuel rate and is therefore excluded as input features. Since these variables were associated with large uncertainty, removing these will consequently reduce the confidence interval. Moreover, there is also consistency between the expected fuel flow and the compressor discharge pressure for time series segments 2 and 3, where the relative additional fuel consumption was found to be lower compared to the reduction in discharge pressure.

## 5.5 Results Summary

Table 5 presents the total performance loss for the studied health parameters for time series segments 1, 2, and 3. The results are given as percentage deviation from the true measurements. The scaled size of the test set, i.e. where the gas turbine is assumed to be deteriorated, is also presented as a reference for the total level of degradation.

Table 5: Summary of the total level of degradation in terms of power output, compressor discharge pressure, and fuel consumption.

| Segment | Duration [Days] | Power Output | Compressor Discharge Pressure | Fuel Consumption |
|---|---|---|---|---|
| 1 | 81.74 | 1.635% ↓ | 1.068% ↓ | 1.411% ↑ |
| 2 | 85.89 | 2.769% ↓ | 1.520% ↓ | 0.964% ↑ |
| 3 | 96.13 | 1.413% ↓ | 1.071% ↓ | 0.743% ↑ |

## 5.6 Discussion

By comparing the results presented in Table 5 to previous findings, shown in Table 4, the relative degradation levels are somewhat similar. For all previous studies, it was found that the power output is the most affected health parameter which is consistent with all the findings of this case study. Giampaolo (2013) found that a 2% decrease in discharge pressure corresponds to an increased fuel rate of 3%. The findings for time series segment 1 agrees well with this result, where a 1.068% decreased compressor discharge pressure resulted in a 1.411% increase in fuel consumption. However, for both time series segments 1 and 3, the relative change in fuel consumption was found to be smaller compared to the reduction in compressor discharge pressure. However, Zwebek and Pilidis (2003) suggested that an 8% decrease in power output would imply a 2.5% increase in fuel consumption, thus a ratio of approximately 3. This result is consistent with both time series segments 2 and 3, with a power-fuel ratio of 2.9 and 2.1, respectively. Tarabrin et al. (1998) argued that the degradation rate is most prominent during the initial operation before it decays exponentially. However, the results from this case study suggest a stable or steadily increasing degradation rate.

All the health parameters investigated revealed clear signs of performance deterioration. Overall, the predictions captured the dynamics of the system and the bias followed a physically meaningful trend. Especially did the expected power output follow the trend accurately. This is expected since the power output is well correlated with both the 1 $^{st}$ and 2 $^{nd}$ compressor variables and the gas turbine parameters. It was also the health parameter that was most affected by degradation, and will thus be less affected by measurement error. However, as Haq and Saravanamuttoo (1991) stated, the health parameter should preferably be measured directly, which is not the case for the power output as it is calculated by the torque and shaft speed. Since the confidence interval only incorporated epistemic uncertainty, including measurement error will increase the actual uncertainty. Compressor discharge pressure, on the other hand, can be measured directly. However, the compressor discharge pressure is less affected by performance deterioration. The fuel rate was overall characterized by low uncertainty and followed the dynamical trends. This could support the hypothesis, addressed in Section 5.1, that using a manipulated variable as a target feature will decrease the uncertainty. To obtain an accurate measurement for the level of degradation it is important to investigate several health parameters to determine the overall degradation level and consistency. This will make the approach more robust to measurement errors and drifts.

There is, however, general limitations of a methodology based on machine learning. The type of degradation, whether it is fouling, erosion, or permanent damage, cannot be determined. That is, the method can only identify the presence and level of deterioration. This can pose a challenge when determining the optimal time of washing since only certain types of degradation, such as fouling, can be removed by washing. Another weakness in the approach is the assumption that the first 10 days of each operating segment can be considered new and clean. Several studies have concluded that degradation will start immediately after washing and maintenance is performed. The machine learning model will, therefore, be trained upon a data set that is, to some extent, sampled under a deteriorated state. Since the machine learning model, conceptually, only detects relative differences compared to the dataset it was trained upon, the bias can be underestimated. However, this is not necessarily a general weakness in the methodology, but rather due to a lack of training data. Ideally, the machine learning model should be trained upon several operating intervals, with a narrowed training set period. An increased training set could also contribute to decreased uncertainty. Moreover, the training set should also cover several dynamical trends. This could improve the dynamics of the predictions for a fluctuating dataset, such as time series segment 3.

# 6 Case Study - Optimal Time Of Maintenance

This case study will investigate the cost-optimal time for maintenance using the proposed method from Section 4.4. The increased fuel consumption due to degradation for each operating interval has already been determined in Section 5.4, and will be included in the costs. To find the reduced throughput of gas in the compressor train, the expected power output found Section 5.2 will be used as an input to a machine learning model. The residual between the actual mass flow and the expected mass flow will be considered as the loss of throughput due to gas turbine deterioration.

Two cases will be considered. The first considers the average loss in reduced power output whereas the second will examine the average loss plus one standard deviation. The first and latter case will represent the best and worst-case degradation rates. There are two reasons for selecting these cases. First, as previously discussed, the level of degradation can be underestimated due to possible performance deterioration in the training set. This would consequently also underestimate the throughput of gas. Taking basis in the average expected power output is therefore referred to as the best-case scenario. Since it is desirable to keep the recommended washing interval to a reasonable range, only one standard deviation will be selected for the worst-case. Figure 31 shows the inputs, in black, during the test phase for the best and worst-case, and the mass flow of gas, in green, as the target variable.

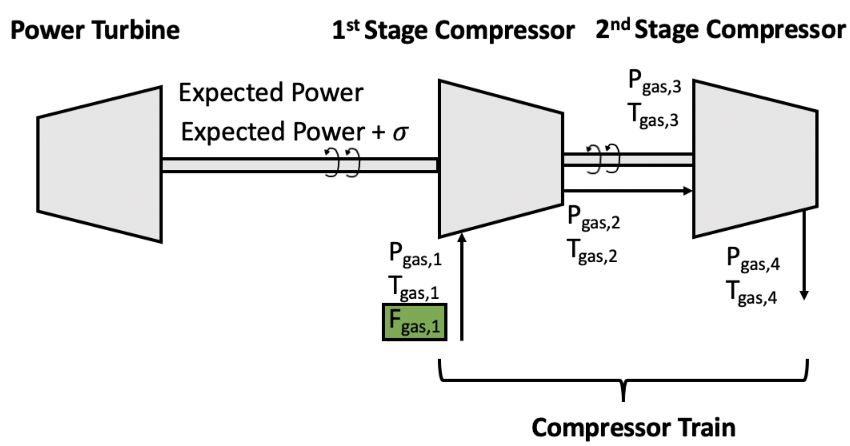



Figure 31: Flow-sheet of the process used to predict the expected throughput of gas in the compressor train. The features used as input in the machine learning model are shown in black. The mass flow of gas is the target variable, as indicated in green.

## 6.1 Segment 1

Figure 32 shows the results for the expected throughput of gas for the best and worst-case degradation rates. For visualization purposes, the results are smoothed. This intermediate result will, together with the average bias for fuel consumption, be used to find the optimal time of washing.

