Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Informatics

Predicting Machine Outages using Deep Learning

Vorhersage von Maschinenausfällen durch Deep Learning

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Alexander Tatowsky, BSc
Matrikelnummer 0625351

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.univ.Prof. Dr. Andreas Rauber
Mitwirkung: Dr. Alexander Schindler

Wien, 9. September 2021

Alexander Tatowsky

Andreas Rauber

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Wien Bibliothek Your knowledge hub

Informatics

Predicting Machine Outages using Deep Learning

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Medical Informatics

by

Alexander Tatowsky, BSc
Registration Number 0625351

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.univ.Prof. Dr. Andreas Rauber
Assistance: Dr. Alexander Schindler

Vienna, 9th September, 2021

Alexander Tatowsky

Andreas Rauber

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Erklärung zur Verfassung der Arbeit

Alexander Tatowsky, BSc
Wien, 1120, Oswaldgasse 27/ 5/ 40

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 9. September 2021

Alexander Tatowsky

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Danksagung

Ich möchte Bernhard Haslhofer, Alexander Schindler und Ana Jalali für ihre fachliche Hilfe danken und meiner Freundin Andrea Fröwis für ihre moralische Unterstützung und Geduld während ich meine Arbeit verfasst habe. Besonderer Dank sollte an Alexander Schindler gerichtet werden für seine Tätigkeit in der Deep Learning Community, wo er Leute zusammen bringt und mir im Endeffeckt die Möglichkeit geschaffen hat diese Arbeit zu schreiben.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acknowledgements

I want to thank Bernhard Haslhofer, Alexander Schindler and Ana Jalali for their professional support and my girlfriend Andrea Fröwis for moral support and patience during the process of writing my thesis. Special thanks should be directed to Alexander Schindler for his work for the deep learning community by bringing people together, which in the end created my opportunity to write this thesis.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek, Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Im Bereich der Industrie 4.0 werden in modernen Fertigungssystemen und Industriezweigen große Anstrengungen unternommen, um effektive Systeme zur Überwachung der Maschinengesundheit und Prognose-Systeme zu entwickeln [KY18]. Neben verschiedenen Methoden, wie modellbasierten oder wissensbasierten Ansätzen, gewinnen datengesteuerte Methoden, insbesondere neuronale Netzwerke an Bedeutung aufgrund ihrer Eigenschaft, bei großen Datenmengen eine gute Leistung zu erbringen und ihrer Fähigkeit, den zukünftigen Status basierend auf aktuellen Informationen vorherzusagen. Nach dem Training anhand von Zustandsdaten und Feedback können sie in integrierte Steuerungen eingebettet werden und ermöglichen eine Echtzeit-Bewertung. Neben hochmodernen Produktionsanlagen, die mit einer großen Anzahl von Sensoren und Messungen ausgestattet sind, gibt es Hersteller, die Messungen auf ungeordnete Weise sammeln. Während beide danach streben, ihre Effizienz zu maximieren, unterscheiden sich die Voraussetzungen zur Vorhersage erheblich. Die Verwendung von neuronalen Netzwerken nach dem Stand der Technik, die für Zeitreihendaten verwendet werden, erfordert kontinuierliche Messungen mehrerer Sensoren und die Kenntnis des Zustands, um ein Modell richtig zu trainieren. In dieser Arbeit wird untersucht, inwieweit eine moderne neuronale Netzwerkarchitektur mit einem Datensatz mit fehlenden Zielvariablen und unzureichender Kenntnis des wahren Zustandes der Maschine funktioniert. Dazu werden zwei Fallstudien durchgeführt: Eine neuronale Netzwerkarchitektur wird anhand eines genau definierten Referenz- Datensatzes [SGSE08] implementiert, verfeinert und ausgewertet. In der zweiten Fallstudie wird das Netzwerk anhand eines realen Datensatzes ausgewertet, der von einer Produktionslinie stammt, jedoch nicht dezidiert für derartige Prognoseaufgaben gesammelt wurde. Implikationen betreffen die Datenerfassung und Messungen zur Verbesserung der Datenqualität, um ausreichende Daten für modernste maschinelle Lernmethoden bereitzustellen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

In the area of industry 4.0 in modern manufacturing systems and industries, great research effort are made in developing effective machine health monitoring and prognosis systems [KY18]. Among other methods like model based or knowledge based approaches, data driven methods, especially neural networks, gain attention due to their characteristic to perform well with large data sets and their capabilities to predict the future state based on up-to-date information. Once trained by utilizing condition data and on site feedback, they can be embedded on on-board controllers and enable real-time assessment. Besides state of the art production plants which are equipped with a great amount of sensors and measurements, there are also manufacturers that collect measurements, but in a disordered way. While both strive to maximize their efficiency, the preconditions differ significantly. The usage of state of the art neural networks used for time series data requires continuous measurements of multiple sensors and the knowledge of the state of the device to properly train a model. This thesis investigates to what extent a neural network architecture performs on a data set with missing target variables and inappropriate ground truth data. Therefore two case studies are conducted: A neural network architecture is constructed, refined and evaluated on a well defined data set [SGSE08]. In the second case study the network is evaluated on a real life data set which was collected from a production line and was not specifically collected for forecasting tasks. Implications concern the data acquisition and measurements to improve the data quality to provide sufficient data for state of the art machine learning methods.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Contents

Kurzfassung **xi**

Abstract **xiii**

Contents **xv**

1 Introduction **1**

2 State of the Art **5**
2.1 Deep Learning . . . 7
2.2 Data Mining . . . 10
2.3 Missing Data . . . 13

3 Study of the Reference Data Set C-MAPSS [SGSE08] **15**
3.1 Methodology . . . 15
3.2 Description of the Data Set . . . 18
3.3 Base Line . . . 19
3.4 Data preparation for Machine Learning . . . 20
3.5 Hyperparameter Evaluation . . . 22
3.6 Experimental Results . . . 28
3.7 Discussion and Potential for Future Work . . . 31

4 Study of a Dataset not Optimized for Predictive Maintenance **39**
4.1 Methodology - Iterative Application of CRISP-DM . . . 39
4.2 Business Objectives . . . 41
4.3 Data Understanding . . . 41
4.4 Data Preparation . . . 55
4.5 Modeling . . . 61
4.6 Evaluation . . . 62
4.7 Discussion . . . 63

5 Conclusions **69**

List of Figures **71**

List of Tables **73**

Bibliography **75**

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 1

Introduction

Maintenance costs are a major part of the total cost of manufacturing in production plants. Already 2002 the awareness of potential savings emerged. Mobley states in the book "An introduction to Predictive Maintenance (Second Edition)", that the maintenance costs can represent between 15 and 60 percent of the cost of goods produced, depending on the domain [Mob02]. Business approaches to maintenance management can be grouped into three main categories [SSP+15]:

- Run-to-failure
- Preventive maintenance
- Predictive maintenance

The naive but also most cost intensive strategy is the Run-to-failure strategy. Maintenance interventions are only performed after the occurrence of failures. Since this is the simplest approach it is adopted frequently especially in small businesses, but a failure results in the longest downtime and is usually more substantial than those associated with planned actions.

A more cost effective approach is called preventive maintenance. To reduce downtime, maintenance actions are carried out according to a planned schedule based on the expected life period of parts. The expected life period is determined on basis of statistical information. By setting the maintenance interval to a point in time where the probability for a failure is low, breakdowns are prevented with a high probability. But at that point in time, maintenance actions are not necessary yet with an equally high probability. By delaying maintenance actions, the efficiency increases at the cost of higher risk of a breakdown. In practice with this strategy failures are usually prevented, but at the cost of unnecessary maintenance actions resulting in inefficient use of resources and increased operating costs.

The third strategy is designed to overcome the inefficiency of unnecessary maintenance actions while still minimizeing the risk of failures by trying to estimate the health status of parts of the equipment. Therefore, a model is built to predict the remaining useful life or the probability for a device or parts of the device to fail in a certain time period. As stated in chapter 2, different approaches tackle the task of generating a capable model. With the evolution of maintenance strategies, the methodology generating such models evolved too from engaging experts who have knowledge about the devices through education and experience to AI based systems, and further from physics based models to data based approaches[KY18]. While physics based approaches are focused on specific machine types, data driven methods tend to generalize health indicators [LWZ+14]. Especially supervised machine learning algorithms are an eligible instrument to automatically learn such indicators. With the arising of the Industry 4.0 area neural networks, which were first introduced by Donald Hebb 1949 [Sha86], obtained new significance. The combination of having big amounts of data available and having enough and affordable storage and processing power triggered extensive research in this field, as described in chapter 2. Neural network architectures discussed in this chapter are not only capable to identify and learn relevant features for a specific device. Due to their way of learning they have also the potential to be able to adapt to different scenarios and therefore generalizing to different devices. For predictive maintenance an ideal method would be sufficient to predict failures of different types of machines with an equally high accuracy and therefore reduce the effort to develop or adopt approaches for them. To move towards such a generalization, two views can be identified: Either the method is able to cope with a weakly defined task with missing target variables or inappropriate ground truth data, or the quality of the data has to fulfill minimum requirements of the used method. These two views are connected. If the output of two different machines, for example, has similar features and fulfills certain quality requirements, the predictive maintenance method can specialize on the features and the degree of quality. With variation within the output the capability of the method of coping with the variation also has to grow. The motivation of this thesis is to push one step towards the ideal predictive maintenance method described above, in order to increase the efficiency of production plants.
To take this step, the following questions are identified as essential basis:

- How accurately is the remaining useful life (RUL) predicted by the proposed neural network architecture using an benchmark data set with known target variables and a well defined ground truth [SGSE08]?
- To what extent does the performance of the prediction degrade at the usage of a real world data set which entails missing values and mislabeled ground truth data?
- What measures can be taken to reduce the degradation of the performance?

To answer the questions the following methodology is pursued:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

1. A comprehensive literature review of methods used to overcome different aspects of machine health management in context of machine learning is conducted. The analysis covers knowledge based approaches, data driven approaches especially neural networks. Furthermore, a review of data mining and methods to overcome the missing data problem can be found in chapter 2.

2. Chapter 3 examines a case study of a prediction method for a scientifically simulated Run-to-failure data set with a known groundtruth. The introduced neural network architecture is inspired by state of the art architectures with a high performance on noisy and sparse data. The capabilities of the method are evaluated by different metrics, with focus on noise resilience and the degradation curve using sparse data. An outlook for potential improvements is given in the discussion.

3. A case study of a scenario with a high level of uncertainty and no known groundtruth is described in chapter 4. An iteratively tailored CRISP-DM [She00] method is applied to understand, prepare and model the data. Therefore, exploratory analysis, cleansing, reverse engineering of the ground truth and preprocessing of the data for machine learning and feature engineering steps are conducted in 6 iterations, each concluded with feedback of domain experts of the data. Basic machine learning algorithms are evaluated by different metrics, and discussed.

4. The performance of the network architecture on the two data sets is measured with different metrics described in Section 3.1.1. Degradation of the performance is implied by the distance between the metrics. Further analysis identifies potential starting points for measures to reduce the difference of the performance of the neural network.

5. In the discussion in chapter 5 the differences and implications for further generalization of the methods are analyzed.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

CHAPTER 2

State of the Art

Diagnostics and prognostics originally come from the medical field [LWZ$^+$14]. As the idea of machine health management grew, those two terms permeated different areas of mechanical engineering, introducing professional instruments, such as sensors, meters, and controllers, but also dedicated computational devices and algorithms. At first diagnostics were introduced in a reactive manner to support decisions of the maintenance experts. Nowadays, in order to maintain the up time of machines at the highest possible level, the strategy shifts to proactive AI based approaches, which forecasts machine health in different ways [KY18]. Figure 2.1 summarizes different categories of machine health management approaches. Viewed from top to bottom, the first category focuses on the knowledge of physic based rules and models. With those mathematical models, highly accurate results can be achieved if an appropriate model for a system is developed. However, it can be difficult to find the explicit mathematical model as the complexity of the system increases and the accuracy degrades quickly with noise [KY18]. Examples are ontology-based reasoning [ULRS11] or a model-based approach [BWES04].

The second category is summarized as probabilistic reasoning, which is already a step towards data driven methods. Examples for the estimation procedures are the Kalman filter [YL99] [AVR$^+$09], hidden Markov models [TMMZT12] and Bayesian networks [MZ09]. These two categories are specialized in specific scenarios and machines to ensure a high accuracy. The specialization has certain disadvantages. Changes to the machine imply effort to adopt the models, new devices require new or at least adjusted models. Also, due to the high degree of specialization this methods are susceptible to noise.
A step towards generalization of the methods is to move away from physics based models to the third category, which are data driven models. Data driven models can again be classified as supervised and unsupervised methods.
Unsupervised machine learning describes a set of algorithms applied to tasks like clustering, anomaly detection, association mining, or latent variable models to unlabeled data sets.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

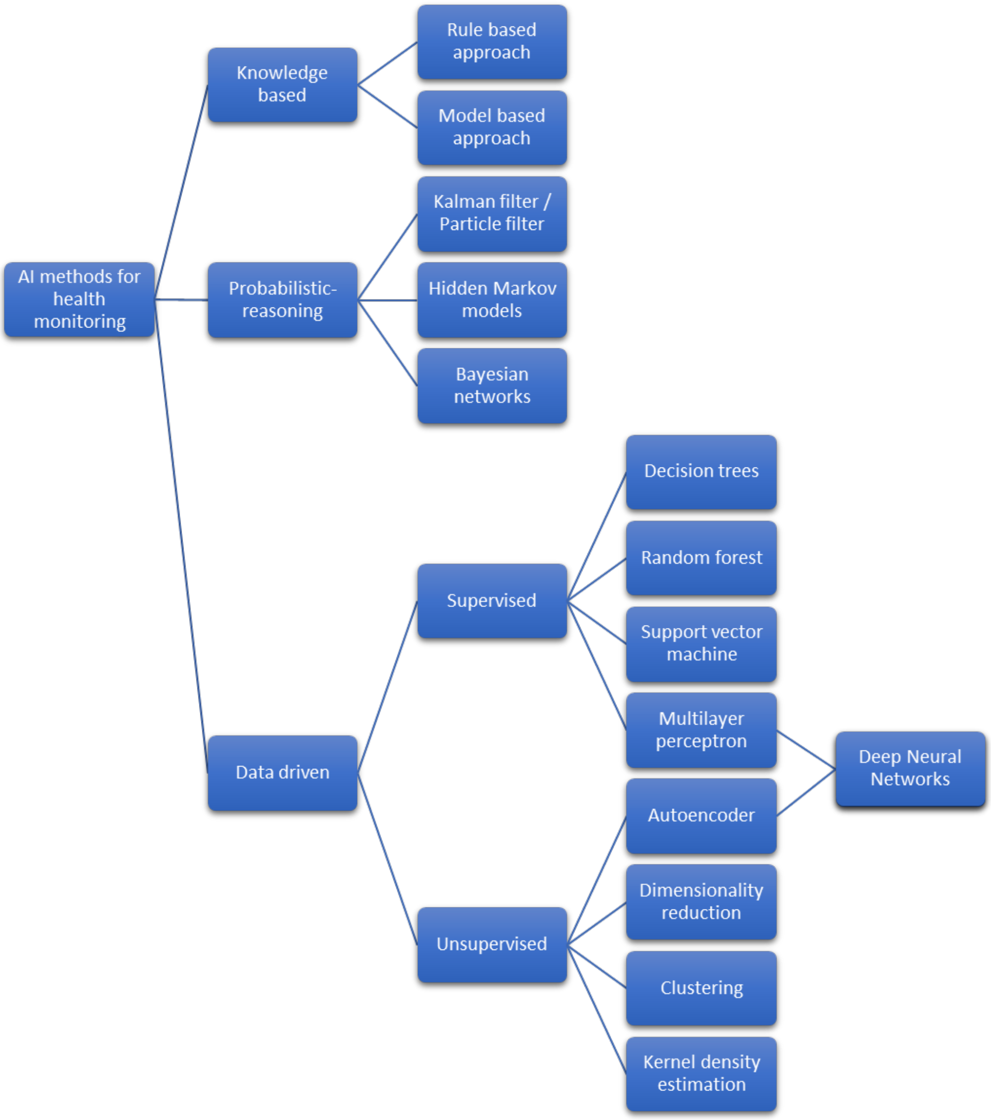


Figure 2.1: Selection of the categorization of AI based methods used in system health monitoring described by Kahn and Yairi [KY18].

That means unsupervised algorithms infer patterns with no knowledge or reference to the outcome. In the context of predictive maintenance, unsupervised machine learning is mainly used to cluster measurements, for example to healthy versus unhealthy data [AG18], or for parameter reduction and noise reduction, for example with Autoencoders

[LWQM17].
Supervised machine learning is a set of learning algorithms which map an input to an output based on a training set of input-output pairs. Therefore, a data set has to provide data points and a corresponding label, also called groundtruth. One of the early supervised machine learning algorithms are decision trees [Qui86]. Decision trees are assembled to random forests, which still is a widely used algorithm for simple classification or regression tasks [SMD17] [YDH08]. However, deep learning became a popular method for machine health management application over the last years [KY18].