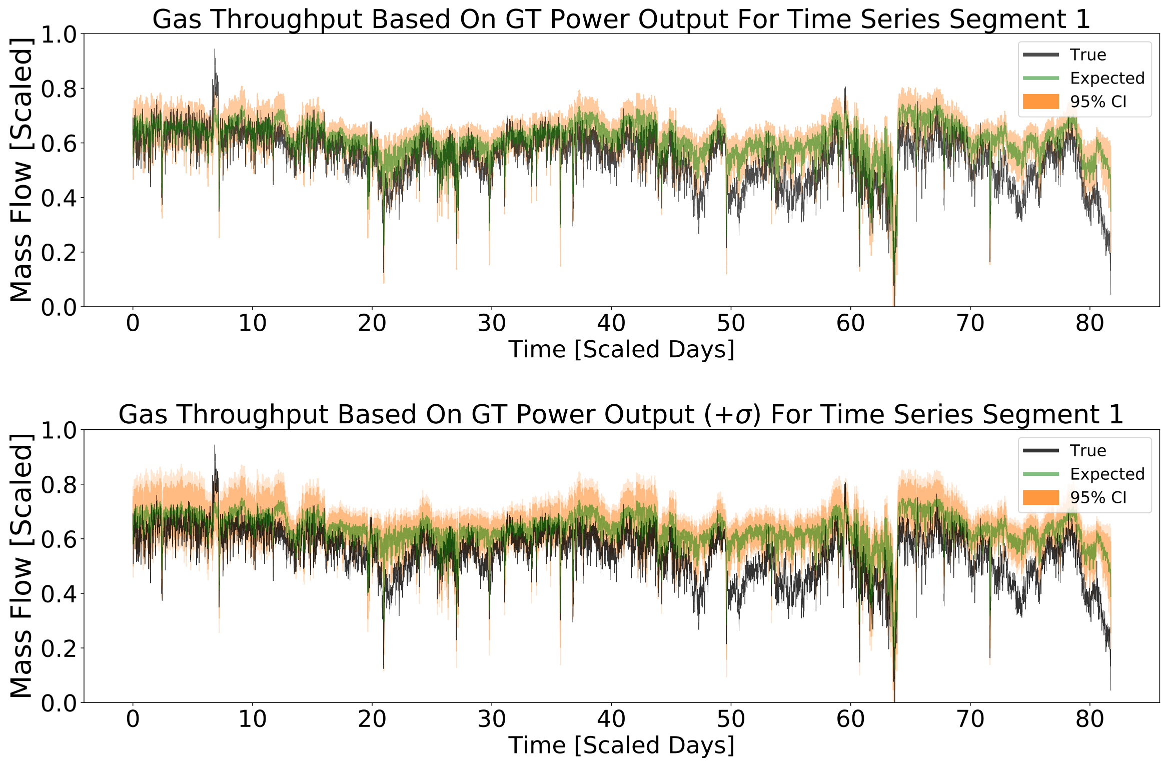



Figure 32: Results for the expected throughput of gas in the compressor train for time series segment 1. The above plot shows the best-case results whereas the lowermost plot shows the worst-case results. The actual and expected gas rate are shown in black and green, respectively.

Figure 33 presents the optimal washing interval, shown in orange. The time of washing for the best and worst-case, is found by the intersection between the cumulative loss in decreased gas throughput and increased fuel consumption, and the loss of gas due to maintenance downtime.

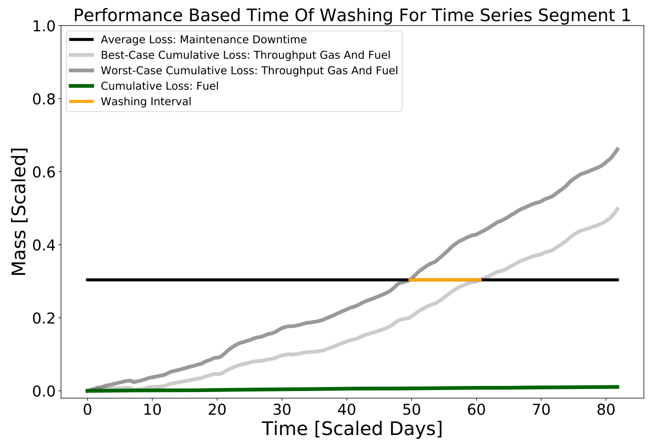



Figure 33: Results for the optimal washing interval for time series segment 1. The black horizontal line shows the average loss for production downtime. The grey and dark grey lines shows the cumulative loss of fuel and gas throughput for the worst and best-case, respectively. The orange line indicates the optimal washing interval.

Table 6: Optimal washing interval for time series segment 1.

| Segment | Washing Interval [$\text{Day}_{\text{WC}}$, $\text{Day}_{\text{BC}}$] |
|---|---|
| 1 | [50, 60] |

Based on the intersections from Figure 33, the recommended time for maintenance is between day 50 and 60. This is earlier than the actual time of washing. However, many maintenance costs are excluded and the optimal washing interval will thus be underestimated.

Ideally, the washing interval should be narrowed down. Since it is the expected power output uncertainty that ultimately determines the washing interval, the uncertainty should, therefore, be decreased to narrow down the interval. Decreasing the uncertainty can be achieved by increasing the training set and possibly introduce feature engineering. It is also possible to resort to other methods to determine the optimal time of washing. For instance, the level of degradation can directly be used to determine the optimal time of washing. This could especially be beneficial for time series segment 1 since the level of degradation was associated with a low uncertainty at the end of the operating interval.

## 6.2 Segment 2

Figure 34 shows the results for the expected throughput of gas for the best and worst-case degradation rates. For visualization purposes, the results are smoothed. This intermediate result will, together with the average bias for fuel consumption, be used to find the optimal time of washing.

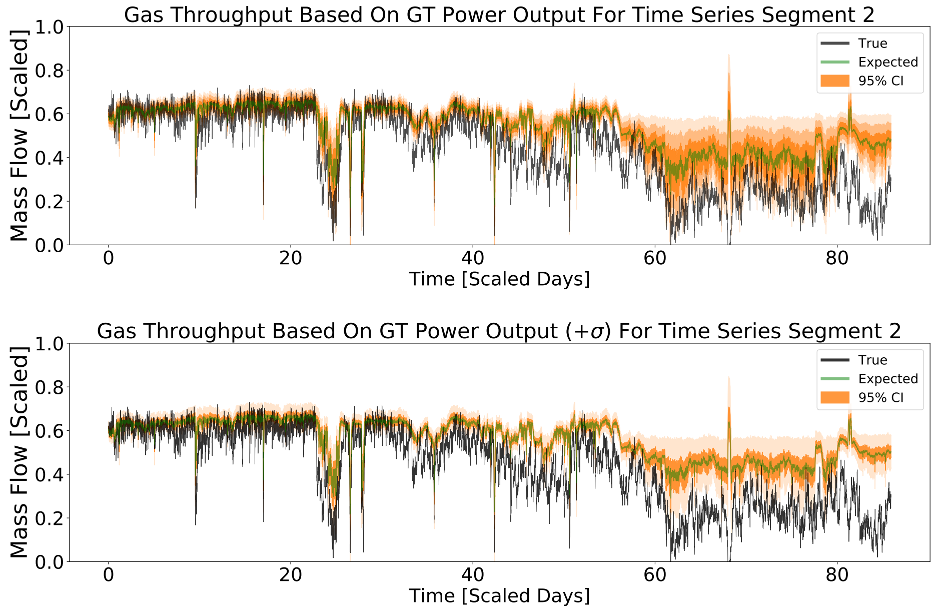



Figure 34: Results for the expected throughput of gas in the compressor train for time series segment 2. The above plot shows the best-case results whereas the lowermost plot shows the worst-case results. The actual and expected gas rate is shown in black and green, respectively.

Figure 35 presents the optimal washing interval, shown in orange. The time of washing for the best and worst-case, is found by the intersection between the cumulative loss in decreased gas throughput and increased fuel consumption, and the loss of gas due to downtime.

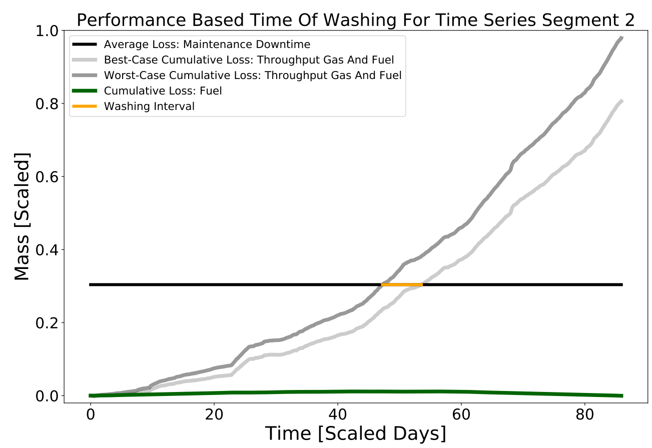



Figure 35: Results for the optimal washing interval for time series segment 2. The black horizontal line shows the average loss for production downtime. The grey and dark grey lines show the cumulative loss of fuel and gas throughput for the worst and best-case, respectively. The orange line indicates the optimal washing interval.

Table 7: Optimal washing interval for time series segment 2.

| Segment | Washing Interval [$\text{Day}_{\text{WC}}$, $\text{Day}_{\text{BC}}$] |
|---|---|
| 2 | [47, 53] |

For time series segment 2, the worst and best-case degradation rates suggests maintenance between day 47 and 53. Compared to time series segment 1, the washing interval is narrowed down to 6 days. This finding was unexpected since time series segment 2 was associated, especially at the end of the operating interval, with large uncertainty. However, several factors may have contributed to this result. Based on the expected power output for time series segment 2, the uncertainty was generally low at the start of the operating interval. This can also be seen in Figure 34, where the throughput of gas for both the best and worst-case degradation yields almost identical rates between day 0 to 50. Since the intersection occurred around the period where the predictions were linked to low uncertainty, the washing interval will, consequently, also be smaller.

## 6.3 Segment 3

Figure 36 shows the results for the expected throughput of gas for the best and worst-case loss. For visualization purposes, the results are smoothed. This intermediate result will, together with the average bias for fuel consumption, be used to find the optimal time of washing.

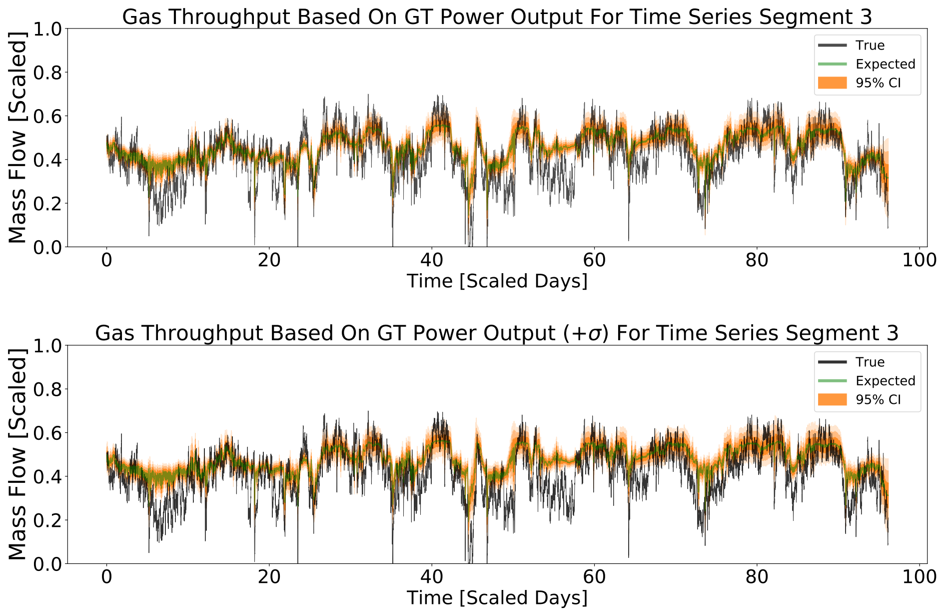



Figure 36: Results for the expected throughput of gas in the compressor train for time series segment 2. The above plot shows the best-case results whereas the lowermost plot shows the worst-case results. The actual and expected gas rate is shown in black and green, respectively.

Figure 37 presents the optimal washing interval, shown in orange. The time of washing for the best and worst-case, is found by the intersection between the cumulative loss in decreased gas throughput and increased fuel consumption, and the loss of gas due to downtime.

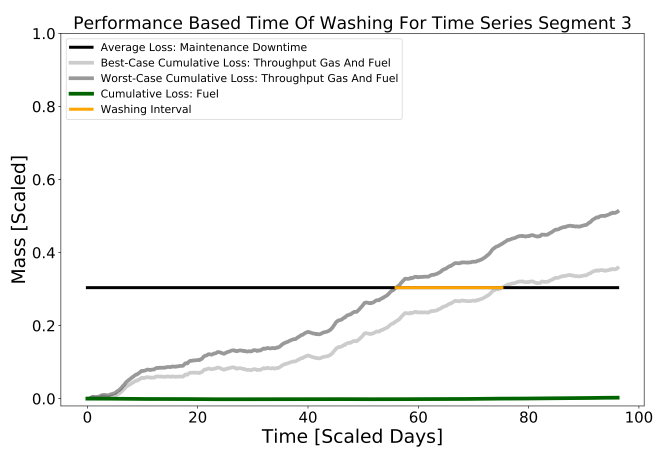



Figure 37: Results for the optimal washing interval for time series segment 3. The black horizontal line shows the average loss for production downtime. The grey and dark grey lines are the cumulative loss of fuel and gas throughput for the worst and best-case, respectively. The orange line indicates the optimal washing interval.

Table 8: Optimal washing interval for time series segment 3.

| Time Series Segment | Washing Interval [Day$_{WC}$, Day$_{BC}$] |
|---|---|
| 3 | [56, 75] |

For time series segment 3, the worst and best-case degradation rates suggest maintenance between day 56 and 75. A later time of washing for time series segment 3 was expected since the operating interval was generally characterized by low degradation rates. As discussed in Section 5.2.3, the expected power output was inconsistent and was generally characterized by intersecting the true power output. From a physical interpretation, this indicates that the performance improves. When evaluating the cumulative loss the ”performance improvement” will, therefore, result in an increased throughput of gas. Thus, the cumulative sum will fluctuate and thus struggle to reach the maintenance loss due to production downtime. Also, the expected power output was also associated with large uncertainties for certain sections. The washing interval for time series segment 3 is, therefore, significantly larger compared to time series segments 1 and 2.

## 6.4 Results Summary

Table 9 summarizes the optimal washing interval for time series segments 1, 2, and 3. The results are given as days after previously performed maintenance. The day the actual maintenance was performed is also presented as a reference to the recommended washing interval.

Table 9: Summary of optimal washing intervals and the actual time of washing for time series segments 1, 2, and, 3.

| Segment | Washing Interval [Day$_{WC}$, Day$_{BC}$] | Actual Time Of Washing [Day] |
|---|---|---|
| 1 | [50, 60] | 82 |
| 2 | [47, 53] | 85 |
| 3 | [56, 75] | 96 |

## 6.5 Discussion

In general, it is possible to find the loss in throughput of gas using the proposed method. With the inclusion of loss in fuel consumption due to degradation, the optimal time of maintenance can be found. Overall, the additional fuel consumption had a negligible effect on the washing interval. This implies that the fuel consumption can be increased without a major effect on the optimal time of maintenance. However, this approach is currently more conceptually than realistic. More cost-related aspects should also be included when determining the optimal time of washing, such as additional costs associated with maintenance. Moreover, the loss due to downtime should also be evaluated more correctly since it greatly impacts the optimal time of washing. For this thesis, the loss due to downtime was found from the average production rate for the whole operating interval. The effectiveness of a wash should also be addressed. Whether the performance recovery is satisfactory after a wash will impact the time of maintenance. For instance, if the recovery is generally poor, it could be more cost-effective to operate the gas turbine in a deteriorated state for longer, since the washing will not improve the performance significantly.