2.1 Deep Learning

Deep learning is a re-branding of artificial neural networks and has its origin back in the 1950s, in the trial to replicate the functionality of the human brain [Sha86]. It is typically described as the application of a neural network with more than one hidden layer. The focus is to model high-level abstractions in data that can either be applied as supervised or unsupervised learning. Such a network consists of a number of layers which include neurons. Neurons between the layers are connected. The training itself is done by adapting the weights between the neurons. Due to its ability to extract hierarchical representation of data automatically on the one side and due to the deep architectures, to learn complex, non linear features on the other side, deep learning has been applied to various applications like computer vision, automatic speech recognition, natural language processing or audio recognition. The different use cases facilitate a variety of designs. The most commonly used architectures according to [KY18] are the following:

- Autoencoders
- Convolutional neural networks
- Recurrent neural networks

2.1.1 Autoencoders

Autoencoders are trained to attempt to copy its input to its output through a number of hidden layers, where the number of neurons of the layers in the middle of the network is typically smaller than the number of neurons at the input and output layers, as illustrated in figure 2.2. The smallest layer is called bottle neck. It splits the network to an encoder (part of the network between the input and the bottle neck) and a decoder (part of the network between the bottle neck and the output). After the training, only the encoder is used to reduce the number of dimensions. Therefore, the learned features are reduced to a limited number of weights between the neurons, which compresses the data in a way similar to PCA. That is used for parameter reduction and noise reduction. Lu et al. introduced an autoencoder architecture for noise reduction [LWQM17]. They proposed a stacked denoising autoencoder with 4 layers, where the number of neurons of every subsequent layer of the encoder is $\frac{n}{2}$, decreasing the input dimensions from 200 to 100,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

50 and 25. By this method they increase the classification accuracy of their data set of a vibration series generated by a bearing test-rig, by approximately 5%.

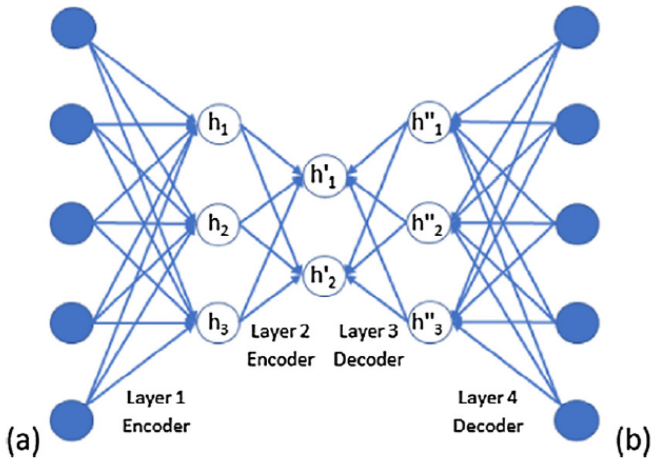


Figure 2.2: [KY18] This figure illustrates the structure of an autoencoder. Input (a) and Output (b) have the same number of dimensions, where the hidden layers show a reduced number of neurons.

2.1.2 Convolutional Neural Networks

A technique that evolved from high-dimensional data, such as images or time-series data, is the convolutional neural network, or short CNN. This specially designed network includes convolution layers as an automatic feature extractor before the classification or regression layer. A typical architecture consists of convolution and pooling layer pairs, as can be seen in Figure 2.3. As the name implies, the convolution layer converts the data using the convolution operation. That means for a two dimensional convolution, a nxm filter which is usually small in relation to the data, is convolved over the two dimensional data. One by one, the dot product is determined and recorded in a new two dimensional space. The pooling layer functions as an threshold and dimensionality reduction layer. As this type of network has its origin in image processing, the operations are conducted on two dimensional planes. To use the CNN architecture in the context of system health management, the input is the data's time and frequency representation images. Janssens et al. [JSV$^+$16] applied a CNN for monitoring a rotating machinery condition. The input is a discrete Fourier Transform of two accelerometers. By adding a regression layer, Babu et al. predicted the remaining useful life on the basis of the time series data of a system [BZL16]. They are able to outperform regression with a multilayer perceptron, the support vector regression, and the relevance vector regression.

As this technique is a well established alternative, it is mentioned for the sake of a complete overview and comparison with known approaches according to [KY18]. Methods in this thesis however are not based on CNNs.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

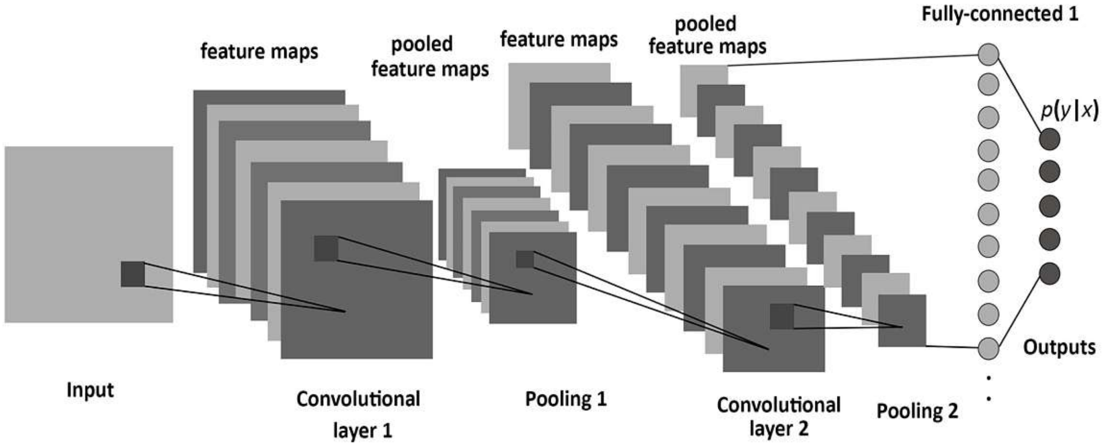


Figure 2.3: Structure of a CNN. [AM17]

2.1.3 Recurrent Neural Networks

Recurrent neural networks, short RNN, are developed to deal with sequential data. The main applications are speech recognition [MGM15] [FK18] [CJL⁺18] and time series analysis. Since sensor measurements for system health management typically consist of timestamp value pairs, RNNs are a suitable candidate for predictive maintenance. Generally, RNNs differ from simple neural networks by taking sequential information from the past into account. In a typical neural network structure, the hidden states $h_1, h_2, ...h_n$ are independent of each other. In an RNN architecture, the hidden state at each step depends on its precursor, as shown in Figure 2.4. Therefore, the output $\hat{y}_t$ is influenced not only by the input at the time t, but also at the time $t-1$. Mathematically, these process can be described using the transition function described in the equations 2.1 and 2.2.

$$h_t = f(w_{hx}x_t + w_{hh}h_{t-1}) \tag{2.1}$$

$$\hat{y}_t = f(w_{yh}h_t) \tag{2.2}$$

where f is an activation function. w_{hx} is the matrix of weights between the input layer and a hidden layer and w_{hh} is the matrix of weights between a hidden layer at the time steps t and $t-1$. That implies the architecture of a deep neural network with one layer per time step and can be trained by backpropagation [MBM15]. In deep network architectures RNNs suffer from the vanishing gradient problem: Typical activation functions have gradients in the range $(0, 1)$. Backpropagation computes gradients by the chain rule, which has the effect that the gradient which is used to adapt the weights to minimize the loss function decreases exponentially. Therefore, with deep architectures the gradient becomes too small to sufficiently train the front layers. To overcome this problem short

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

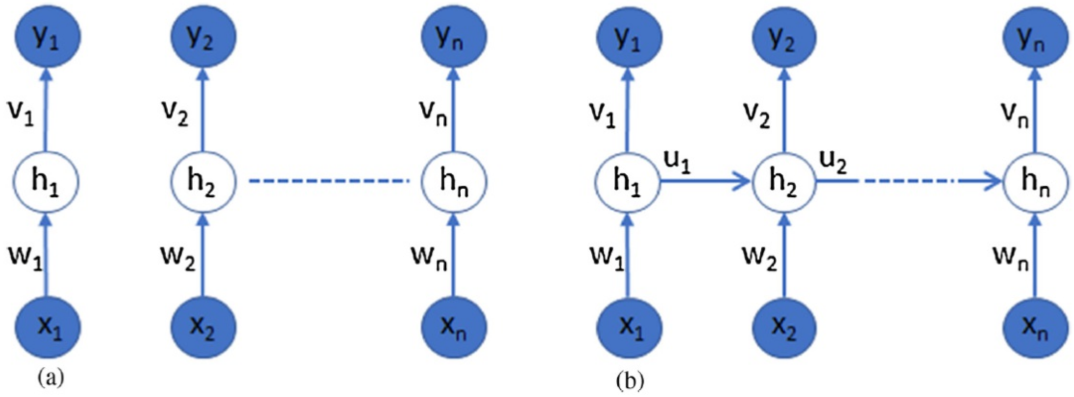


Figure 2.4: Comparison between a simple neural network (a) and the RNN structure (b). [KY18]

term memory techniques were introduced to RNN networks. The literature distinguishes two famous methods: the long short-term memory (LSTM) and the gated recurrent units (GRUs) [HHAL18]. Both replace the activation function in a cell with gated cells. These cells differ between the LSTMs and GRUs. Memory cells of an LSTM include three gates. The *forget gate* regulates if information will be remembered or forgotten, the *input gate* regulates the degree of significance of new information and the *output gate* regulates the extent to which the value in the cell is used for the calculation for the next module of the chain within the RNN. Gated cells of the GRU include only two gates. The *update gate* functions similar to a combination of the input gate and the forget gate of the LSTM and the *reset gate* regulates the extent to which past information should be forgotten, similar to the output gate of the LSTM. Due to the smaller complexity the GRUs are more time efficient. Yuan et al. compared the three variants RNN, GRU and LSTM and evaluated their performance on the turbofan data set [SGSE08], concluding, that the LSTM is the most suitable method in this setting [YWL16]. Therefore, the favored approach in chapter 3 is based on LSTMs.

2.2 Data Mining

Machine Learning in terms of training, predicting and evaluating is only part of a bigger process when it comes to performing data science tasks in an industrial environment like the scenario that is investigated in the second case study of this thesis. To describe the surrounding steps, the 'Cross industry standard for data mining', or CRISP-DM was introduced [She00]. CRISP-DM is an open standard for data mining. The iterative process can be seen in Figure 2.5. It groups the data mining task to the following 6 phases:

- **Business understanding**

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

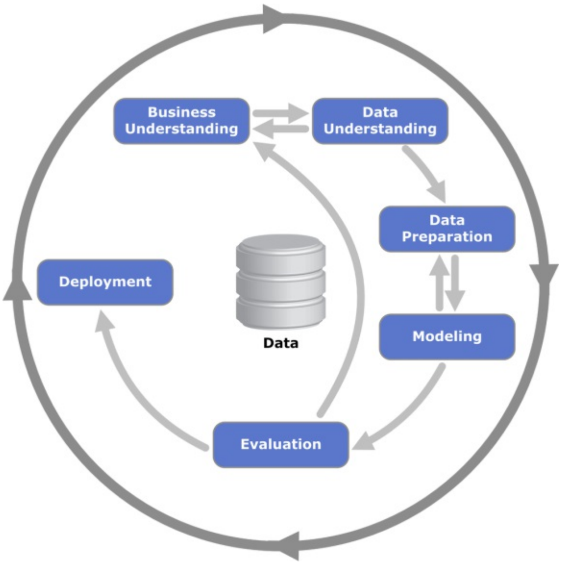


Figure 2.5: Standard process for extracting information from data, according to the 'Cross industry standard for data mining' [She00]

In the first step, the requirements from a business perspective are analyzed. The following tasks are performed in this step:

- Determine business objectives
- Situation assessment
- Determine data mining goal
- Produce project plan

- **Data understanding**
  In an iterative loop back to the business understanding step, the data is collected and analyzed. The main view concentrates on whether the data is sufficient to meet the business goals defined in the first step. The tasks below refine the second step.
  - Collect initial data
  - Describe data
  - Explore data
  - Verify data quality
- **Data preparation**
  With satisfactory business and data understanding the data is prepared for further processing. The tasks below are conducted to, for example, convert the data to a tabular format, to cope with missing values, to convert data to different types or to scale numerical values. Especially missing values are an important subject in thesis. Further details on that topic can be found in the following section.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

  - Select data
  - Clean data
  - Construct data
  - Integrate data
  - Format data

- **Modeling**
  In the modeling step machine learning models are selected and refined. Since models require certain formats, finding a model and prepare the data appropriately is an iterative process, involving the following tasks:
  - Select modeling technique
  - Generate test design
  - Build model
  - Assess model

- **Evaluation**
  Finally three evaluation tasks are conducted. The technique is evaluated against the business requirements defined in the first step, the process is reviewed and as result next steps are discussed. If the results of the technique are not sufficient, the requirements are reviewed and a new iteration of the steps above is started.
  - Evaluate results
  - Review process
  - Determine next steps

- **Deployment**
  Besides the further refinement of the technique the current version can be deployed to the production environment. This step is not relevant in this thesis. Data mining in the second case study is based on theoretical analysis and consultation of stakeholder.
  - Plan deployment
  - Plan monitoring and maintenance
  - Produce final report
  - Review project

The CRISP-DM describes an established sequence of steps for performing qualitative data mining tasks. Quality is achieved by the following measurements: By first understanding the business goal and the data the further analysis steps can be executed targeted purposefully towards the goal. Clear requirements also enable a detailed design of the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

evaluation metrics. Iterations between business understanding and data understanding increase the clarity of the requirements and the initial understanding of the data, as well as the selective collection and verification of the data. For the exploratory analysis in the second case study of this thesis, the iterations between data collection and analysis steps, and the domain experts and stakeholder of the business cases are a mayor part, in order to be able to develop approaches to meet the goals of the stakeholders.
Also the quality of the data preparation and modeling steps is increased by a feedback loop, providing the chance to increase the quality in every iteration. After the evaluation, the model can be deployed but the process starts with step one for the next iteration of maximizing the quality of the performance with new additional data or improvements of the steps. As described above, iterative improvement and feedback loops play a mayor part for the quality of the output with the CRISP-DM process. Hence, the methodology in this thesis focuses on them as well.

2.3 Missing Data

The third step of the CRISP-DM process is data preparation. Tasks to ensure quality within the data including data selection, cleansing, constructing, integrating and formatting, are typically addressed by scripted procedures. One issue during the data preparation step proposes a challenge in particular in the second case study, and therefore is examined more in detail.
The term 'missing data' describes missing values or continuously false values. There are different reasons for missing data like environmental disturbances, broken sensors, unclean implementation of the persistence of measurements and many more. Especially in the second case study of this thesis the main reasons were slow and non synchronous sampling frequencies between the sensors. For data mining methods, missing values or false values may give false or insufficient results. Therefore understanding the data to find the appropriate method for coping with missing values is an important step. Imputation methods should both find the value closest to the true value and maintain the data structure [BS16]. Another dimension is the complexity of the used method. Both, time and effort have to be considered and should not extend the scope of this thesis.

According to [CFB+03], there are different strategies to handle missing data:

1. Ignore the tuples.
   This is a time effective method, since no additional calculations are needed. But useful or critical information may be lost in the process.

2. Fill the value manually.
   Manually filling in values is a time consuming task, requires domain knowledge and may not be feasible with large data sets. Also here is potential of human bias.

3. Use a global constant to fill missing values.
   This is a simple method where a label for 'unknown' is set or an integer value which

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

does not occur in the data is used as placeholder for missing values. These constant values may have an impact on data mining methods since all missing values are substituted by the same constant and therefore are considered interesting.

4. Use a measure of central tendency to fill the missing values, such as mean or median. A more advanced method is to first classify the data and then determine the mean or median for each class.

5. Use the most probable value to fill the missing values.
   Values are filled by predicting the most probable value by methods such as Maximum Likelehood predictions. These methods build models with the existing data. The models then are used to predict the missing values.

All procedures except the first may bias the data, since the inferred values may be wrong. That means conversely, that potentially biased data is the trade of to preserve information. Manually filling values is not a feasible approach for this thesis due to two reasons. First, the domain experts were only available for short discussions or questions. Even with extensive availability, the amount of missing data was too much to be filled in manually in the scope of this thesis. To minimize the complexity in order to focus on the degradation of the proposed methodology, predictions for missing values are not used in this thesis. But as stated in the conclusion, using advanced methods for data imputation wold be an interesting approach for future work. Besides ignoring tuples, this leaves the measure of central tendency and using a global constant. These three strategies are used in the second case study as described in section 4.4.2.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 3

Study of the Reference Data Set C-MAPSS [SGSE08]

For the first case study, the CRISP-DM methodology is used to develop a predictive maintenance algorithm for a widely used scientific data set with the goal to examine the possibilities in a well defined environment. Therefore, the C-MAPSS turbo engine data set [SGSE08] is chosen. Besides the clearly defined properties of the data set, a difference to the second case study is that this data set is used in a variety of papers since 2008. This provides a good entry point to different methodologies, as well as the possibility to compare methodology and results to various publications, which is not only interesting for drawing conclusions, but also provides a basis for further research. Further, sparsity of the data is simulated and the degradation of the results are measured and discussed.