Overall, the proposed methodology for finding the reduced throughput of gas requires good results for the expected power output. If these results are poor it will also be reflected as diminished results for the washing interval. This was especially evident for time series segment 3 where the inconsistent bias in expected power output resulted in an *increased* expected throughput of gas. However, this two-step approach can also have advantages. As discussed in the previous case study, the power output proved to be a good candidate for degradation monitoring. Therefore, it is possible to evaluate the correctness of the washing interval by also examining the expected power output. In other words, if the expected power output follows the trends, has a physical meaningful bias, and is linked to a small uncertainty, it would also imply that the results for the throughput of gas, and thus the washing interval, are also reliable.

# 7 Conclusion And Future Work

The project aimed to measure the degradation of a gas turbine, assessing the uncertainty and find the optimal time of washing using machine learning. It was proposed a general methodology for obtaining the level of degradation using health parameters. By training the machine learning model on data that is assumed to be sampled under new and clean operating conditions, the expected values for the rest of the operating interval could be obtained. The bias of the predictions, also referred to as the expected values, for several health parameters was used to measure the level of degradation. Power output, compressor discharge pressure, and fuel consumption was investigated as health parameters. All the health parameters showed clear signs of performance deterioration. The dataset was divided into three operating intervals, referred to as time series segments, and it was found that the degradation rate was unique for each interval.

All the studied health parameters are qualified candidates for monitoring the degradation. The power output and compressor discharge pressure showed related trends regarding uncertainty, which was expected since the input features were similar. However, since the power output is most affected by degradation the bias was more evident. Besides, monitoring the power output is crucial for determining the economic loss associated with performance deterioration, and is, therefore, an essential health parameter to monitor. Monitoring the compressor discharge pressure could be beneficial since it is measured directly and is consequently less prone to measurement error. Whereas the power output requires two measurements to be calculated. Evaluating these health parameters together is, therefore, crucial to capture the complete picture of the degradation. Monitoring the fuel consumption also led to noteworthy results. Since the compressor train variables were excluded, the overall uncertainty was also decreased. This implies that the uncertainty is associated with drifting compressor train operating conditions, and would consequently cause the test set to extend outside the training set. It could also support the hypothesis that using manipulated variables are target variables and controlled variables as input features will decrease the uncertainty. However, this statement should be investigated more thoroughly for different manipulated variables to determine whether it is generalizable. Similar to the power deterioration, the fuel rate consumption should be monitored for evaluating the increased costs. It can also be beneficial to monitor the fuel consumption to determine the increased emissions caused by gas turbine degradation.

The case study considered three operating intervals and concluded that the degradation rate was unique for all segments. This also supports the motivation for using a machine learning methodology since it can capture deviation that can be difficult to incorporate in an analytical model or by direct inspection of the measurements. The uncertainty was also different for all operating intervals and contributed to assessing the correctness of the results. Time series segments 1 and 2 did, especially, yield good results. The predictions followed the dynamics of the system and the bias followed a physically meaningful trend.

Time series segment 3, on the other hand, struggled more to follow the dynamics and the bias was fluctuating. The results can be improved by increasing the training set or possibly select a different machine learning model. However, the bias did overall follow an increasing trend and suggests that degradation is present.

The results from the case study were used to determine the optimal washing interval. A machine learning model was trained to predict the throughput of gas in the compressor train. Then, the expected power output was used as an input feature to determine the shortfall in gas production. However, the major assumption behind this approach is that the throughput of gas is the bottleneck. It was considered two cases, the best-case where the expected power output was used an input, and worst-case which used the expected power output plus one standard deviation as input. Together with the increased fuel consumption, the optimal washing interval was found as the intersection between the cost associated with performance deterioration and the loss in production due to maintenance downtime. Overall, it was found that the optimal washing interval was earlier than the actual time of washing. Since most costs associated with maintenance was excluded, the case study was more conceptually rather than estimating the true washing interval. However, the method proved to be a good strategy for determining the optimal washing interval but should be advanced to achieve more accurate estimations.

An important consideration when evaluating predictions should be whether the machine learning model is certain. It is commonly accepted that predictions are unreliable if the model is tested outside the distribution it was trained upon and machine learning models are therefore often considered as "a prisoner of its training data set" (Minns and Hall, 1996). This fact is often neglected when deploying machine learning models into the real world (Varoonchotikul, 2003). There are different sources of uncertainty associated with the predictions from a machine learning model.

For future work, more health parameters should be investigated to ensure a more comprehensive consistency between the degradation rates. Typical parameters that can be examined are the inlet mass flow of air, exhaust gas temperature, and isentropic efficiency. The uncertainty could be significantly improved by the inclusion of more training data. More training data would also allow for training the model on a smaller segment for each operating interval, thus yielding more accurate results. The proposed method only assesses epistemic uncertainty and the measurement uncertainties should, therefore, also be incorporated into the confidence intervals. The proposed methodology could also be utilized for evaluating the washing efficiency. By evaluating the performance right after maintenance is performed, the relative recovery compared to the previous new and clean interval can be determined. Overall, there is still room for improvement. However, this thesis has shown that the level of degradation and optimal washing interval can be determined using the proposed methodology. Incorporating uncertainty estimates have also provided valuable information for determining the correctness of the results. All things considered, data-driven condition monitoring for gas turbines can contribute to a more robust approach to measure degradation rates and for making cost-optimal maintenance decisions.

# References

Adamowicz, M. and Zywica, G. (2018). Advanced gas turbines health monitoring systems. *Diagnostyka*, 19(2):77–87.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175.

Batayev, N. (2019). Axial compressor fouling detection for gas turbine driven gas compression unit. *Indonesian Journal of Electrical Engineering and Computer Science*, 15:1257.

Bhandari, A. (2020). Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization.

Bikmukhametov, T. and Jäschke, J. (2020). Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. *Computers & Chemical Engineering*, 138:106834.

Boyce, M. P. (2012). 1 - An Overview of Gas Turbines. In Boyce, M. P., editor, *Gas Turbine Engineering Handbook (Fourth Edition)*, pages 3–88. Butterworth-Heinemann, Oxford.

Boyce, M. P. and Gonzalez, F. (2005). A Study of On-Line and Off-Line Turbine Washing to Optimize the Operation of a Gas Turbine. *Journal of Engineering for Gas Turbines and Power*, 129(1):114–122.

Breeze, P. (2016). Chapter 1 - An Introduction to Gas-Fired Power Generation. In Breeze, P., editor, *Gas-Turbine Power Generation*, pages 1–7. Academic Press.

Brownlee, J. (2019). A Gentle Introduction to Uncertainty in Machine Learning.

Diakunchak, I. S. (1992). Performance Deterioration in Industrial Gas Turbines. *Journal of Engineering for Gas Turbines and Power*, 114(2):161–168.

Drakos, G. (2018). Handling Missing Values in Machine Learning.

El Hadik, A. A. (1990). The Impact of Atmospheric Conditions on Gas Turbine Performance. *Journal of Engineering for Gas Turbines and Power*, 112(4):590–596.

Enyia, J., Yiguang, L., Thank-God, I., and Igbong, D. (2015). Industrial Gas Turbine On-Line Compressor Washing For Power Generation. *international journal of energy research and technology*.

Fentaye, A., Baheta, A., Gilani, S., and Kyprianidis, K. (2019a). A Review on Gas Turbine Gas-Path Diagnostics: State-of-the-Art Methods, Challenges and Opportunities. *Aerospace*, 6:83.

Fentaye, A. D., T. Baheta, A., and Gilani, S. (2019b). A Review on Gas Turbine Gas-Path Diagnostics: State-of-the-Art Methods, Challenges and Opportunities.

Gal, Y. (2015). A Theoretically Grounded Application of Dropout in Recurrent Neural Networks.

Gal, Y. (2016). *Uncertainty in Deep Learning*. PhD thesis, University Of Cambridge.

Giampaolo, T. (2013). *The Gas Turbine Handbook: Principles and Practices*. Fairmont Press.

Giles, C. L., Lawrence, S., and Tsoi, A. C. (2001). Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference. *Machine Learning*, 44(1):161–183.

Gonfalonieri, A. (2019). How to Implement Machine Learning For Predictive Maintenance.

Graves, A. (2011). Practical Variational Inference for Neural Networks.

Gupta, A. and Lam, M. S. (1996). Estimating Missing Values Using Neural Networks. *The Journal of the Operational Research Society*, 47(2):229–238.

Hanachi, H., Liu, J., Ding, P., Yong Kim, I., and Mechefske, C. K. (2018). Predictive Compressor Wash Optimization for Economic Operation of Gas Turbine. *Journal of Engineering for Gas Turbines and Power*, 140(121006).

Haq, I. and Saravanamuttoo, H. I. H. (1991). Detection of Axial Compressor Fouling in High Ambient Temperature Conditions. In *GT1991*, Volume 3: Coal, Biomass and Alternative Fuels; Combustion and Fuels; Oil and Gas Applications; Cycle Innovations. V003T07A006.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*.

Jerez, J., Molina, I., García Laencina, P., Alba, E., Ribelles, N., Martin, M., and Franco, L. (2010). Missing Data Imputation Using Statistical and Machine Learning Methods in a Real Breast Cancer Problem." Artificial Intelligence in Medicine, 50, 105-115. *Artificial intelligence in medicine*, 50:105–15.

Jiang, X., Pang, Y., Li, X., Pan, J., and Xie, Y. (2018). Deep neural networks with Elastic Rectified Linear Units for object recognition. *Neurocomputing*, 275:1132–1139.

Kiakojoori, S. and Khorasani, K. (2016). Dynamic neural networks for gas turbine engine degradation prediction, health monitoring and prognosis. *Neural Computing and Applications*, 27(8):2157–2192.

Kingma, D. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Krampf, F. M. (1992). A Practical Guide for Gas Turbine Performance Field and Test Data Analysis. In *GT1992*, Volume 3: Coal, Biomass and Alternative Fuels; Combustion and Fuels; Oil and Gas Applications; Cycle Innovations. V003T07A016.

Kurz, R. (2005). *Gas Turbine Performance*.

Kurz, R. and Brun, K. (2011). *Fouling Mechanism in Axial Compressors*, volume 134.

Kurz, R. and Brun, K. (2012). Fouling Mechanisms in Axial Compressors. *Journal of Engineering for Gas Turbines and Power*, 134(032401).

Lars Hulstaert (2017). Understanding objective functions in neural networks. *towardsdatascience*.

Lawrence, S. and Giles, C. (2000). *Overfitting and neural networks: Conjugate gradient and backpropagation*, volume 1.

Lévesque, J.-C., Gagné, C., and Sabourin, R. (2016). *Bayesian Hyperparameter Optimization for Ensemble Learning*.

M. Raghavan, S., Palatel, A., and Simon, J. (2019). Artificial Intelligence Based Gas Turbine Compressor Wash: A Predictive Approach. In *GTINDIA2019*, Volume 2: Combustion, Fuels, and Emissions; Renewable Energy: Solar and Wind; Inlets and Exhausts; Emerging Technologies: Hybrid Electric Propulsion and Alternate Power Generation; GT Operation and Maintenance; Materials and Manufacturing (Including Coatings, Composites, CMCs, Additive Manufacturing); Analytics and Digital Solutions for Gas Turbines/Rotating Machinery. V002T09A006.

Mak, K. K., Lee, K., and Park, C. (2019). Applications of machine learning in addiction studies: A systematic review. *Psychiatry Research*, 275:53–60.

Meher-Homji, C., Bromley, A. F., and Stalder, J.-P. (2019). Appendix - Gas Turbine Performance Deterioration and Compressor Washing. In Forsthoffer, M., editor, *Forsthoffer's Component Condition Monitoring*, pages 101–185. Butterworth-Heinemann.

Meyer Zu Wickern, V. (2019). *Challenges and Reliability of Predictive Maintenance*.

Minns, A. W. and Hall, M. J. (1996). Artificial neural networks as rainfall-runoff models. *Hydrological Sciences Journal*, 41(3):399–417.

Molin, S. (2019). *Hands-On Data Analysis with Pandas: Efficiently perform data collection, wrangling, analysis, and visualization using Python*. Packt Publishing Ltd.

Natrella, M., Croarkin, C., and Tobias, P. (2012). *Handbook of Statistical Methods*.

Neal, R. M. (1996). Introduction. In Neal, R. M., editor, *Bayesian Learning for Neural Networks*, pages 1–28. Springer New York, New York, NY.

Noordermeer, J. and Eng, P. (2019). Understanding Gas Turbine Performance.

Ogbonnaya, E. A. (2011). Gas Turbine Performance Optimization Using Compressor Online Water Washing Technique. *Engineering*, 335058:500–507.

Phi, M. (2018). Illustrated Guide to LSTM's and GRU's: A step by step explanation.

Razak, A. (2007). 8 - Behaviour of gas turbines during off-design operation. In Razak, A., editor, *Industrial Gas Turbines*, pages 202–240. Woodhead Publishing.

Saravanamutto, H. and Lakshminarasimha, A. (1985). A Preliminary Assessment of Compressor Fouling. *ASME*.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Sherstinsky, A. (2018). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.

Smith, R. (2014). *Chemical Process: Design and Integration*. Wiley.

Snoek, J., Larochelle, H., and Adams, R. (2012). PRACTICAL BAYESIAN OPTIMIZATION OF MACHINE LEARNING ALGORITHMS.

Soares, C. (2008). 4 - Gas Turbine Major Components and Modules. In Soares, C., editor, *Gas Turbines*, pages 119–166. Butterworth-Heinemann, Burlington.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*.

Tarabrin, A. P., Schurovsky, V. A., Bodrov, A. I., and Stalder, J.-P. (1998). An Analysis of Axial Compressor Fouling and a Blade Cleaning Method. *Journal of Turbomachinery*, 120(2):256–261.

Varoonchotikul, P. (2003). *Flood Forecasting using Artificial Neural Networks*. A.A . Balkema Pubhshers.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2017). Chapter 10 - Deep learning. In Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J., editors, *Data Mining (Fourth Edition)*, pages 456–459. Morgan Kaufmann.

Yakowitz, S. and Lugosi, E. (1990). Random Search in the Presence of Noise, with Application to Machine Learning. *Siam Journal on Scientific and Statistical Computing*, 11.

Yarin Gal and Zoubin Ghahramani (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR.

Zaba, T. (1980). Losses in gas turbines due to deposits on the blading.