3.1 Methodology

As described in Section 3.2, the groundtruth for the C-MAPSS data set is known and it is well defined. It contains data from different devices, which can directly be merged to one data frame without further preprocessing or harmonizing steps. There are 100 run-to-failure scenarios and time series data that leads to the failure. Over all the CRISP-DM process is followed for this case study. The objective for this case study is to refine a method which is robust for a noisy data set which will further be used for the second case study. The machine learning task is defined similar to the assignment in the PHM08 challenge [SGSE08]. The underlying process used for data mining in this thesis is CRISP-DM as described in Chapter 2. Since the data set is generated artificially and well documented for the challenge, the data understanding step can be done by literature review. The data preparation is reduced to format the data to suit the input requirements of the used frameworks. Therefore, this case study focuses mainly on the refinement iterations of data preparation, modeling and evaluation of the model, in order to find the most capable approach and hyperparameters for the given application. In a last step the degradation with sparse data is evaluated.

After extensive research of state of the art architectures of neural networks, which is summarized in the Chapter 2, a bidirectional LSTM architecture is used for this thesis. First the hyperparameters are evaluated for a network with one hidden layer. After that, experiments including deeper architectures and an autoencoder for noise reduction are conducted and evaluated with the goal of refining the accuracy. To substantiate the decisions in the refinement iterations, 4 different metrics are used. The metrics are described in detail in Section 3.1.1. For the implementation, python version 3.6 is used with the Keras framework [C+15]. Keras works on top of the TensorFlow framework [AAB+15], which automatically assumes the architecture of a network given a variety of hyperparameters.

3.1.1 Evaluation metrics

To evaluate different views of the results, four different metrics are used for the hyperparameter evaluation as well as for the evaluation of the proposed method.

Mean Absolute Error

The mean absolute error (MAE) measures the distance between two continuous variables, as can be seen in equation 3.1. For the evaluation of the machine learning methods used in this thesis, the distance of the predicted remaining useful life $\hat{X}$ and the actual remaining useful life X is described. n is the number of samples.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{X}_i - X_i| \tag{3.1}$$

Mean Squared Error

Another quality criterion to evaluate a regression estimator is the mean squared error (MSE). It measures how much a point estimator scatters around the value that is estimated. Compared to the MAE, it emphasizes large aberrations of the estimated value compared to the ground truth by squaring their difference. If a vector of n predictions is generated from a sample of n data points, and X is the vector of observed values of the variable being predicted as $\hat{X}$, then the MSE of the predictor is computed as can be seen in equation 3.2.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{X}_i - X_i)^2 \tag{3.2}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Coefficient of Determination

The coefficient of determination, also called R^2, measures the proportion of the variance in the predicted variable from the independent variable(s). Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable. It is therefore a measurement for how well the known outcome is reproduced by a model, based on the total proportion of the variation of predictions, as can be seen in equation 3.3. Given a data set of n values, with a known ground truth X_i, each associated with a predicted value $\hat{X}$ and $\tilde{X}$ is the mean of the observed data, the coefficient of determination is determined by as follows:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{3.3}$$

where SS_{res} is the sum of sqares of residuals, given by

$$SS_{res} = \sum_i (X_i - \hat{X})^2 \tag{3.4}$$

and SS_{tot} is the total sum of squares, given by

$$SS_{tot} = (X_i - \tilde{X})^2 \tag{3.5}$$

Score

Saxena et al. [SGSE08] propose a score for the evaluation of the remaining useful life reflecting on the key aspect for machine health prognostics being the avoidance of failure. Therefore, for an engine degradation scenario an early prediction is preferred over late predictions. Thus, late predictions are penalized more heavily than early predictions, as can be seen in equation 3.6. Given

S is the computed score,
n is the number of predictions,
d is the difference (estimated RUL - true RUL),
$a_1 = 10$ and
$a_2 = 13$

the score is determined as follows:

$$s = \begin{cases} \sum_{i=1}^{n} e^{-(\frac{d}{a_1})} - 1 \text{ for } d < 0 \\ \sum_{i=1}^{n} e^{-(\frac{d}{a_2})} - 1 \text{ for } d \geq 0 \end{cases} \tag{3.6}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.2 Description of the Data Set

The C-MAPSS data set is a simulated run to failure data set provided by NASA. It is created by the simulation of a commercial aircraft engine of the $90,000$ lb thrust class. For the purpose of failure modeling the degradation of the high pressure compressor (HPC) is modeled. This component is one of a set of regulators and limiters of the simulation. It prevents the static pressure from going too low. A comprehensive logic structure integrates the components in a manner similar to that used in real engine controllers. Additionally, an initial degradation $d > 0$ is added, allowing the data generation process to start at an arbitrary point in the wear space. To consider various noise sources like manufacturing and assembly variations, process noise, measurement noise and others, a mixture of distributions is combined to add non-trivial noise in a realistic manner. The data is provided in csv format. It contains the following entries:

- Unit number (ascending identifier for each of the engines)
- Ascending cycles for time measurement per engine
- 3 operational settings
- 21 sensor measurements

The purpose of the settings and measurements is not specified. The set is already split into a training set and a test set. The training set consist of 20631 rows of data, which contain 100 devices. For each device the operational settings and sensor measurements are available until the engine fails. Therefore the remaining useful life is known implicitly by backwards counting of the the cycles. The lifespans of the engines vary between 128 and 362 cycles. Table 3.1 shows some basic statistics of the data.

Figure 3.1 shows boxplots of the C-MAPSS data set. Due to the degradation curve data points outside the the $0,25$ and $0,75$ quartiles exist.

The data set is known to be run to failure with a provided ground truth. The degradation can be seen in the example measurements for the first unit within the training data set as shown in Figure 3.2. The concealment of the interpretation or meaning of the measurements or operational settings is intentional and part of the challenge. Therefore, there are no further analysis steps conducted towards the interpretation of the data. Also, a reverse engineering of the groundtruth as described in the second case study is not necessary. For the evaluation set, the groundtruth is directly provided as a file containing the remaining useful life after the last provided measurement. The training set is a run to failure set. Therefore, to receive the groundtruth for each cycle, the difference to $m_{last} + 1$ has to be counted, where m_{last} is the last measurement before the failure.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

attribute	min	max	mean	median	standard deviation
op-s1	-8,70E-03	8,70E-03	-8,87E-06	0,00E+00	2,19E-03
op-s2	-6,00E-04	6,00E-04	2,35E-06	0,00E+00	2,93E-04
op-s3	1,00E+02	1,00E+02	1,00E+02	1,00E+02	0,00E+00
n 1	5,19E+02	5,19E+02	5,19E+02	5,19E+02	6,54E-11
n 2	6,41E+02	6,45E+02	6,43E+02	6,43E+02	5,00E-01
n 3	1,57E+03	1,62E+03	1,59E+03	1,59E+03	6,13E+00
n 4	1,38E+03	1,44E+03	1,41E+03	1,41E+03	9,00E+00
n 5	1,46E+01	1,46E+01	1,46E+01	1,46E+01	3,39E-12
n 6	2,16E+01	2,16E+01	2,16E+01	2,16E+01	1,39E-03
n 7	5,50E+02	5,56E+02	5,53E+02	5,53E+02	8,85E-01
n 8	2,39E+03	2,39E+03	2,39E+03	2,39E+03	7,10E-02
n 9	9,02E+03	9,24E+03	9,07E+03	9,06E+03	2,21E+01
n 10	1,30E+00	1,30E+00	1,30E+00	1,30E+00	4,66E-13
n 11	4,69E+01	4,85E+01	4,75E+01	4,75E+01	2,67E-01
n 12	5,19E+02	5,23E+02	5,21E+02	5,21E+02	7,38E-01
n 13	2,39E+03	2,39E+03	2,39E+03	2,39E+03	7,19E-02
n 14	8,10E+03	8,29E+03	8,14E+03	8,14E+03	1,91E+01
n 15	8,32E+04	8,58E+04	8,44E+04	8,44E+04	3,75E+02
n 16	3,00E-02	3,00E-02	3,00E-02	3,00E-02	1,56E-14
n 17	3,88E+02	4,00E+02	3,93E+02	3,93E+02	1,55E+00
n 18	2,39E+03	2,39E+03	2,39E+03	2,39E+03	0,00E+00
n 19	1,00E+02	1,00E+02	1,00E+02	1,00E+02	0,00E+00
n 20	3,81E+01	3,94E+01	3,88E+01	3,88E+01	1,81E-01
n 21	2,29E+05	2,36E+05	2,33E+05	2,33E+05	1,08E+03

Table 3.1: Statistics of the C-MAPSS data set. The first three rows are operational settings, followed by 21 rows of sensor measurements.

3.3 Base Line

As stated at the beginning of this chapter, the results are compared with state of the art methods which published results on the C-MAPSS data set in table 3.26. For further comparison with a well known method that is not based on neural networks, a base line is created by training a random forest regression algorithm. This algorithm is described as a basic technique with high performance by Khan [KY18]. In this thesis, the random forest is chosen as a base line algorithm due to its fast performance, deterministic characteristic and because it can be called and evaluated easily by the *scikit-learn* framework [PVG+11]. To approximate a maximum for the method, five hyperparameters provided by the framework are evaluated in a grid search. Table 3.2 shows the range in which the evaluation was done and the parameters that enabled the best performance. The scores reached by the algorithm after refining the hyperparameters can be seen in Table 3.3.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

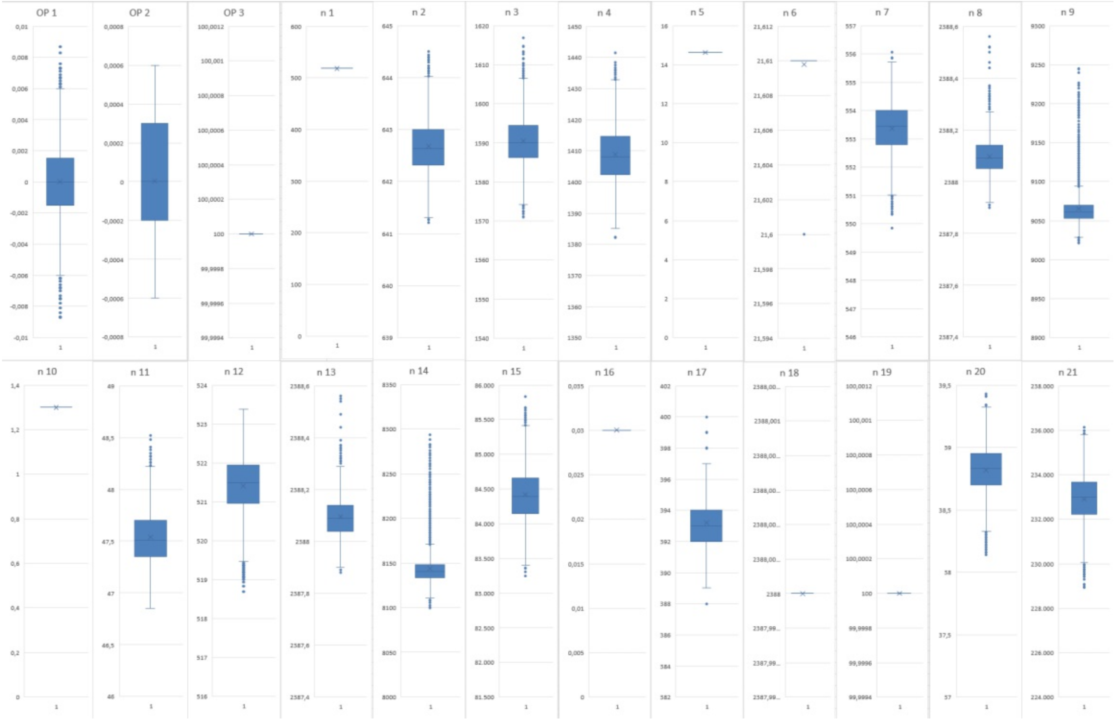


Figure 3.1: Boxplots of the C-MAPSS data set.

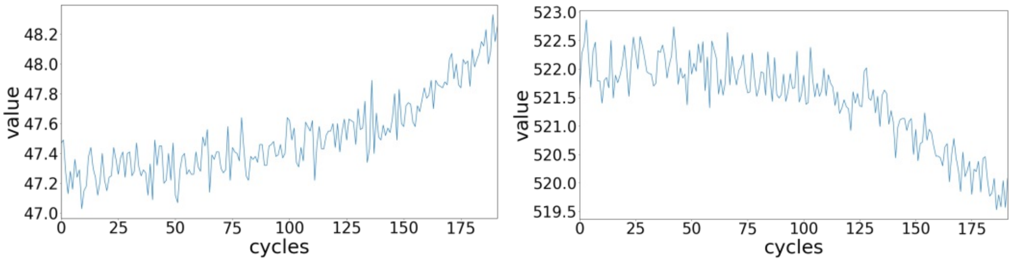
Evaluated hyperparameter	**evaluated values for the parameter**	**chosen parameter**
The Maximum depth of a tree	5, 20, 50, 80, 110, 150	150
The number of features to consider when looking for the best split	2, 12, 24	2
The minimum number of samples required to be at a leaf node	1, 3, 5, 7	1
The minimum number of samples required to split an internal node	10, 24	10
The number of trees in the forest	100, 200, 300, 1000	300

Table 3.2: Grid search of the hyperparameters for the random forest regression algorithm and the parameters with the best result.

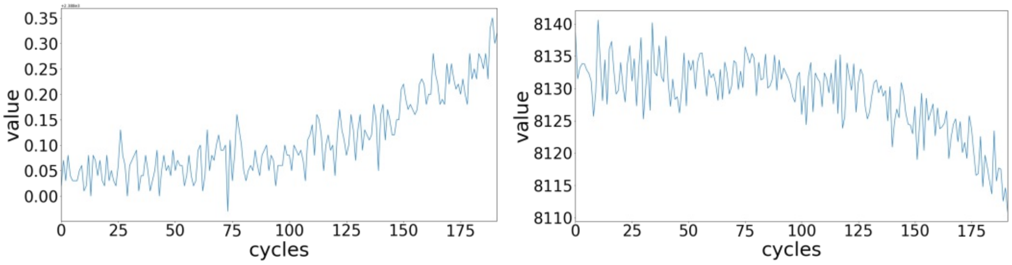
3.4 Data preparation for Machine Learning

To prepare the training data for machine learning, the unit number and the cycles are removed. In the next step, the data is rescaled to a range between 0 and 1 with a min-max normalization, to allow the gradient descent to converge fast [RL16], [SOVP12]. Since the used framework requires input data to be scaled between 0 and 1, the min-max normalization is chosen over the z-score normalization.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub



(a) Sensor 11 (b) Sensor 12



(c) Sensor 13 (d) Sensor 14

Figure 3.2: Selected sensor data of C-MAPSS data for device 1. The degradation of the device can be seen in the curves.

MAE	**MSE**	R^2	**Score**
23,73	995,01	0,42	22152,25

Table 3.3: Performance of the random forest prediction algorithm after the hyperparameter evaluation. This results are produced by using the parameters of table 3.2

The data set is split into a train set and a test set. Furthermore, the training data is split to a train set and an evaluation set by an 80:20 ratio, to evaluate the learning progress and detect overfitting while applying the training. The structure of the data is adapted to a format that is accepted by the Keras framework. Therefore, a window of the size $n_{timesteps}$ is convoluted over the rows of measurements, which contain 24 values each. The content of the window is concatenated to vectors of the length $24 * n_{timesteps}$, creating vectors which contain the features of a time span of $n_{timesteps}$. The label for each vector is the remaining useful life until the known breakdown of the last set of 24 measurements in the vector, which is the n^{th} timestep. The generation of the feature vector is illustrated in Figure 3.3.

After the above steps, the training set, the evaluation set and the test set are prepared properly as input data for the LSTM provided by the Keras framework.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

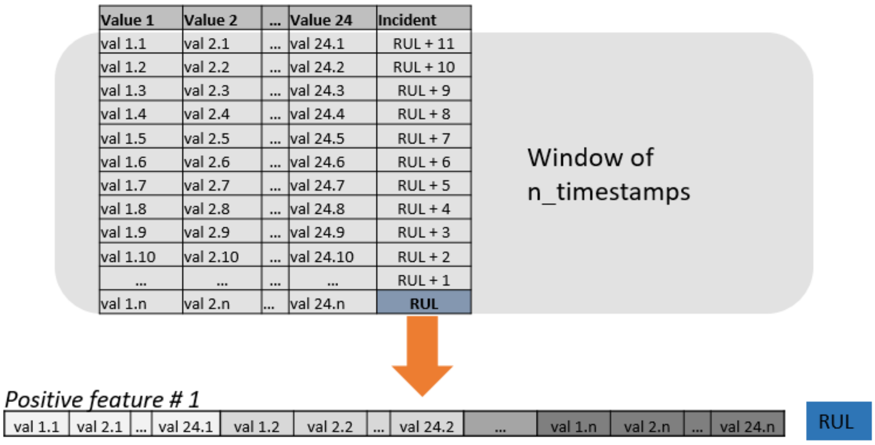


Figure 3.3: Generation of a feature vector by sliding a window of the size $n_{timesteps}$ over the rows of measurement.