Zagorowska, M., Schulze Spüntrup, F., Ditlefsen, A.-M., Imsland, L., Lunde, E., and Thornhill, N. F. (2020). Adaptive detection and prediction of performance degradation in off-shore turbomachinery. *Applied Energy*, 268:114934.

Zaremba, W., Sutskever, I., and Vinyals, O. (2014). Recurrent Neural Network Regularization.

Zhang, X., Chen, X., Yao, L., Ge, C., and Dong, M. (2019). *Deep Neural Network Hyperparameter Optimization with Orthogonal Array Tuning.*

Zohuri, B. (2015). Gas Turbine Working Principles. pages 147–171.

Zwebek, A. and Pilidis, P. (2003). Degradation Effects on Combined Cycle Power Plant Performance—Part I: Gas Turbine Cycle Component Degradation Effects. *Journal of Engineering for Gas Turbines and Power*, 125(3):651–657.

# A Machine Learning Background

This section will provide the necessary machine learning background and will give a brief introduction to common concepts. The main concepts which will be covered are the working principle of neural networks and regularization techniques.

## A.1 Neural Networks

Artificial neural networks(ANN) are computational networks that try to reproduce the decision process of how a nerve cell in human body functions. For regression problems, the objective, $J$, is to minimize the error between the predictions, $\hat{y}_i$, and the true values, $y_i$. Several metrics can be used to evaluate the error. One of the most common for regression problems is the mean squared error (Lars Hulstaert, 2017).

$$J = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{24}$$

A feed-forward neural network is built from many interconnected neurons. The neurons are the decision-makers and determine the importance of each feature. This is similar to the LSTM cell that will be described in Section 3. A layer in a neural network consists of several neurons. The input layer is the first collection of nodes that receives the information and defines the start of the workflow. The intermediate stacked layers are referred to as hidden layers. The final layer, or output layer, generates the final predictions. The topology of a neural network is shown in Figure 38.

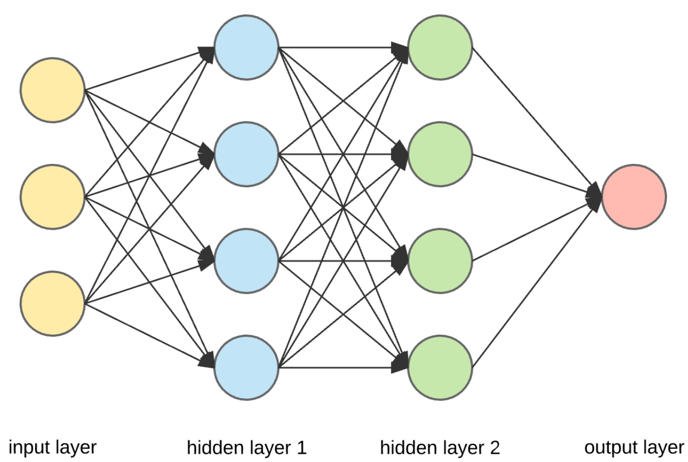



Figure 38: An illustration of a neural network with one input layer, two hidden layers and an output layer.

Each neuron is associated with a weight matrix and a bias. The weight matrix either amplifies or dampens based on the significance of the input w.r.t. the learning objective (Schmidhuber, 2015). The bias is an additional parameter used in the neural network to adjust the output to ensure the best fit to the data. In each neuron, the input or output from a previous neuron is multiplied with the weight function and the bias term is added.

$$Y = \sum_{i=1}^{N}(w_i x_i) + b_i \tag{25}$$

Where $w_i$ denotes the weight matrix, $b_i$ the bias and $x_i$ the input. Whether a neuron is activated is regulated by an activation function. The activation function determines if $Y$, from Equation 25, is larger than a certain threshold. Intuitively, the activation function determines the significance of the input w.r.t. the target variable. There are several types of activation functions. For neural networks, rectified linear unit, or ReLU is commonly used. ReLU is defined as linear for all positive values and zero for all negative values Jiang et al. (2018).

$$f(x) = \begin{cases} 0, & \text{if x} < 0 \\ \text{x}, & \text{if x} \geq 0 \end{cases}$$

The sigmoid activation function is also widely used for neural networks. It outputs a value between 0 to 1 and is, therefore, especially used for predicting a probability or signifying importance. The sigmoid activation function is defined as,

$$f(x) = \frac{1}{1 + e^{-x}} \tag{26}$$

The hyperbolic tangent activation function, or tanh, is also used in neural networks. It outputs a value between -1 and 1. Tanh is often used because the sigmoid activation tends to get stuck during training.

Back-propagation is the essence behind training a neural network. Based on the loss during each pass (epoch) the weights are updated to train the network. An important hyperparameter for back-propagation is the learning rate. The learning rate is the hyper-parameter that determines the adjustment of weights w.r.t. the loss gradient. Stochastic gradient descent is the core technique for updating the weights of the neural network and provides information about the direction the function should take to reach the minimum. Equation 27 shows how stochastic gradient descent updates the weights based on the objective function, $J$, and learning rate, $lr$.

$$w_{i+1} = w_i - lr \frac{\partial}{\partial w_i} J(w_i) \tag{27}$$

$$b_{i+1} = b_i - lr \frac{\partial}{\partial b_i} J(b_i) \tag{28}$$

Selecting the learning rate is crucial during the training phase. If the learning rate is too small, the convergence will, consequently, be slower and can increase the run-time drastically. Besides, the optimization is also more prone to getting stuck in local minimums and can lead to poor generalization. If the learning rate a too large, the optimization can easily overshoot the minimum and thus fail to converge (Zhang et al., 2019).

As mentioned, Stochastic Gradient Decent (SGD) presented in Equation 27 is the essence behind the optimization and weight updates. However, an issue with SGD is that the step size is equal for all parameters thus it tends overshoot near the exact minima. Other techniques have been proposed to ensure smoother optimization. Generally, Adaptive Moment Estimation (Adam) is preferred for training deep neural networks. The method is based on an adaptive learning rate for each parameter and results in fewer oscillations and faster convergence (Kingma and Ba, 2014).

The essential purpose of machine learning is to generalize the underlying statistical relationships on the training data to produce accurate predictions for an unseen dataset. As for all machine learning models, neural networks tend to overfit. Overfitting occurs if the model is fitted too exactly to the training set (Lawrence and Giles, 2000). There are different methods to prevent the neural net from overfitting. All methods that aim to prevent overfitting are referred to as regularization. Regularization incorporates several techniques such a modifying the model architecture, tuning hyperparameters, penalizing large weight updates, and applying dropout.

Dropout is a technique to prevent overfitting (Srivastava et al., 2014). The idea is to randomly drop units and connections, with a probability $(1 - p)$ during training. Dropout can also be seen as ”thinning” out the network such that it prevents the units from co-adapting too tight to the training data. Preventing co-adapting will also ensure that all the connections have equal predictive capabilities. For each pass of the training data, a new ”thinned” network is created, sampled, and trained. Figure 39 shows how dropout is applied in the neural network.

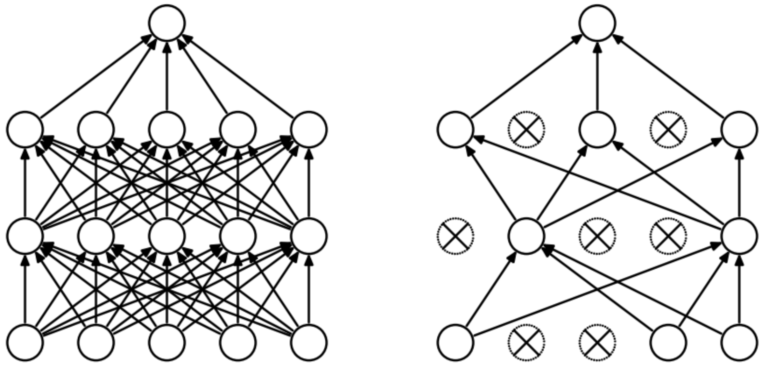


Figure 39: Illustration of dropout. The picture to the right shows a neural network with 2 hidden layers. The picture to the lefts shows the neural networks architecture after dropout is applied (Srivastava et al., 2014).

# B Preprocessing

One of the most important steps in machine learning is preprocessing the data. Preprocessing will including replacing missing data, removing noise, and feature scaling. For the received dataset two issues required the data to be carefully preprocessed.

1. The compressor discharge temperature was missing a large (up to 30%) portion of measurement data.
2. The measurements were overall noisy.

## B.1 Missing Values

Missing values is a frequent challenge for real data sets. Time series data are especially postponed to having missing values since the data often originates from sensors that are prone to breakdowns. The simplest method is substituting the missing values with the mean or median of the dataset. However, this can lead the skewness and a biased dataset (Drakos, 2018). There have been proposed several advanced methods and techniques to deal with missing values. To impute the missing values for the compressor discharge temperature a feed-forward neural network was used. To ensure good generalizability, the model was trained and validated with bayesian hyperparameter optimization and early stopping. A feed-forward neural network was chosen because no statistical assumption could be made, such a seasonality or general trends. This fact is supported by Gupta and Lam (1996) which found that imputing missing values using neural networks can outperform other techniques if any statistical assumption cannot be made. Jerez et al. (2010) also argued that machine learning techniques were most suited for replacing missing values compared to other statistical procedures.

## B.2 Exponential Smoothing

Exponential smoothing is used for removing noise in the dataset. The technique is based on an exponential windows function, where past observations are weighted stronger compared to previous observations (Natrella et al., 2012). The smoothed value of a raw measurement can be expressed as,

$$S_t = \alpha(y_{t-1}) + (1-\alpha)S_{t-1} \tag{29}$$

Where $S_t$ denotes the smoothed value, $\alpha$ the smoothing coefficient, $y_{t-1}$ the raw data point, and $S_{t-1}$ the previous smoothed value. By expanding Equation 29, it can be shown that the expanding equation can be written on an exponential form,

$$S_t = \sum_{i=1}^{t-2} (1-\alpha)^{i-1} y_{t-1} + (1-\alpha)^{t-2} S_2 \tag{30}$$

To perform exponential smoothing, the tuning parameter $\alpha$ must be determined. For this thesis, $\alpha$ was selected by inspecting the smoothed results to determine the trade-off between maintaining the general trends and removing obvious outliers. Based on this, $\alpha$ was then selected to be 0.7 which preserved trends but eliminated apparent noise.

## B.3 Data Scaling

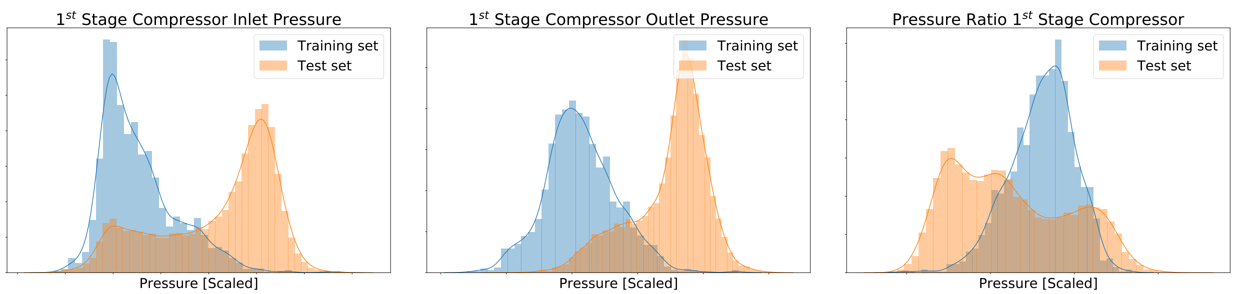
Machine learning models that utilize optimization techniques such a gradient descent requires the data to be scaled (Bhandari, 2020). Features of similar scale contribute with a quicker and smoother converges towards to minimum. Scaling also ensures that each feature receives the same emphasis. Standardization is a common scaling technique where the values are centered around the mean with the unit standard deviation and are used in this thesis. Equation 31 shows how a raw data point is standardized.

$$X' = \frac{X - \mu}{\sigma} \tag{31}$$

Where $X'$ denotes the scaled value, $X$ the raw data point, $\mu$ the mean and $\sigma$ the standard deviation.

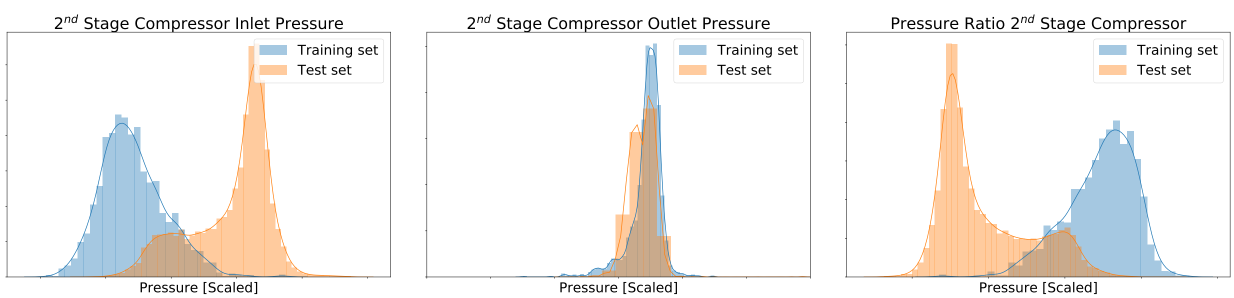
# C Feature Engineering For Expected Power Output

Rather than feeding the raw data right into the machine learning model, the data can be transformed into features. Feature engineering can optimize the information density and represent the underlying pattern better. Also, introducing features could allow the machine learning model to discover and exploit new relationships. For instance, the power requirement in a compressor is proportional to the pressure ratio. Moreover, if the training and test set distributions for two features are evenly shifted, it can also be introduced new features that are more mutually overlapping. This section investigates the effect on uncertainty by transforming the inlet and outlet pressure of the 1 $^{\text{st}}$ and 2 $^{\text{nd}}$ into the pressure ratios. The expected power output for time series 2 will be considered. Figure 40 and 41 shows the original inlet and outlet pressure distributions and the resulting distribution for the pressure ratio for the 1 $^{\text{st}}$ and 2 $^{\text{nd}}$ stage compressor.




(a) Inlet pressure distribution (b) Outlet pressure distribution (c) Pressure ratio distribution

Figure 40: Original and transformed distributions for the 1 $^{\text{st}}$ stage compressor.




(a) Inlet pressure distribution (b) Outlet pressure distribution (c) Pressure ratio distribution

Figure 41: Original and transformed distributions for the 2 $^{\text{nd}}$ stage compressor.

Figure 42 shows the standard deviations for the case with raw features for the case with feature engineering. Overall, the uncertainty decreases. This could be due to more overlapping training and test distributions. However, the effect is small since there are still several sources of uncertainty in the training and test sets. Also, feature engineering did not significantly improve the resulting pressure ratio distribution for the 2$^{nd}$ stage compressor.