3.5 Hyperparameter Evaluation

To find a performance optimum for the LSTM with one hidden layer, the hyperparameters of the network are optimized. The evaluation is not done by an over all grid search because the duration of the experiment is not acceptable for this thesis.
To give an example: The first parameter that is evaluated are the number of rows within the data, which are concatenated to generate one feature vector as described in Figure 3.4. To preserve the terminology used by the creators of the data set, the number of rows used to concatenate the feature vector is called *cycles*. Details of the experiment are described in Section 3.5.1. The 20 executed rounds of training the data set and evaluation had a duration of approximately 19 hours on an Intel(R) Core(TM) i7-7500U CPU with $2,7$GHz. To enhance the efficiency, a subset of parameters which is assumed to have a large impact is evaluated independently or in pairs. To find an optimum, the metrics described in Section 3.1.1 are used to evaluate the predictions. Since 4 metrics are used (MAE, MSE, R^2 and score) a ranking is created by assigning penalties to the results as follows: The best performing parameters are penalized with 1 point. For the following rankings, the penalty is increased by 1. That means the n^{th} place is penalized with n points if there are no equal results. The final ranking is the sum of the penalty points of the different metrics.

The following parameters are evaluated:

- Number of cycles used for a feature vector
- Overlapping of the time while creating input data
- Number of epochs and the batch size

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- Optimization algorithm
- Loss function
- Number of neurons in the hidden layer

3.5.1 Number of Cycles Used for a Feature Vector

Keras accepts the input for a LSTM in the format $[samples, timesteps, features]$. The parameter *samples* is the number of different devices, in this case 100 including trainings set and test set. The *features* value describes the number of measurements or input variables at one data point. To do supervised machine learning with time series data, n rows within the data, here called cycles, are concatenated to one input vector as shown in Fgure 3.3. The parameter *timesteps* describes the number of cycles that are concatenated. The values for *samples* and *features* are predefined by the data set. Therefore, the number of cycles used to generate one vector of features is evaluated as follows. The input is reshaped to a length of 5 to 100 cycles, increasing the length by steps of 5. The remaining parameters which have not been evaluated at the time of the experiment are assumed by an educated best guess as shown in Table 3.4

Parameter	**value**
Overlapping of the time while creating input data	no overlapping
Number of epochs	50
Batch size	72
Optimization algorithm	adam
Loss function	mae
Number of neurons in the hidden layer	50

Table 3.4: Best guesses of the remaining parameters while doing the evaluation for the number of cycles used for the feature vectors.

The results of the evaluation can be seen In Table 3.5. The best MAE of $12,04$ is reached with 55 cycles. With 65 cycles the best MSE $(274,94)$ and the best R^2 $(0,85)$ is reached, while the best score of $390,33$ is accomplished with 100 cycles in one batch. After the final ranking, a number of 65 cycles is assumed best fitting for this scenario.

3.5.2 Overlapping of the Time Windows while Creating Input Data

While creating feature vectors as described in Section 3.5.1, the vectors can either be generated cycle by cycle, or by a step of k, while $k \leq n_c$ used for one feature vector, where n_c is the number of cycles used as discussed above. If $k = n_c$, the vectors do not overlap. Otherwise, $n_c - k$ data points overlap, as illustrated in Figure 3.4

The overlapping of 0 to 40 in steps of 10 is evaluated in Table 3.7. The other parameters are again assigned values according to an educated best guess. The values are listed in Table 3.6

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Number cycles	Sum of penalties	Custom score	MAE	MSE	R^2	Penalty score	Penalty MAE	Penalty MSE	Penalty R^2
65	13	562,85	12,92	**274,94**	**0,85**	9	2	1	1
55	20	950,52	**12,04**	287,09	0,84	12	1	5	2
100	22	**390,33**	12,95	284,57	0,76	1	3	3	15
50	23	576,33	13,03	287,71	0,84	10	4	6	3
80	27	540,69	13,11	297,90	0,81	7	5	8	7
85	29	450,45	14,01	277,19	0,77	3	10	2	14
90	29	450,56	13,30	285,74	0,77	4	8	4	13
60	30	539,12	13,28	299,52	0,80	6	7	9	8
40	33	842,81	13,26	319,95	0,83	11	6	12	4
70	36	446,23	14,12	301,83	0,78	2	12	10	12
95	39	462,03	14,12	288,21	0,75	5	11	7	16
30	40	1685,42	13,34	332,97	0,83	13	9	13	5
75	44	548,05	14,55	318,85	0,78	8	14	11	11
35	47	3048,34	14,43	399,95	0,82	14	13	14	6
25	54	3313,01	15,36	417,96	0,79	15	15	15	9
45	59	5927,98	15,45	447,25	0,79	17	16	16	10
20	67	3520,36	17,53	529,06	0,73	16	17	17	17
15	72	7527,74	20,14	728,52	0,70	18	18	18	18
10	76	12632,93	23,72	941,29	0,61	19	19	19	19
5	80	24995,90	26,54	1158,70	0,57	20	20	20	20

Table 3.5: Metrics and the rankings for the evaluation of the number of cycles, which are packed to one vector to create the input for the LSTM.

Parameter	value
Number of cycles used for a feature vector	65
Number of epochs	50
Batch size	72
Optimization algorithm	adam
Loss function	mae
Number of neurons in the hidden layer	50

Table 3.6: Best guesses of the remaining parameters while doing the evaluation for the number of cycles that are overlapping.

An overlapping of 30 cycles achieves the best results according to the MAE, MSE and the R^2 metric and therefore is also ranked best in total.

3.5.3 Number of Epochs and the Batch Size

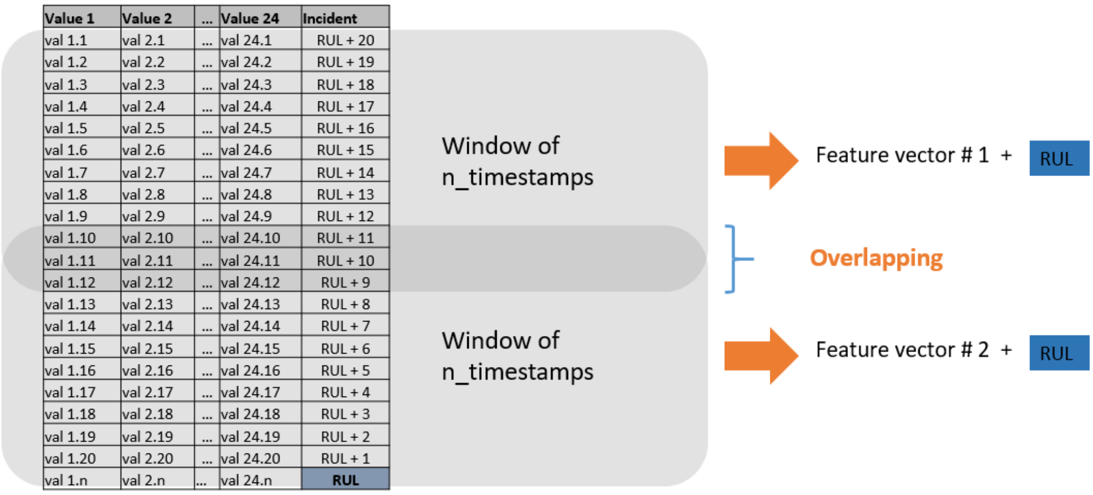
The number of epochs for the training and the batch size are evaluated as pairs in a grid search. The other hyperparameters are listed In Table 3.8. Table 3.9 shows the top 10 rankings with the values of the different used metrics.

It can be seen that the best ranking is achieved with a batch size of 60 and a training over 40 epochs.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub



Value 1	Value 2	...	Value 24	Incident
val 1.1	val 2.1	...	val 24.1	RUL + 20
val 1.2	val 2.2	...	val 24.2	RUL + 19
val 1.3	val 2.3	...	val 24.3	RUL + 18
val 1.4	val 2.4	...	val 24.4	RUL + 17
val 1.5	val 2.5	...	val 24.5	RUL + 16
val 1.6	val 2.6	...	val 24.6	RUL + 15
val 1.7	val 2.7	...	val 24.7	RUL + 14
val 1.8	val 2.8	...	val 24.8	RUL + 13
val 1.9	val 2.9	...	val 24.9	RUL + 12
val 1.10	val 2.10	...	val 24.10	RUL + 11
val 1.11	val 2.11	...	val 24.11	RUL + 10
val 1.12	val 2.12	...	val 24.12	RUL + 9
val 1.13	val 2.13	...	val 24.13	RUL + 8
val 1.14	val 2.14	...	val 24.14	RUL + 7
val 1.15	val 2.15	...	val 24.15	RUL + 6
val 1.16	val 2.16	...	val 24.16	RUL + 5
val 1.17	val 2.17	...	val 24.17	RUL + 4
val 1.18	val 2.18	...	val 24.18	RUL + 3
val 1.19	val 2.19	...	val 24.19	RUL + 2
val 1.20	val 2.20	...	val 24.20	RUL + 1
val 1.n	val 2.n	...	val 24.n	**RUL**

Figure 3.4: Feature generation by overlapping windows.

Over-lapping	**Sum of penalty**	**Custom score**	**MAE**	**MSE**	**R^2**	**Penalty score**	**Penalty MAE**	**Penalty MSE**	**Penalty R^2**
30	5	650,16	**12,63**	**288,50**	**0,85**	2	1	1	1
10	11	**541,89**	13,51	333,00	0,75	1	2	3	5
40	12	893,09	13,94	326,09	0,82	4	4	2	2
0	14	661,14	13,64	338,96	0,80	3	3	4	4
20	18	1000,94	14,81	362,47	0,81	5	5	5	3

Table 3.7: Metrics and the rankings for the evaluation of the overlapping.

Parameter	**value**
Number of cycles used for a feature vector	65
Optimization algorithm	adam
Loss function	mae
Number of neurons in the hidden layer	50

Table 3.8: Best guesses of the remaining parameters while doing the grid search evaluation for the number of epochs and the batch size.

3.5.4 Optimization Algorithm

For optimization, the framework provides 7 possibilities. The different optimization algorithms are evaluated ranked by the metrics discussed above. Table 3.11 shows the evaluation sorted by the best overall ranking ascending. It can be seen that the over all best performance is achieved by the *adamax* optimization algorithm. The other hyperparameters are again listed In Table 3.10.

Number of epochs	**Batch size**	**Sum of penalty**	**Penalty score**	**Penalty MAE**	**Penalty MSE**	**Penalty R^2**
40	60	8	1	1	1	5
100	60	40	5	2	4	29
70	100	49	15	10	3	21
40	30	51	16	16	10	9
50	60	55	11	22	8	14
70	80	58	10	19	11	18
110	100	71	65	3	2	1
30	50	80	4	21	7	48
30	30	85	3	14	5	63
80	40	92	19	6	27	40

Table 3.9: Metrics and the rankings for the grid search of the number of epochs and the batch size.

Parameter	**value**
Number of cycles used for a feature vector	65
Overlapping of the time while creating input data	30
Number of epochs	40
Batch size	60
Loss function	mae
Number of neurons in the hidden layer	50

Table 3.10: Best guesses of the remaining parameters while doing the evaluation for the optimization algorithm.

Optimization algorithm	**Sum of penalty**	**Custom score**	**MAE**	**MSE**	**R^2**	**Penalty score**	**Penalty MAE**	**Penalty MSE**	**Penalty R^2**
adamax	5	**428,21**	**12,97**	**286,12**	0,80	1	1	1	2
nadam	8	914,79	14,03	352,97	**0,83**	3	2	2	1
adadelta	14	604,19	14,82	361,38	0,72	2	3	3	6
adagrad	16	1604,63	15,12	401,85	0,79	5	4	4	3
rmsprop	19	1315,93	16,21	432,94	0,77	4	5	5	5
adam	22	1853,19	16,68	474,17	0,77	6	6	6	4
sgd	28	183188,91	38,99	2509,41	-326,87	7	7	7	7

Table 3.11: Metrics and the rankings for the evaluation of the optimization algorithm.

3.5.5 Loss Function

Table 3.13 shows the evaluation of the loss functions. It can be seen that the *mean absolute error* is ranked first by every used metric, and therefore is assumed to be the best loss function. The hyperparameters used for this evaluation run can be seen In Table 3.12.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Parameter	**value**
Number of cycles used for a feature vector	65
Overlapping of the time while creating input data	30
Number of epochs	40
Batch size	60
Optimization algorithm	adamax
Number of neurons in the hidden layer	50

Table 3.12: Best guesses of the remaining parameters while doing the evaluation for the loss function.

Loss function	**Sum of penalty**	**Custom score**	**MAE**	**MSE**	**R^2**	**Penalty score**	**Penalty MAE**	**Penalty MSE**	**Penalty R^2**
mean_absolute _error	4	**7,10E+02**	**1,39E+01**	**3,29E+02**	**7,83E-01**	1	1	1	1
logcosh	8	8,41E+02	1,50E+01	3,58E+02	7,60E-01	2	2	2	2
mean_squared _logarithmic_error	12	1,35E+03	1,68E+01	4,40E+02	7,52E-01	3	3	3	3
mean_squared_error	16	4,55E+03	2,35E+01	7,87E+02	5,78E-01	4	4	4	4
mean_absolute _percentage_error	20	5,13E+05	7,44E+01	7,26E+03	-5,53E+08	5	5	5	5
categorical_hinge	24	5,16E+05	7,45E+01	7,27E+03	-3,69E+09	6	6	6	6
hinge	32	2,94E+16	2,86E+02	8,38E+04	-1,22E+11	8	8	8	8
squared_hinge	28	2,94E+16	2,86E+02	8,38E+04	-2,22E+11	7	7	7	7

Table 3.13: Metrics and the rankings for the evaluation of the loss function.

3.5.6 Number of Neurons in the Hidden Layer

For the evaluation of the number of neurons used in the hidden layer the the parameters are initialized as listed In Table 3.14. The number of neurons is evaluated between 5 and 160. The metrics and ranking can be seen In Table 3.15. The best performance is achieved with 155 neurons.

Parameter	**value**
Overlapping of the time while creating input data	30
Number of cycles used for a feature vector	65
Number of epochs	40
Batch size	60
Optimization algorithm	adamax
Loss function	mae

Table 3.14: Best guesses of the remaining parameters while doing the evaluation for the number of neurons used in the hidden layer.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Number neurons	Sum of penalty	Custom score	MAE	MSE	R^2	Penalty score	Penalty MAE	Penalty MSE	Penalty R^2
155	6	439,98	**11,44**	**235,59**	**0,85**	3	1	1	1
65	10	**397,57**	11,86	260,35	0,83	1	2	4	3
160	11	408,82	12,49	256,75	0,83	2	3	2	4
145	12	442,96	13,13	257,28	0,82	4	4	3	5
125	22	462,93	13,53	272,31	0,82	5	8	5	8
35	22	599,23	13,26	295,38	0,81	6	5	6	9
85	26	723,07	13,32	303,57	0,83	13	6	8	2
55	30	604,03	13,61	302,39	0,80	8	9	7	10
75	35	700,42	14,00	305,11	0,82	12	12	9	6
15	36	613,78	13,51	317,04	0,78	9	7	10	14
25	39	602,88	13,82	334,06	0,77	7	10	11	15
135	45	670,91	14,61	357,01	0,80	11	14	12	12
115	47	1274,09	13,95	379,17	0,82	17	11	16	7
105	49	664,31	14,45	369,08	0,77	10	13	14	16
45	52	998,47	15,24	372,54	0,80	14	16	15	11
95	52	1018,67	14,91	367,81	0,80	15	15	13	13
5	63	1115,00	20,19	567,95	0,40	16	17	17	17

Table 3.15: Metrics and rankings for the evaluation of the number of neurons used in the hidden layer.

3.6 Experimental Results

After the optimization of the hyperparameters for the LSTM, the network is evaluated and the architecture is extended step by step on the basis of state of the art literature described in Section 2. The evaluation of the hyperparameters as well as the evaluation of the experiments is based on 4 different metrics as described in the Section 3.1.1

3.6.1 Experimental Expansion of the Architecture

After evaluating a set of hyperparameters, the assumed optimum is summarized in Table 3.16. Using this set of hyperparameters, the MAE is reduced by $6,59$ compared to the baseline set by the random forest regression. The MSE is reduced by $517,74$ and the score is reduced by $19585,92$. The R^2 increased by $0,35$. The results of the champions of the different experiments are summarized in Table 3.20

Stacked LSTM

As stated in Section 2, deep learning approaches were able to outperform LSTMs with one hidden layer. Therefore, an evaluation of the optimized layer stacked to up to 10 times is conducted. The best performance is reached by 8 stacked layers. The stacking reduces the score to $566,92$, the MAE by $0,32$ and the MSE by $4,7$, as can be seen in Table 3.20.
Zheng et al. propose a architecture with 4 layers of LSTMs [ZRFG17]. The proposed size of the layers is $32, 64, 16, 16$ neurons. The comparison to the layers of 155 dimensions can be seen in Table 3.17. This Table also shows that the by Zheng et al. proposed

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Parameter	value
Overlapping of the time while creating input data	30
Number of cycles used for a feature vector	65
Number of epochs	40
Batch size	60
Optimization algorithm	adamax
Loss function	mean_absolute_error
Number of neurons in the hidden layer	155

Table 3.16: Summary of the hyperparameters, which are assumed to build an optimum for the scenario evaluated in Section 3.5.

architecture performs slightly better than 4 hidden layers with 155 neurons. But the best performance is achieved by 8 hidden layers with 155 dimensions each.

Description	score	MAE	MSE	R^2
8 hidden layers of the size 155	**566,92**	**12,12**	**269,10**	**0,85**
4 hidden layers of the size 155	1161,83	13,64	337,45	0,83
4 hidden layers as proposed in [ZRFG17]	989,78	13,20	316,56	0,83

Table 3.17: Comparison of the performance of stacked LSTM layers.