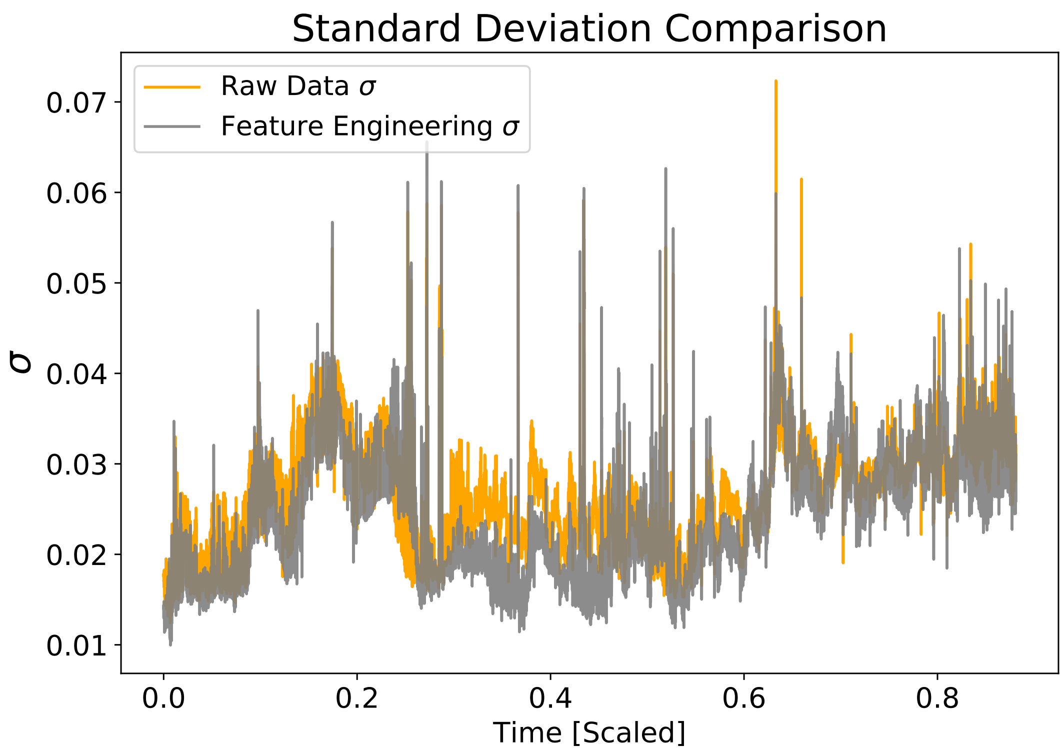



Figure 42: Standard deviations for the expected power output with raw data and feature engineering.

# D Bayesian LSTM Implementation

The following code shows the implementation of the Bayesian LSTM used in this thesis. The implementation is based on the PyTorch library. The code provides a flexible approach to test networks with different architectures and hyperparameters.

```
import torch
import torch.nn as nn
import pandas as pd
import numpy as np
import math

class BayesianLSTM(nn.Module):

    """
    Creates object of a Bayesian LSTM network with dropout probability, number of layers,
    hidden dimension and output dimension.

    Methods
    --------
    forward(x):
        Computes forward propagation of the network

    """

    def __init__(self, input_dim, hidden_dim, layer_dim, output_dim, dropout):

        super(BayesianLSTM, self).__init__()

        """
        Parameters
        ----------
        input_dim : int
            Size of input feature space
        hidden_dim : int
            Hidden dimension
        layer_dim : int
            Number of stacked LSTM layers
        output_dim : int
            Size of output dimension
        dropout : float
            Dropout probability
        """
```

```
        # Hidden dimensions
        self.hidden_dim = hidden_dim
        #input dimension
        self.input_dim = input_dim
        # Number of hidden layers
        self.layer_dim = layer_dim
        #dropout
        self.dropout = dropout

        # Sequence of LSTM cells
        self.rnns = []
        for l in range(self.layer_dim):
            self.rnns.append(
                            nn.LSTM(input_dim if l == 0 else self.hidden_dim,
                            self.hidden_dim,
                            num_layers=1)
                            )

        self.rnns = torch.nn.ModuleList(self.rnns)

        # Readout layer
        self.fc = nn.Linear(hidden_dim, output_dim)


    def forward(self, x):
        """
        Computes forward propagation of the network
        """

        #Define input dropout mask
        input_mask = torch.ones(1, 1, self.input_dim).bernoulli_(1-self.dropout)

        #Define hidden dropout mask for all intermediate LSTM layers
        hidden_mask = []
        for l in range(self.layer_dim):
            hidden_mask.append(torch.ones(1, 1, self.hidden_dim).bernoulli_(1-self.dropout))

        #Define output dropout mask
        output_mask = torch.ones(1, 1, self.hidden_dim).bernoulli_(1-self.dropout)

        # Initialize hidden state with zeros
        h0 = torch.zeros(1, x.size(1), self.hidden_dim).requires_grad_()
        # Initialize cell state
```

```
        c0 = torch.zeros(1, x.size(1), self.hidden_dim).requires_grad_()

        #Apply same dropout mask to input layer
        x = x*input_mask

        #For each lstm cell apply the same dropout mask for recurrent connections.
        for l, lstm in enumerate(self.rnns):
            if l == 0:
                out, (hn, cn) = lstm(x, (h0, c0))
                hn = hn*hidden_mask[l]
            else:
                out, (hn, cn) = lstm(out, (hn, cn))
                hn = hn*hidden_mask[l]

        #Apply same output dropout mask
        out = out*output_mask

        #Readout layer
        out = self.fc(out)

        return out

    class BayesianLSTM_Wrapper():

    """
    Creates object of a Bayesian LSTM wrapper to fit and evaluate the loss on validation set.

    Methods
    -------
    fit(x_train, y_train, epochs):
        Computes the backpropagation and updates the weights

    evaluate(x_val, y_val)
        Computes the loss for validation set. Used for fitting the network early stopping.
    """

    def __init__(self, model, lr):

        """
        Parameters
        ----------
        model : object
            Network from BayesianLSTM class
        lr : float
```

```
            Learning rate
        optimizer : object
            Optimizer for backpropagation.
        criterion : object
            Objective function, MSE loss
        """

        self.model = model
        self.lr = lr
        self.optimizer = torch.optim.Adam(model.parameters(), lr=lr)
        self.criterion = nn.MSELoss()

    def fit(self, x_train, y_train, epochs):
        """
        Trains the Bayesian LSTM network with backpropagation.

        Parameters
        ----------
        x_train : torch tensor
            Training input data
        y_train : torch tensor
            Training target data

        """

        for epoch in range(epochs):

            #Initialize optimizer
            self.optimizer.zero_grad()

            #Calculates forward pass from the BayesianLSTM object
            output = self.model(x_train.float())

            #Calculate loss
            loss = self.criterion(output.view(-1), y_train.view(-1).float())

            #Calculate backpropagation
            loss.backward()

            #Update weights
            self.optimizer.step()

    def evaluate(self, x_val, y_val):
```

```
"""
Evaluate the loss on validation set

Parameters
----------
x_val : torch tensor
    Validation input data
y_val : torch tensor
    Validation target data

"""
#List of samples
samples = []

#Sample predictions from the model
for i in range(100):
    preds = self.model(x_val.float()).view(-1)
    samples.append(preds.detach().numpy().flatten())

samples = np.array(samples)

#Calculate mean
mean = (samples.mean(axis = 0)).reshape(-1)

#Calculate mean squared error
mse = ((np.array(y_test).reshape(-1)-np.array(mean))**2).mean()

return mse
```

Victoria Glott

Condition Monitoring Of A Gas Turbine Using Bayesian Recurrent Neural Networks.

NTNU

Norwegian University of
Science and Technology