Autoencoder for Noise Reduction

Creators of the C-MAPSS data set state that the data set is enriched with various noise distriptions as described in Section 3.2. As stated in Section 2, the use of autoencoders is an effective method to overcome inaccurate predictions caused by noisy data. Therefore, different architectures of autencoders for noise reduction are evaluated for the data set.

In a first step, an autoencoder with one hidden layer of different dimensions is evaluated. The data contains 24 input dimensions. The output is reduced by the encoder to $3, 6, 12$ and 18 dimensions. The performance is evaluated after 300 epochs of training, by evaluating the predictions of the random forest regression algorithm, which is also used to set the baseline, as described in Section 3.3. In this experiment, the increasing of dimensions implies better performance, as can be seen in Table 3.18.
Lu et al. propose a deep autoencoder for noise reduction in the context of machine health prediction [LWQM17]. They use 4 hidden layers, where the number of neurons is divided by two for each subsequent layer. For this approach a slightly modified version of the autoencoder is used, since the performance with the proposed architecture with 4 hidden layers with the dimensions $24, 12, 6, 3$ performs badly, as can be seen in Table 3.18. An architecture with 4 hidden layers with the dimensions $24, 18, 12, 6$ outperforms the single hidden layer architecture with 18 dimensions in the MAE and the R^2 metric, but perform worse by the score and the MSE.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Output dimensions	**score**	**MAE**	**MSE**	**R^2**
3	3083218,90	39,58	2597,21	-0,50
6	61874,74	28,37	1376,42	0,20
12	27650,15	28,85	1369,09	0,21
18	**24131,15**	27,08	1250,71	0,28
(24, 12, 6, 3)	822165,90	37,30	2293,40	-0,33
(24, 18, 12, 8)	31153,20	**24,65**	**1109,51**	**0,36**
Baseline (no noise reduction)	**22152,25**	**23,73**	**995,01**	**0,42**

Table 3.18: Comparison of different architectures of autoencoders with random forrest prediction.

The autoencoder with 4 hidden layers of the dimensions $24, 18, 12, 8$ used as noise reduction for the input of a LSTM with 1 hidden layer and optimized parameters, produces the following results:

Output dimensions	**score**	**MAE**	**MSE**	**R^2**
(24, 18, 12, 8)	34088,69	24,50	1086,04	0,37

Table 3.19: Output of the 4 dimensional autoencoder used as noise reduction for the LSTM with 1 hidden layer.

Therefore, the performance of all 4 metrics is lower than without noise reduction. Using the output of the autoencoder as input for the deep LSTM consisting of 8 hidden layers, the performance improves, but is still outperformed by the same LSTM without noise reduction, as can be seen in Table 3.20. When it comes to the combination of the denoising network and the regression network, the overall best performance is achieved by the LSTM with 8 stacked hidden layers, which receives an 18 dimensional input from an autoencoder with only 1 hidden layer. This architecture produces predictions with a score of $468, 73$, a MAE of $11, 88$, a MSE of $257, 98$ and R^2 of $0, 84$, as can also be seen in table 3.20.

By reducing the dimensions to 16, the results degrade rapidely. The score increases to 1170, the MAE and MSE to $20, 03$ and $528, 25$. The R^2 reaches only $0, 42$.

3.6.2 Degradation with sparse data

In order to measure the degradation of the method with sparse data, the proposed architecture is evaluated with

- different sparse numbers of features
- different sparse numbers of devices

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Description	Score	MAE	MSE	R^2
Baseline: Random Forest	22152,25	23,73	995,01	0,42
LSTM with 1 hidden layer	672,54	12,42	273,87	0,85
Stacked LSTM: 8 hidden layers	566,92	12,12	269,10	**0,85**
Stacked LSTM: 4 layers as proposed in [ZRFG17]	989,78	13,20	316,56	0,83
Stacked LSTM with 1 layer autoencoder of 18 dimensions	**468,73**	**11,88**	**257,98**	0,84
LSTM with deep autoencoder	34088,69	24,50	1086,04	0,37
Stacked LSTM with deep autoencoder	290389,70	17,59	681,30	0,73

Table 3.20: Overview of the champions of the different network architectures. It is ordered by the complexity of the layers from simple to most complex.

which are randomly selected.

Table 3.21 shows the results of the stacked LSTM with falling number of features available for training and prediction. The experiments are conducted with 100 devices and the features are picked randomly. Noticeable is a spike when using 6 features.

Number of features	**Score**	**MAE**	**MSE**	**R^2**
24 features	468,73	11,88	257,98	0,84
18 features	4096,41	20,83	731,10	0,04
12 features	206035,63	50,23	3106,17	-1,47
6 features	497967,67	58,537	4017,53	-4,23
3 features	156674,88	38,87	2448,68	-28,32

Table 3.21: Degradation of the results of the stacked LSTM with 1 layer autoencoder of 18 dimensions with reduced number of features for training.

Table 3.22 shows the results of the experiments with reduced number of devices. The experiments are conducted in a range between 100 to 1 devices. The selection of the devices is done randomly. The score is the sum of the penalties of the prediction for a device. With decreasing number of devices the score is not comparable as such. To make it comparable, the aliquot score for 100 devices is determined by multiplying it with the factor x as follows: $x = \frac{100}{n}$, where n is the number of devices. The degradation of the results can clearly be seen in each of the metrics. Also the combination of 1 device and 10 features is evaluated, since those are the numbers of devices and features for the second case study.

3.7 Discussion and Potential for Future Work

Since the random forest regression is described as basic technique with high performance by Khan [KY18], it is used as baseline against which the proposed method is evaluated. After evaluating five hyperparameters as described in Section 3.3, the random forest reaches a MAE of $23,73$, a MSE of $995,01$, a R^2 of $0,42$ and a Score of $22152,25$. The

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Number of devices	**Score**	**Score aliquot to 100 devices**	**MAE**	**MSE**	**R^2**
100 devices	468,73	468,73	11,88	257,98	0,84
75 devices	610,43	813,91	13,66	288,30	0,82
50 devices	920,72	1841,44	16,15	466,39	0,69
25 devices	643,18	2572,72	14,94	371,73	0,67
12 devices	2840,78	23673,17	25,92	888,79	-0,26
6 devices	134092,16	2234869,33	25,92	3223,97	-1,02
1 device	870552,64	87055264,00	44,61	3665,01	-25,35
1 device and 10 features	2325376,78	232537678,20	55,78	4681,59	-20,43

Table 3.22: Degradation of the results of the stacked LSTM with 1 layer autoencoder of 18 dimensions with reduced number of devices for training.

overall performance comparison is shown in Figure 3.5. Using a one layered LSTM with a combination of default parameters and parameters estimated by educated best guess, which can be seen in Table 3.23, the results improve already: The score is reduced by $529,83$, the MAE is reduced by $103,9$, the MSE drop by $662,04$ and the R^2 is increased by $0,41$, as can be seen in Figure 3.5. These results emphasize the capabilities of the LSTM for the use case of fault detection by using time series data of machine sensors.

Parameter	**value**
Overlapping of the time while creating input data	no overlapping
Number of epochs	50
Batch size	72
Optimization algorithm	adam
Loss function	mae
Number of neurons in the hidden layer	50
Number of cycles for the feature vector	30

Table 3.23: Parameters used for the first trial set by an educated best guess, as starting point for the parameter evaluation.

The improvement gets more explicit after the hyperparameter evaluation as shown in Figure 3.5. By using the optimized hyperparameters which can be seen in Table 3.16, the score can be reduced to 672, which outperforms the LSTM before the optimization step by $1013,42$ and the random forest by $1543,25$. The MAE is reduced by $9,2$ compared to the not optimized LSTM and by $113,1$ compared to the random forest. Further improvement is provoked by introducing more LSTM layers. By stacking 8 layers of LSTMs with the same dimensions and hyperparameters, the score can be reduced by $105,62$. The MAE, MSE and R^2 only improve slightly by $0,3$, $4,77$ and $0,02$. A comparison between predicted remaining useful life (RUL) and the true RUL can be seen in Figure 3.7 and Figure 3.8 before and after noise reduction by an autoencoder with 1 layer with 18 dimensions. The Figures show that the accuracy decreases with higher RUL. This implies that the degradation of the measurements is detected and interpreted better, the closer a

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Wien Bibliothek Your knowledge hub

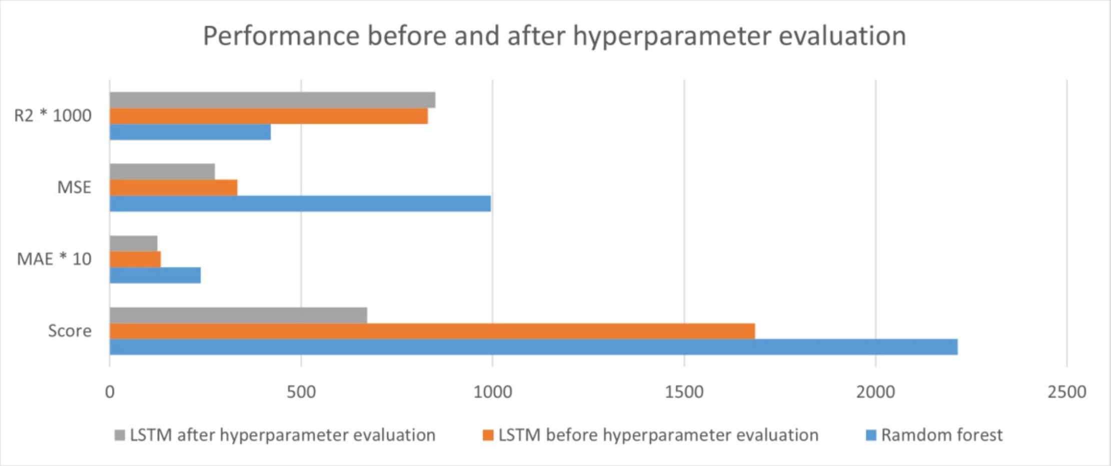


Figure 3.5: Performance of the LSTM with one hidden layer before- and after the hyperparameter evaluation. The R^2 and the MAE is scaled by a factor of 1000 and 10, respectively for better visibility of the graph.

failure comes. In figure 3.6, it can be seen that the score gets especially high for remaining cycles bigger than 50.
The Figures also show, that the predictions tend to predict the failures later than they occur. The penalty while determining the score is higher if the predicted RUL is higher than the true RUL. Therefore, those predictions especially influence the score negatively. Looking at the scores of the predictions per device, it is noticeable that the 10 predictions with the highest score represent $69,49\%$ of the total score. The worst 5 scores still represent $56,9\%$ of the total score, as can be seen in Table 3.24. The table again shows that out of those 10 predictions only 3 predict the failure before it happens.

Device	Prediced RUL	True RUl	Score
67	121,37	77	83,54
22	154,96	111	80,13
93	32,09	85	57,57
14	147,60	107	56,98
79	101,16	63	44,44
9	141,91	111	20,99
78	136,02	107	17,21
15	110,08	83	13,99
95	97,18	128	9,7
86	112,43	89	9,41

Table 3.24: 10 predictions with the highest score of the stacked LSTM before noise reduction.

On the basis of the work of Lu et al. [LWQM17], autoencoders are introduced for denois-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

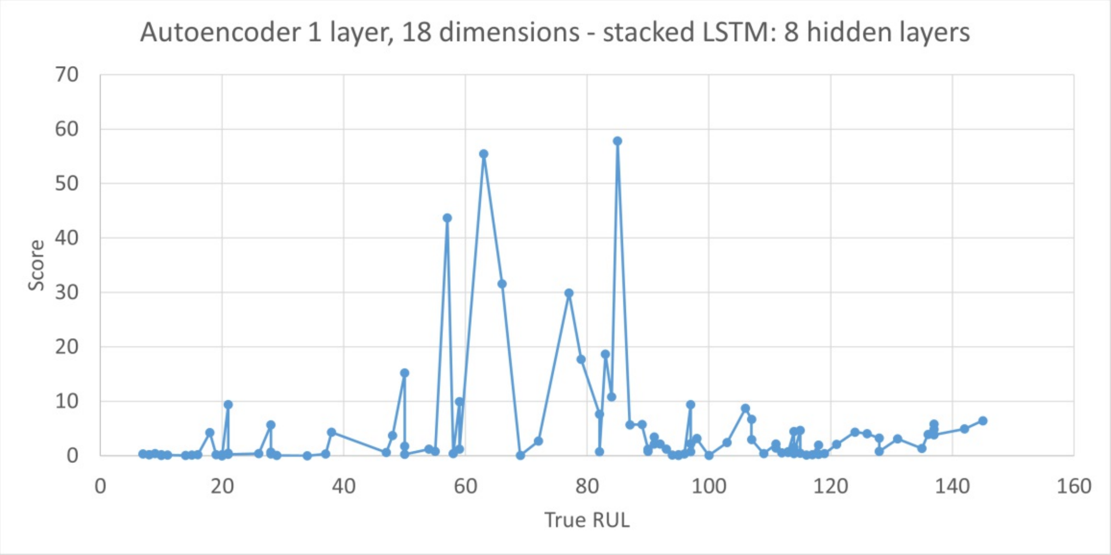


Figure 3.6: Score of the stacked LSTM after noise reduction for the remaining cycles of the devices.

ing and feature selection before conducting the regression step. Originally, 4 decoding layers are proposed, where every layer reduces the previous by half. The proposed setup increases the score by 422, 86. Also the MAE, MSE and R^2 degrade. The best performance is reached by a one layered autoencoder with 18 dimensions. The idea of denoising autoencoders is to reduce noise by filtering data, which consists of high frequencies by reducing the dimensionality through layers low dimensionality. Underlying is the assumption that noise consists of high frequency data. The layer with the lowest dimensionality is the so called *bottle neck*. The trade off thereby is that useful information gets lost too. Looking at the random forest regression, the noise reduction with autoencoders are not able to improve the results, as can be seen In Table 3.18. Using the stacked LSTM, the results can be improved using an autoencoder with 18 output dimensions as stated above. Therefore, the experiments suggest that a small bottle neck of 3, 6 and 12 dimensions dismisses too much useful information.

For future work, there are different approaches for denoising which can be investigated. One would be to not only filter high frequency data through the bottle neck of an autoencoder, but to emphasize the robustness against noise by distortion of the input data at the training phase of the autoencoder. This could for example be done by a small random offset of the data or some random distribution functions.

The comparison between predicted RUL and true RUL can be seen in Figure 3.7. The scatter plot is tilted even more to the right, which implies that the predicted RUL tend to lie after the true failure. But according to the score that can be compensated by higher accuracy. The by 0, 24 smaller MAE proves that assumption. Table 3.25 shows that out of the 10 predictions with the highest score, only one predicts the RUL before the true failure. The higher score penalty for late predictions was introduced by [SGSE08] to

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

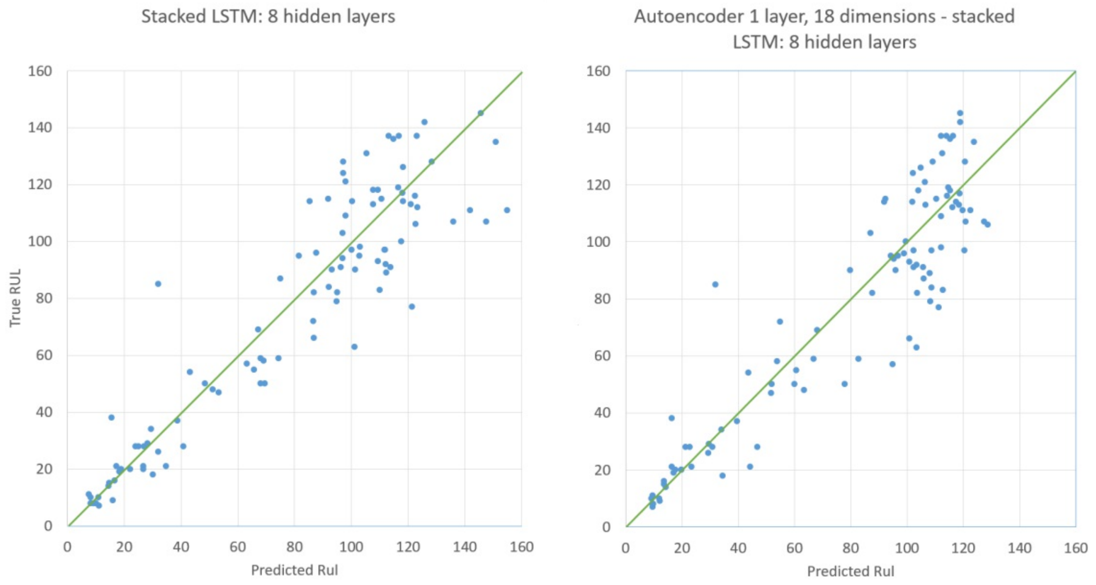


Figure 3.7: Predicted RUL on x axis versus the true RUL on the y achsis of the stacked LSTM before and after noise reduction by an autoencoder with 1 hidden layer of 18 dimensions. The green line symbolizes the most accurate prediction.

reconstruct the business need for of predictions to be rather early than too late in order for technicians to be able to react in time. For the use case of predictive maintenance that outcome means one possible improvement of this method is to optimize the model such that the predictions occur rather earlier than too late. An easy way is a linear approach. The prediction can be shifted to an earlier cycle in a way such as prediction p is $p - i$, where i is optimized by minimizing the score while training.

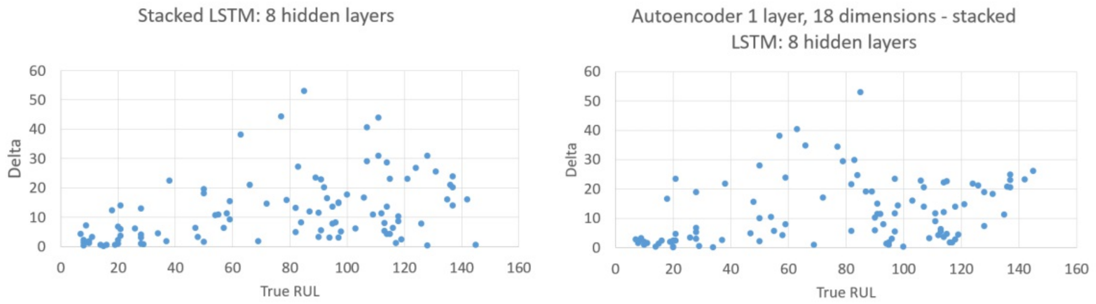


Figure 3.8: Delta between the true RUL and the predicted RUL of the stacked LSTM plotted along the axis of the true RUL before and after noise reduction by an autoencoder with 1 hidden layer of 18 dimensions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Device	Prediced RUL	True RUl	Score
93	32,03	85	57,82
79	103,33	63	55,44
21	95,00	57	43,70
27	100,84	66	31,59
67	111,30	77	29,88
15	112,77	83	18,63
50	108,29	79	17,71
72	77,86	50	15,02
16	108,69	84	10,80
98	82,90	59	9,91

Table 3.25: Worst 10 predictions of the stacked LSTM after noise reduction by an autoencoder with 1 hidden layer of 18 dimensions.

In comparison to other methods, the proposed method shows comparable results. Table 3.26 shows the score, MSE and MAE of methods, which are evaluated best on the test date set of the PHM'08 Challenge, published by Ramasso et al. [RS14]. Additionally chosen methods published 2020 are compared below. With a score of $468,73$, the proposed method would rank first in this table, the MSE of $257,98$ would be third and the MAE of $11,88$ reach the fourth place.

Rank of Ramasso et al. [RS14]	**Score**	**MSE**	**MAE**
1	512,12	152,71	8,67
2	740,31	224,79	10,77
3	873,36	265,62	11,47
4	1218,43	269,68	11,87
5	1218,76	331,30	13,81
6	1232,27	334,52	14,14
7	1568,98	394,46	15,37
8	1645,77	330,02	13,47
9	1816,60	359,97	13,82
10	1839,06	377,01	14,31
Method	**Score**	**MSE**	**MAE**
DOS-ELM [BMK$^+$20]	189,77	unknown	unknown
Multi-scale deep convolutional neural network [LZZZ20]	196,22	unknown	unknown
Proposed method	**468,73**	**257,98**	**11,88**

Table 3.26: Selection of metrics for the algorithms ranked best on the PHM'08 Challenge data based on the test set published by Ramasso et al. [RS14] and other publications.

To simulate a scenario that is comparable to the second case study, the number of available features and devices are reduced. Experimental results can be seen in table 3.21 and 3.22. Figure 3.10 shows the degradation in a graphical way. The score, MAE and MSE increase with decreasing number of features available for the algorithm up to 6

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

features. With 3 features the errors decrease, but the results of individual experiments scatter exponentially depending on which sensors are chosen randomly in the experiment. Nevertheless, the score increases roughly by a factor of 440 between 24 and 12 available features. Also the MAE doubles and the MSE increases by a factor of roughly 12. The deviations of the predictions from the true failures can be seen in figure 3.9. In comparison to the usage of all available data which can be found in figure 3.7, the predictions using 12 features scatter already around the true RUL. Predictions using 3 features are heavily skewed to the right of the diagram. That means the predictions assume the failures of the devices later than they actually occur. This happens especially for failures which appear in less than 60 cycles. For the usage of predictive maintenance that is not optimal. The goal is to be able to provide a warning before the device will fail. The importance of a correct prediction increases the closer the failure comes, as the urgency to act increases. Predicting the failure after the actual failure implies the false assumption of safety.

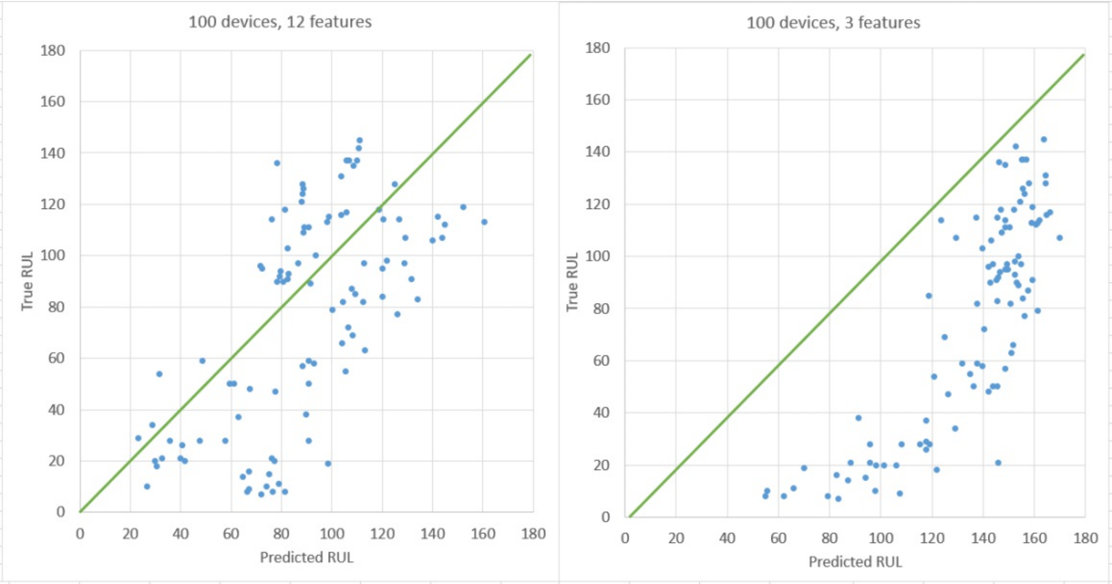


Figure 3.9: Predicted RUL on x axis versus the true RUL on the y achsis of the algorithm with 12 features and 3 features. The green line symbolizes the most accurate prediction.

Looking at the degradation of the predictions with decreasing number of devices, the score increases with decreasing number of devices available. Conversely, the learning curve increases with increasing number of devices. The score increases by a factor of 185726 by decreasing the number of devices from 100 to 1 device. The MAE for that example increases by 32, 73 and the MAE increases by 3407, 03. Figure 3.11 shows the degrading predictions for the different number of devices used for the experiment. A setup similar to the data available for the second case study is the combination of 1 device and 10 features. The errors and the score of the experiment with the combination of reduced data exceed those in the degradation experiments for either a decreasing number

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

of features or a decreasing number of devices: The score is by 145482414, 20 higher than the score with 1 device and 24 features and by 232380999, 32 higher than the score with 100 devices and 3 features.

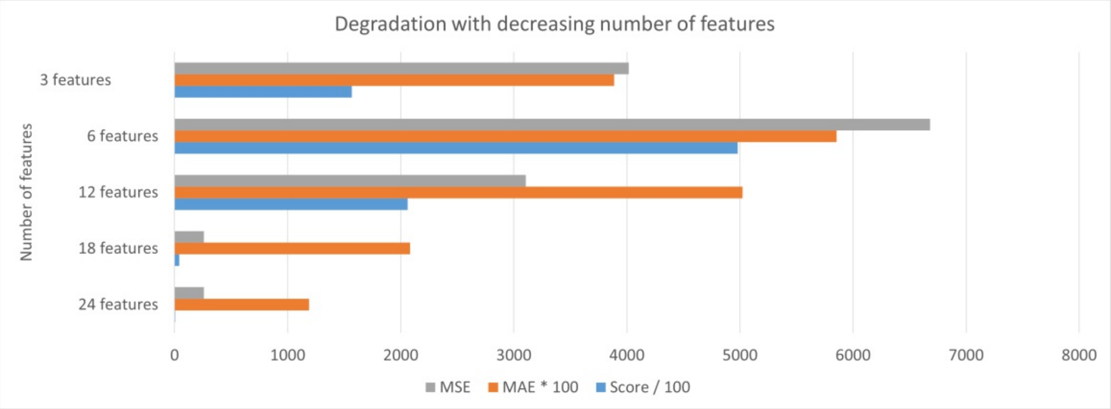


Figure 3.10: Degradation of the results with decreasing number of available features. To improve the visualization, the score is divided by 100 and the MAE is multiplied by 100.

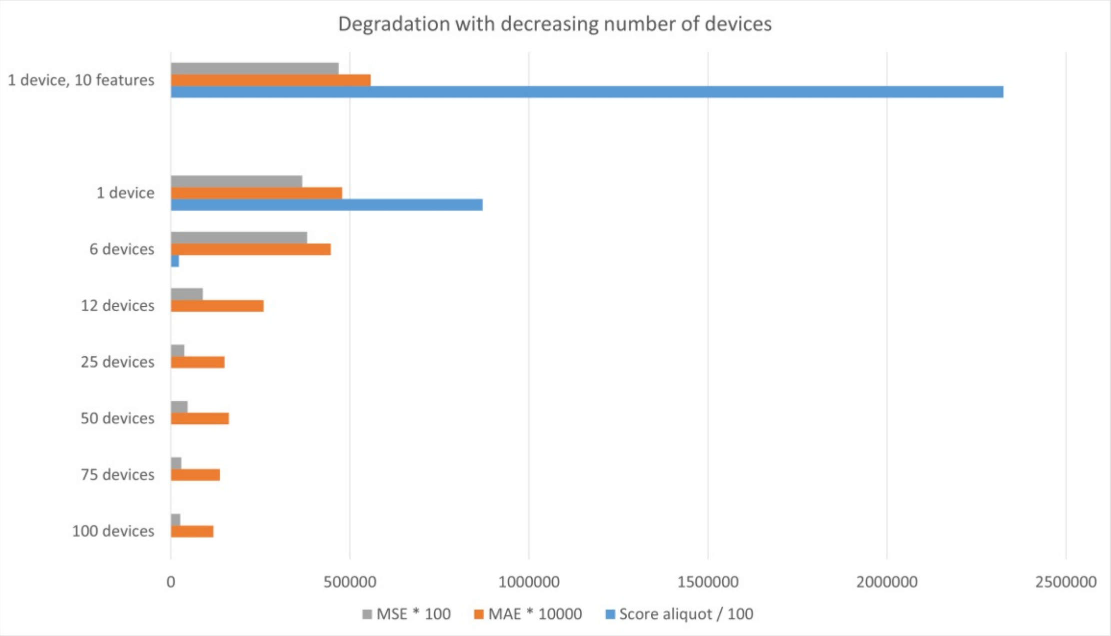


Figure 3.11: Degradation of the results with decreasing number of available devices. To improve the visualization, the score is divided by 100, the MAE is multiplied by 10000 and the MSE is multiplied by 100.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 4

Study of a Dataset not Optimized for Predictive Maintenance

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In contrast to the widely used scientific C-MAPSS turbo engine data set analyzed in Chapter 3, this chapter covers the application of predictive maintenance in respect of an application with a high level of uncertainty. This case study was conducted on data acquired from a food production line. Especially the absence of a known ground truth necessitates an explorative approach and therefore focuses on preprocessing steps within the CRISP-DM. Extensive involvement of domain experts was necessary, resulting in iterative exploratory analysis. The goal was to prepare the data for the method described in Chapter 3 to be able to evaluate the compatibility.

4.1 Methodology - Iterative Application of CRISP-DM

As stated in Capter 2, the chosen method for this thesis is CRISP-DM. This standard describes an iterative approach with several sub- iterations. The first 6 iterations were conducted between the first 2 steps of the CRESP-DM, *Business Understanding* and *Data Understanding*. One iteration had a duration of roughly two weeks. At the end of the iteration the results were presented and discussed in cooperation of a board of stakeholders that are concerned by the presented topics. At this meeting, the business needs were verified against the new findings and the next analysis steps were identified to accomplish the business goals. The following analysis iterations were executed:

- First iteration
    - Identification the key stakeholders
    - Initial understanding of the business need

  - Data aggregation and import
  - Initial understanding of the data

- Second iteration
  - Statistical analysis of all features
  - Analysis of the sampling rates
  - Analysis of the data types
  - Classification of the structure (categorical, numerical)

- Third iteration
  - Analysis of the semantic meaning of the features
  - Visual inspections

- Fourth iteration
  - Analysis of the machine states

- Fifth iteration
  - Analysis of alarms and errors

- Sixth iteration
  - Assembling of the groundtruth

The next steps according to the CRISP-DM are *Data Preparation* and *Modeling*. The preparation of the data was subdivided to data cleansing, preprocessing and feature engineering. This steps were executed for 3 models, respectively:

- Regression of the RUL by random forest regression
- Regression of the RUL by the method discussed in the first case study
- Classification of incidents by random forest classification
- Classification of incidents by a multi layer perceptron

Each of the models was evaluated. The last step, *Deployment*, was not executed since the framing conditions were academic.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.2 Business Objectives

Originally, the partner developed a state of the art procedure for food production lines based on special hardware. This hardware was responsible for advantages on the market. To ensure advantages to competitors the partner began to collect basic measurements of the devices in order to monitor the devices and to be able to prove uptime in a first step. The next step shall be to train models on the base of the data that are able to predict machine outages. In the course of the analysis intermediate steps were identified while assessing the current situation of the data as described above. The main goal of the study is to verify to what extent a model can be trained on basis of the current data in order to predict a critical incident of the production line. In a best case scenario, an accurate prediction of the remaining live of a machine improves the efficiency by optimizing maintenance intervals, while reducing the down time. In a first step a warning 30 minutes before an incident happens would give technicians the chance to react and therefore reduce damage of the machine and down time. Translated to the data mining goals, that means the collected data should be prepared and evaluated with simple models for the capabilities of making predictions for the given time frame.

4.3 Data Understanding

After clarifying the business need, the next step was to iteratively refine the understanding of the data and verify the potential of the data with the initial goal.

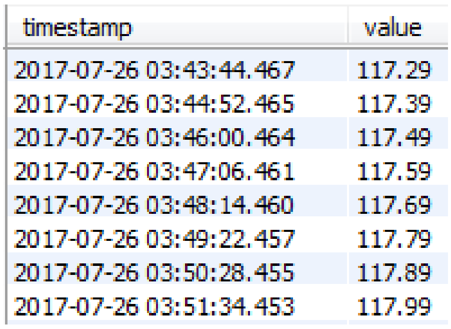
4.3.1 Initial Description of the Data Set

The data is provided within a SQL database dump, which contains sensor data, meta information and data for the operating system of the control panel of the machines. In total the dump of the device includes 764 tables. For each source of information within one device, the data is stored to a separate table in the format *timestamp, value*. These tables contain between 75 and 2212710 data points and lie in the time span from 2017/07/25 07:53:24 to 2018/04/05 01:46:49, covering a total duration of approximately 36 weeks. Entries in the column *value* are either numbers, strings or lists. The device contains 682 tables in the format shown in Figure 4.1, which leaves 82 tables of additional data, which varies strongly in size of the tables and their content. In contrast to the scientific data set used for the first case 3, the data in the second case present a high level of uncertainty, due to the following issues:

- The sampling rate of the data is unknown and differs between measurements
- The content of the data is not comprehensible directly
    - Unites are not always persisted within the data
    - It is unclear if the data is numerical, categorical or ordinal

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

  - Therefore, the proper interpolation between data points cannot be determined
  - Data cannot intuitively be distinguished in sensor data and meta data
- The proper functionality of the sensors cannot be assumed automatically
- It is not clear whether the data contains incidents of failure of a system or a sub system
- The environment of the devices differs and may contain interaction with humans or other machines
- The possibility of states which the device can be in is unclear and outstrips a simple on, off or failure model



timestamp	value
2017-07-26 03:43:44.467	117.29
2017-07-26 03:44:52.465	117.39
2017-07-26 03:46:00.464	117.49
2017-07-26 03:47:06.461	117.59
2017-07-26 03:48:14.460	117.69
2017-07-26 03:49:22.457	117.79
2017-07-26 03:50:28.455	117.89
2017-07-26 03:51:34.453	117.99

Figure 4.1: Extract of one of the 682 tables in the format *timestamp, value*.

4.3.2 Exploratory Data Analysis

To prepare the preprocessing steps sound understanding of the data is necessary. Therefore exploratory analysis with aspects of statistics, context and the structure of the data is conducted and visualized for reporting and discussions with the domain experts. The analysis is sharpened iteratively by intensive involvement of the experts. After the initial aggregation of the data, analysis steps are executed and reports are prepared, which initiate discussions that lead to decisions for further analysis. For the scope of this thesis, six formal iterations are traversed within the exploratory analysis. Additionally, a basic amount of collaboration is pursued between the iterations. Every iteration the focus topic shifted towards the aggregation of the ground truth.

4.3.3 Statistical Analysis

To localize the value ranges of the provided data and identify groups of attributes within the same range, for every table that contains numerical data, the following statistical

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

measurements are computed:

- Number of entries in the table
- Median
- Mean
- Minimum value
- Maximum value
- Quantiles (0.25,0.75)
- Standard deviation σ
- First and last recording
- Mean sampling interval
- Median sampling interval
- Maximum interval between samplings
- Minimum interval between samplings

The sampling rate varies highly between the features. While partly the sampling rate is constant at least for periods of time, other parts have large gaps between all entries. Table 4.1 shows meta statistics of the time stamp analysis. The over all smallest interval is $0,21$ seconds, the smallest median of intervals is 2 seconds. Depending on the state the machine is in, the sampling rate of continuously recording sensors is 2 seconds, which is reflected by the smallest median of sampling intervals. Data which is not recorded in constant time intervals is only written within the change of a value. 659 of the 682 tables contain only 75 data points. Thus, the intervals between the recordings are large, and due to the large fraction in contrast to the 23 remaining features with more data points, the median and mean is also influenced strongly. Further analysis steps show, that the sparse tables contain operational settings and lists of constant values and can be filtered, as described in Section 4.4.1.

4.3.4 Context Analysis of the Data

In a next step, basic semantics and the context of the data is analyzed. Therefore, the data type of the values is selected, as well as whether there are subsequent equal entries within a table. Subsequent values are, together with the sampling rate and the deviation of samples from the sampling rate, an indicator whether only changes are persisted when they happen or data is persisted periodically, independently of their status. Furthermore, there is a mapping between the tables stored to the database and data which is displayed

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

	smallest overall value (s)	**biggest overall value (s)**	**overall median (s)**
min(sampling interval)	$0,21$	$163,53$	$162,81$
max(sampling interval)	$3,62e+6$	$3,62e+6$	$3,62e+6$
mean(sampling interval)	$9,91$	$2.93e+5$	$2.93e+5$
median(sampling interval)	2	$8,01e+4$	$8,01e+4$

Table 4.1: Illustration of the boundary values of the sampling rates. The different rows show statistics over the data tables: minimum, maximum, mean and median. Columns summarize the statistics over all data tables: The smallest minimum of all tables, the biggest minimum of all tables and the average minimum of all tables, and so on. It can be seen that the interval between two samplings lies between $0,21$ seconds and $3,62e+6$ seconds. The median of the means is $2,93e+11$ and the median of medians is $8,01e+4$.

in a graphical user interface. This mapping is used to extract context information about the data. This information is used to distinguish between numerical and categorical data, but also to generate knowledge to improve future steps like the selection of parameters and the extraction of a ground truth for supervised machine learning and evaluation. The tables can be categorized as follows:

- Array of values
  - List of numbers
  - List of boolean values
- Numerical values with a partially constant sampling rate include temperature, different speed measurements and power consumption
- Numerical values which are stored if a change occurs consist of various tables used for the machine state tracking
- Categorical values, such as operational settings

Another possible classification is the division of the data to what they describe. Based on the names of the tables and their content, the following categories can be identified:

- measurements
- alarms
- operational settings
- state tracking

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

4.3.5 Visual Inspections

For deeper understanding of what happens within the data, visualizations are generated and discussed in connection with the statistical analysis and the contextual analysis iteratively with experts of the production line. The objective is to identify sensor data and descriptive data which reflects the representation of the machine health. This is important to extract relevant data, but also as an entry point for the reverse engineering of a ground truth. The reporting and discussions with the machine experts benefit from the visible breakdown of the data too.

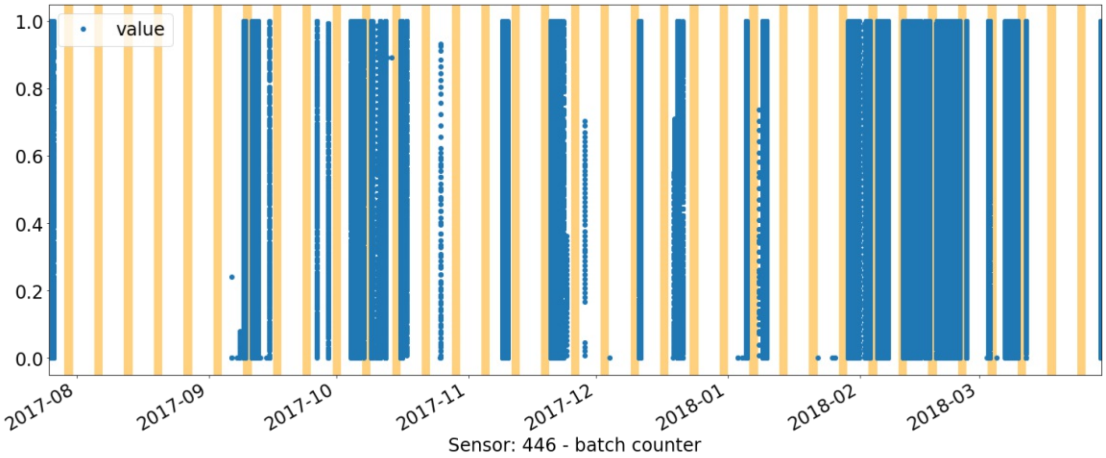


Figure 4.2: Batch counter of the device over the whole time period. The counter of a fixed number of pieces in the production is normalized to obscure the actual number of pieces that is produced. The orange bars mark weekends.

Analysis of the intervals between measurements shows gaps, which implicate the assumption that the device didn't produce to its full capacity in the observed time period. In Figure 4.2, a representative pattern can be recognized by mapping data of a batch counter to a central European calendar. Values are plotted blue, the orange bars mark weekends. Besides gaps of 5 weeks during August and Septemper 2017 and approximately 1 week several times, it can be seen that the device was shut down during weekends. In this figure a counter is plotted which counts products from 1 up to a batch size of n before it is reset. The rate with which this counter is transmitted to the data set is approximately 2 seconds. As long as the production street is running, this sensor writes saw-tooth textured data to the data base, as shown in Figure 4.3. It can be seen that the batch counter provides data for approximately 50% of the time. Also the biggest gap between measurements of all sensors can be seen in the beginning of the recorded data, which is $3,62e+6$, or approximately about 6 weeks.

Figure 4.4 shows 10 remaining features for the machine health after the dimensionality reduction described in Section 4.4.1. Since the data has no continuous sampling rate,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

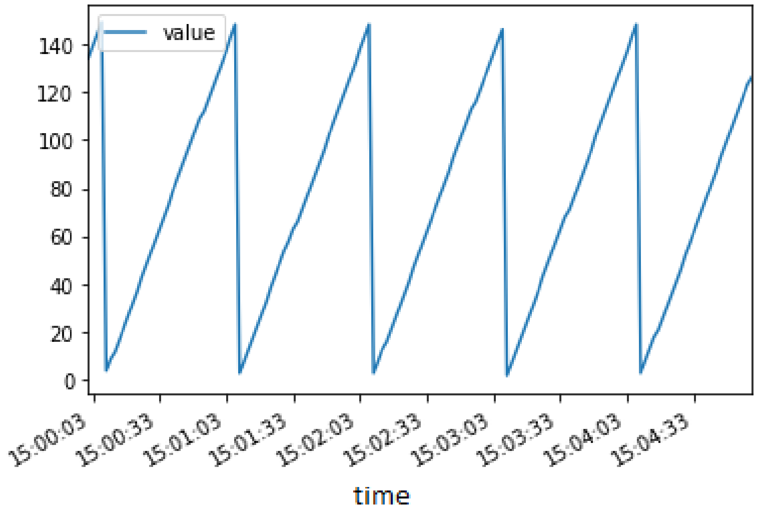


Figure 4.3: Saw-tooth textured measurement of the batch counter.

values between the samples are interpolated. Again, the orange blocks mark the weekends. This figure was the primary basis of discussions with domain experts. Correlation between sensor 445, 446, 449, 450, 456 and 774 can be seen directly. Sensor 186 accumulates over time, but is reset several times, which could indicate a reset of the device. The sensor 447 spikes only at certain points, which could indicate events where problems occur. Also indicative for failure events is the sensor 456, according to the experts. This sensor measures the temperature within the heating unit. Two different scenarios of decreasing temperatures can be found: slow and abrupt. In a consulting session with experts, the two patterns are interpreted as follows. Normally when the device is shut down, the heating unit stops and the production line runs until the temperature falls below a critical limit. Therefore, a slow decreasing of the temperature, especially in combination of a gap in sensor 446, which shows the batch counter discussed in the Figure 4.2, is the standard scenario for shutting down the device. However, an abrupt dropping of the temperature shows that the heating unit was opened. Reason for that can be a maintenance action. But the temperature alone is not sufficient to identify an incident, because the unit could also be opened due to pollution or for other reasons.

4.3.6 Reverse Engineering of the Ground Truth

To enable supervised machine learning methods, as well as for sufficient evaluation, a ground truth is needed. Since no ground truth is provided for this case study it has to be created. The approach taken in this thesis is based on reverse engineering the independent variable from sensor data by applying the iterative approach as described in the previous section. Therefore, the machine state is analyzed as well as the occurrences of alarms and errors within the device. With the obtained information of these analysis steps the ground truth is reverse engineered.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

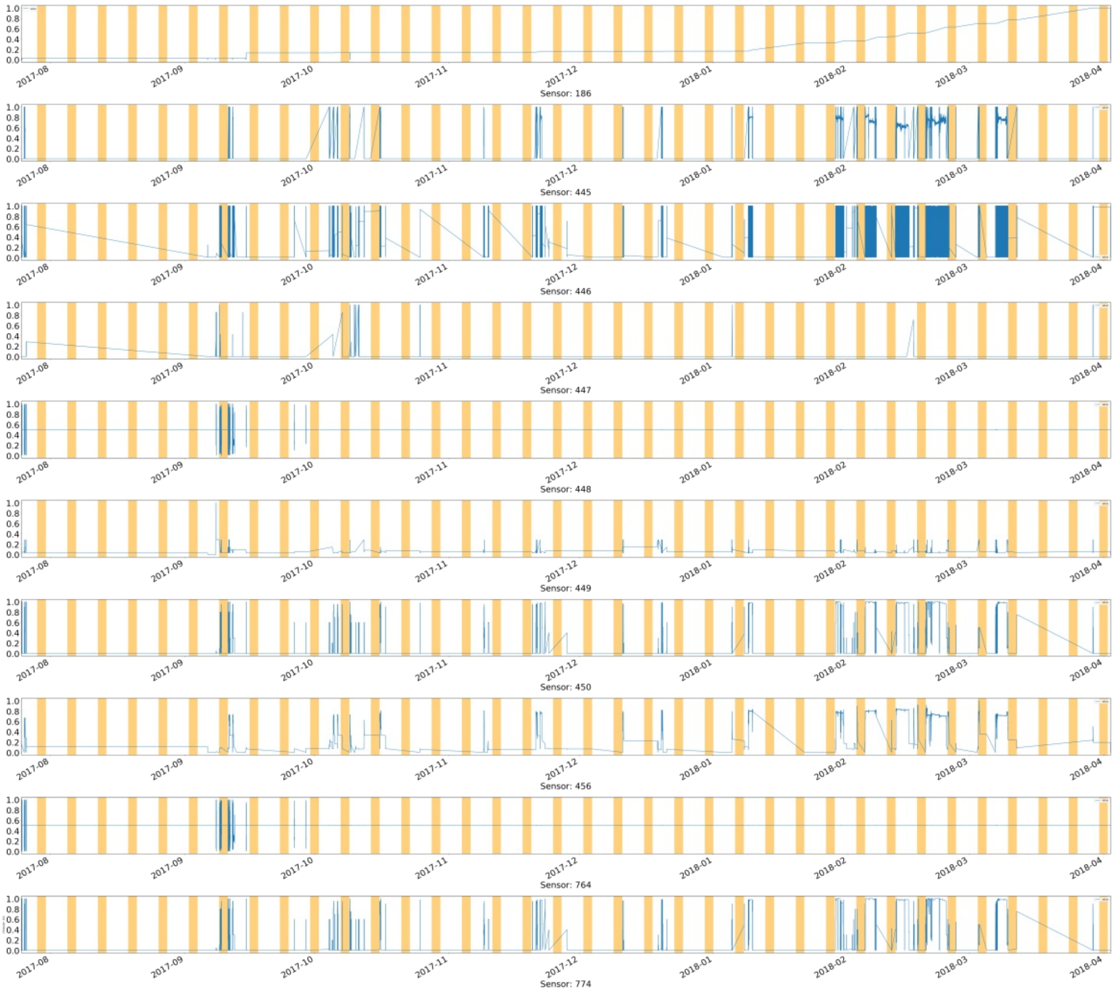


Figure 4.4: The 10 remaining features after conducting the dimensionality reduction. The data is normalized and the table names are obscured by random ids. The orange blocks mark weekends.

Machine State Analysis

When it comes to understanding the health condition of machines and especially for extracting incidents of system failure or the failure of a component, knowledge about the state the device is in makes a difference: While for example the dropping of the temperature of a heating element is normal while the machine is stopped, it can be evidence of a failure if the machine is running at full capacity. Therefore, special attention is directed to collecting detailed information of the state the device is in at all times.
Within this device the states are stored by the two different strategies: States that are needed internally for the controlling of the device and states which are used to compute statistical measurements by definitions of the the ISA-88 standards [SS08]. There are 11

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

internal states and 4 ISA-88 states.

Internal States
When an internal state changes, an identifying number is written to a table. As long as the state stays the same, the table has no entry. There is no proper documentation of the internal states available.

ISA-88 States
The ISA-88 states are triggered independently from the internal states. Four of the 17 orginally defined states are implemented by the device. Additionally to the undefined state '0', which is not part of the official ISA-88 standard, the following states can be found in the corresponding table:

- Clearing (ISA-88 - id: 1)
- Stopped (ISA-88 - id: 2)
- Starting (ISA-88 - id: 3)
- Holding (ISA-88 - id: 10)

Like the internal states, the changes of an ISA-88 state is stored to a table. But additionally, counters are stored which increase by 1 every minute, the machine stays in the current state. Therefore, the ISA-88 are easily accessible for further analysis.

The larger number of internal states indicates that the ISA-88 states generalize the internal states. But the triggers for the storage are independent. Therefore, the correlation between the two tables is investigated further. Figure 4.6 shows the mapping between the two state tables. Possible correlations between the states are visualized by sliding a time frame over the ISA-88 states and counting the number of internal states within the time frame. In Figure 4.6a, a time frame of 2 seconds is used, due to the median of the sampling interval of 2 seconds, which can be seen in Table 4.8. Objective of this experiment is to find data points in the tables that represent the same entry point of a state. To emphasize the relation the total occurrences of states in the whole records can be seen in Table 4.2. It can be seen that for states 1, 2 and 10 of the ISA-88 table a state of the internal table can be found, which relates to the time of the occurrences with a low number of outliers. The state 3 of the ISA-88 table is divided into state 5 and 14 of the internal states. Mentioned relations that contain over 100 occurrences of data base entries within a time frame of 2 seconds are summarized in Table 4.3.

Furthermore, the window size is increased by 2 and 4 seconds. Figure 4.6b and 4.6c show the results for the sliding window analysis for 4 and 6 seconds total, respectively. One thing that can be noticed is the increase within the time frame of 4 seconds of states

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

State ID of internal state	**total occurrences of the state**	**State ID of ISA-88 state**	**total occurrences of the state**
0	341	0	57
2	388	1	394
5	205	2	345
11	48	3	547
12	74	10	764
13	821		
14	630		
15	285		
16	285		
17	170		
18	3		

Table 4.2: Number of occurrences of the states.

ISA-88 state	**internal state**	**occurrences in ISA-88**	**occurrences in internal**	**within 2 seconds**	**within 4 seconds**	**within 6 seconds**
0	11	57	48	25	47	47
1	2	394	388	226	387	387
2	0	345	341	228	336	336
3	5	547	205	127	193	193
3	14	547	630	228	347	347
10	13	764	821	479	740	740

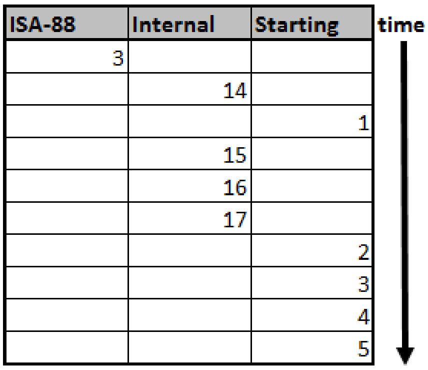
Table 4.3: Connection between ISA-88 and internal states, which occur together within a time frame of 2 seconds over 100 times.

whose relation is found by the time frame of 2 seconds. The number of correlations doesn't change in the experiments with the 6 second time frame. That indicates that the trigger mechanisms of the states have an offset of ≤ 4 seconds. Further increasing of the window size makes the result intransparent to an extent where the correlations cannot be interpreted. One pattern can be found repeatedly. It is shown in Figure 4.5. The ISA-88 state 3 marks the start of the production street. With the start, the device traverses the internal states 14, 15, 16 and 17, while the cummulative counter for the ISA-88 state called *starting* starts to increase by one every minute. Since the ISA-88 *execute* state is not implemented, it is assumed that the machine is executing after it reaches the internal state 17 within the starting sequence.

To investigate at which time the production street is in a producing state, the ISA-88 states are plotted over the production speed and the production temperature of the production line, which can be seen in Figure 4.7. Figure 4.7a shows the production speed over the whole time span of the recordings. Since the recording starts with the launching of the device, it is not producing at the beginning. Except for approximately the last third of the recording, the machine is either in the *stopped* state or the *clearing* state

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



ISA-88	Internal	Starting
3		
	14	
		1
	15	
	16	
	17	
		2
		3
		4
		5

Figure 4.5: The starting sequence which can be found repeatedly in the data.

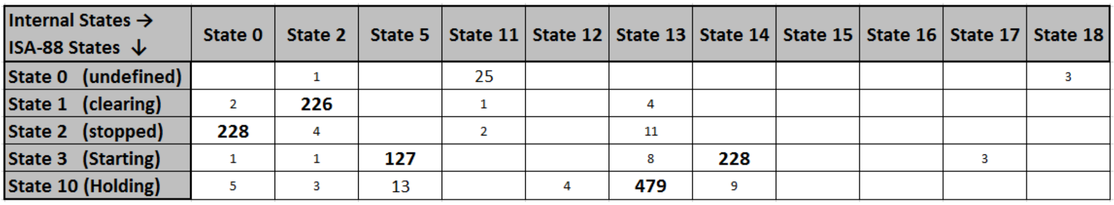
ISA-88 state	mean duration (sec)	median duration (sec)
Undefined (0)	1714,00	131
Clearing (ISA-88 - id: 1)	4793,51	63
Stopped (ISA-88 - id: 2)	5035,02	19
Starting (ISA-88 - id: 3)	2102,97	67
Holding (ISA-88 - id: 10)	677,01	35

Table 4.4: Mean and median duration of the ISA-88 states.

over large periods of time. While in the state *stopped*, the production speed equals zero, as expected. For *clearing* and *holding* no clear behavior can be seen, but within the state *starting*, the production speed returns approximately a constant speed, as can be seen more clearly in Figure 4.7b. Figure 4.7c shows that the production temperature of the heating unit is at an approximately constant value bigger than zero too, within the state *starting*. Therefore, the state *starting* is assumed as evidence that the machine is in a producing state.
The mean and median duration of the states can be seen in Table 4.4. While the median duration of the states is only between 19 and 131 seconds, the mean is between $677,01$ and $5035,02$ seconds. The difference reflects short changes which are partially part of the set up process of the production street, but also occur later in the life time of the machine for unknown reasons. The device stays within the state *starting* no longer than 67 seconds in 50% of the number of startings that are assumed. It continues running longer than 120 seconds 34% of the assumed startings and for 8% it produces longer than one hour.

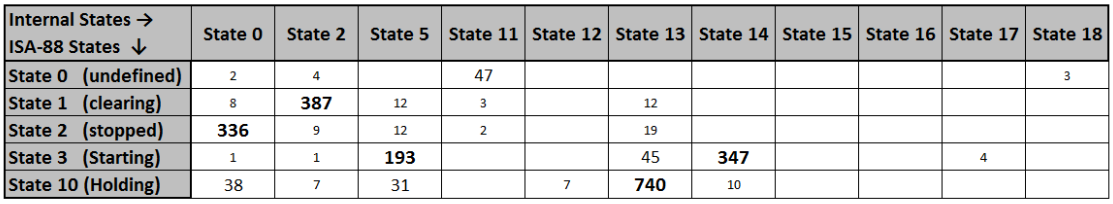
Analysis of Alarms and Errors

For the reverse engineering of the ground truth, important information is not only the knowledge of the machine state, but also alarm data, which is displayed at the graphical user interface. Besides informational announcements which are irrelevant for the machine



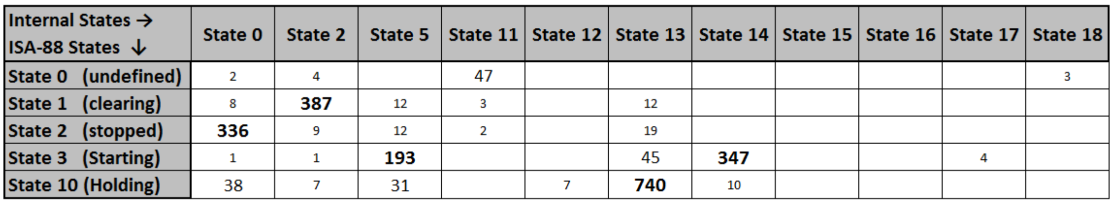
Internal States → ISA-88 States ↓	State 0	State 2	State 5	State 11	State 12	State 13	State 14	State 15	State 16	State 17	State 18
State 0 (undefined)		1		25							3
State 1 (clearing)	2	**226**		1		4					
State 2 (stopped)	**228**	4		2		11					
State 3 (Starting)	1	1	**127**			8	**228**			3	
State 10 (Holding)	5	3	13		4	**479**	9				

(a) Correlation between internally used states and the ISA-88 states. The correlation is measured by a sliding window method with a time frame of 2 seconds.



Internal States → ISA-88 States ↓	State 0	State 2	State 5	State 11	State 12	State 13	State 14	State 15	State 16	State 17	State 18
State 0 (undefined)	2	4		47							3
State 1 (clearing)	8	**387**	12	3		12					
State 2 (stopped)	**336**	9	12	2		19					
State 3 (Starting)	1	1	**193**			45	**347**			4	
State 10 (Holding)	38	7	31		7	**740**	10				

(b) Correlation between internally used states and the ISA-88 states. The time frame for this Table is expanded to 4 seconds.



Internal States → ISA-88 States ↓	State 0	State 2	State 5	State 11	State 12	State 13	State 14	State 15	State 16	State 17	State 18
State 0 (undefined)	2	4		47							3
State 1 (clearing)	8	**387**	12	3		12					
State 2 (stopped)	**336**	9	12	2		19					
State 3 (Starting)	1	1	**193**			45	**347**			4	
State 10 (Holding)	38	7	31		7	**740**	10				

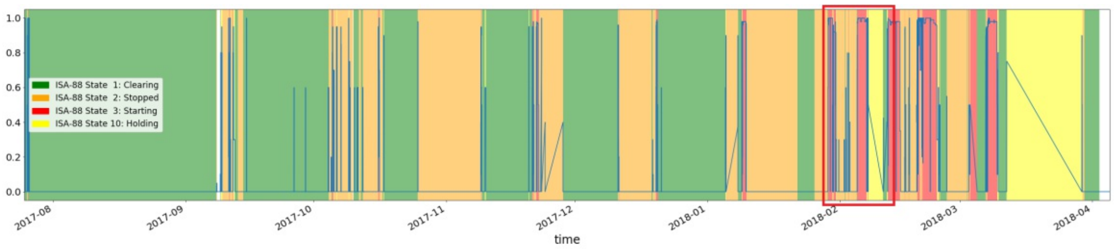
(c) Correlation between internally used states and the ISA-88 states. The time frame for this Table is expanded to 6 seconds.

Figure 4.6: Correlation between internally used states and the ISA-88 states. To measure the simultaneous occurrences of the states, a time frame is slid over the ISA-88 states and the occurrences of the internal states within the frame are counted. The number of occurrences is emphasized by the size of the font.

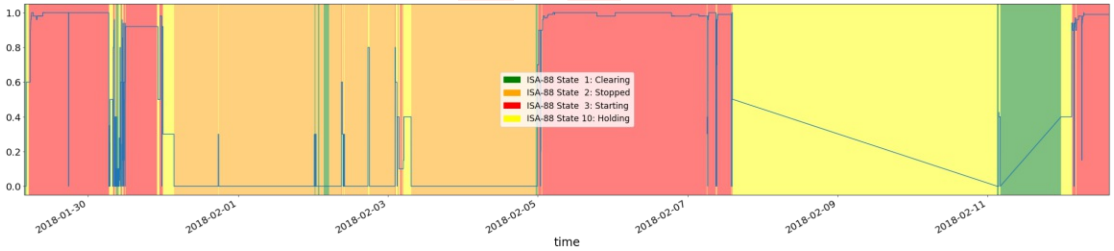
health, the system provides a number of alarms that indicate problems within components. In combination with the machine state, these messages and their time stamps build the basis for the reverse engineering of incidents which have happened in the past. In total, there are 1962 alarms and 295 errors. Figure 4.8 shows the occurrences of alarms and errors. For visualization purposes they are plotted over the production speed (Sensor 450) and the production temperature (Sensor 456).

Errors
Errors are either lists of Boolean values, which are a one hot indicator for a list of predefined errors, or a single value. Neither the lists, nor the values can be mapped to

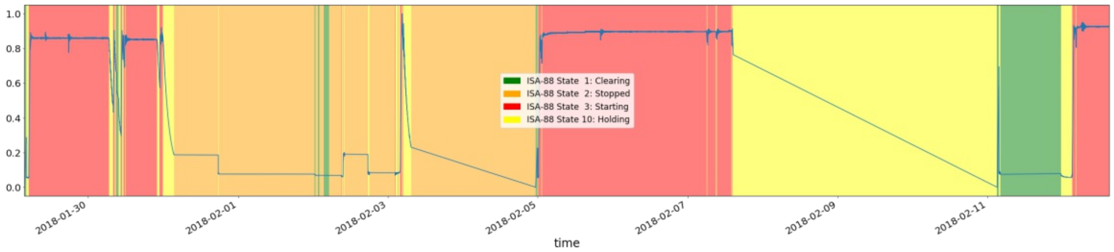
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



(a) The production speed over the full recordings. The red square marks the time span that is brought out in more detail in the plots 4.7b and 4.7c.



(b) A view to a shorter period of time. The correlation between the state *starting* in red and the production at constant speed can be seen more clearly.



(c) In this Figure the states are painted over the temperature of the heating unit. It can be seen that the temperature stays approximately the same within the state *starting*.

Figure 4.7: Production speed and the production temperature of the heating unit overlapped with the ISA-88 states the production street is in. The normalized production speed is plotted as a function and the states *clearing*, *stopped starting* and *holding* are painted as green, orange, red and yellow blocks.

known information in retrospect. Errors are designed only for live occurrence. Therefore, the only known information of the errors is the time of their occurrence. Figure 4.9a shows all errors over the full time span of the recordings. The errors are stored in 3 tables. Since the semantic meaning is not accessible, this is redundant information. Furthermore, the time stamps are equal in 2 of the tables, and a sub set in the third. As can be seen errors often occur at the change of state of the production street. This is, according to experts, not necessarily a failure in the machine. For example, certain errors are provoked by thresholds, such as the target temperature. When the device switches from a producing state to being stopped, the temperature decreases slowly. But the limit for the target temperature is set within the state. Therefore, an error is triggered, which doesn't

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

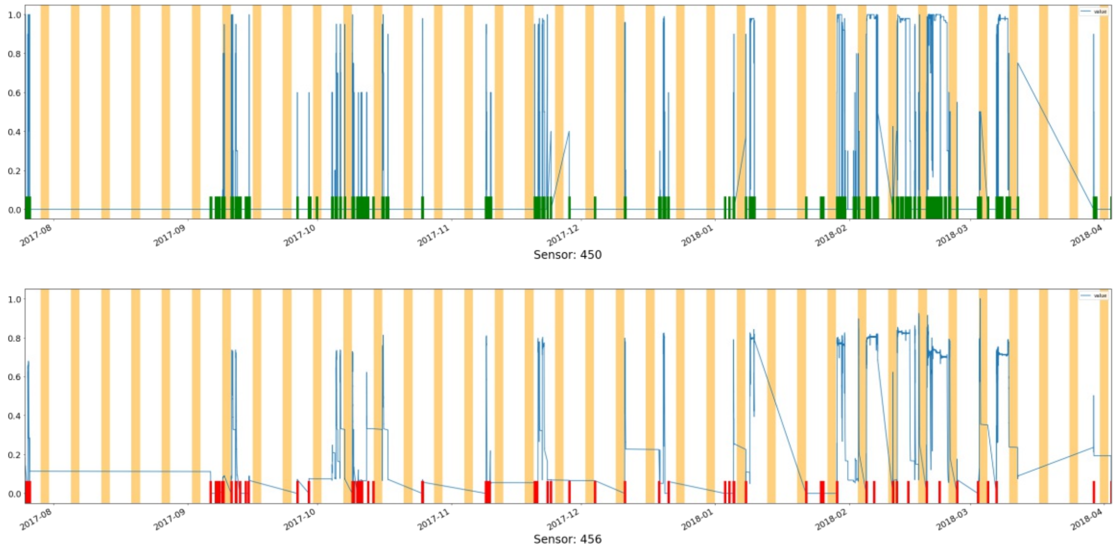


Figure 4.8: Production speed (450) and the production temperature of the heating element (456). The orange bars mark weekends. Occurrences of alarms are marked with green lines and errors are marked in red.

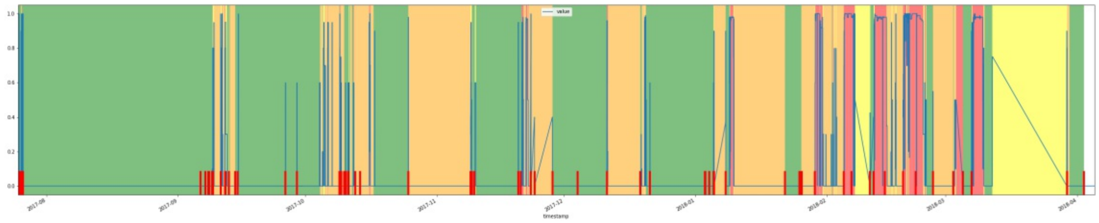
State	Number of errors	Number of alarms
Undefined (0)	71	132
Clearing (1)	142	444
Stopped (2)	50	118
Starting (3)	11	695
Holding (10)	21	573

Table 4.5: Number of errors and alarms in the particular states.

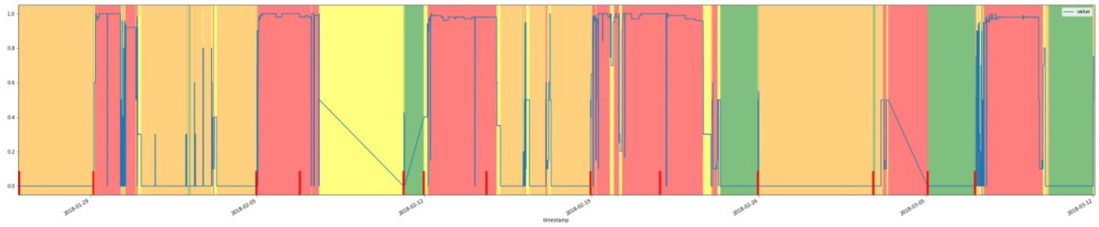
indicate a problem in terms of machine health. Figure 4.9b shows a more detailed view of a time period of approximately 6 weeks. It can be seen that errors occur in this period while the production street is running at the highest observed capacity. In this case it is more likely that a critical event occurred. Table 4.5 shows the number of occurring errors for each ISA-88 state.

Alarms

Alarms are stored in lists of Integers which can be mapped to messages that are displayed on the graphical user interface of the production street. So each of the 1962 data points in the alarm table contains a list of 0 to 32 alarms. A list with the length n indicates that n alarms are triggered at the same time. The alarms appear in different states of the production street. The number of occurrences can be seen in Table 4.5. 695 alarm lists are triggered in the state *starting*, which are the most interesting for the reverse engineering of incidents that are critical for the machine health as described in Section 4.3.6. Alarms



(a) Occurrence of errors over the full time span of the recordings.



(b) Occurrence of errors in a time frame from 2018/01/26 to 2018/03/12.

Figure 4.9: Function of the production speed. The occurrence of errors is marked with red lines on the bottom of the plot. The ISA-88 states *clearing*, *stopped*, *starting* and *holding* are painted as green, orange, red and yellow blocks.

can be identified as machine relevant, such as described in Table 4.6, and humen health critical, such as open doors or triggered light barriers.

Assembling of the Ground Truth

Since for the scope of this thesis there is neither a manual service book that contains information about maintenance efforts, nor a sufficient digital recording of failures or incidents, the ground truth has to be reverse engineered. This is done by combining the results of the analysis steps above and with the involvement of domain experts. Criteria for defining irregularities within the machine health, whose prediction is an advantage for the business partner, is based on alarm messages, errors, the machine state, and knowledge of qualified personal.
For measuring the status of the production street the ISA-88 states are used. The decision against using internal states is made for to two reasons. First, only errors which occur while the machine is producing are used as critical incidents. The producing state is documented well by the ISA-88 state *starting*, as discussed in Section 4.3.6. Seconds, the interpretation of the internal states is partly possible as discussed in Section 4.3.6, but also contains potential for misinterpretation. Therefore, the simpler and safer variant for state measurement is used for the extraction of critical incidents.

Incidents Based on Errors
Table 4.5 shows the relation of errors to the state of their occurrence. 11 of the errors are triggered while the production street is in the state *starting*, which is assumed to be

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the state where the machine is actually producing, according to Section 4.3.6. The errors occur between approximately 11 and 23 hours after reaching that state. Within the 11 errors, 3 times 3 different errors and 1 time 2 different errors appear within one second. Those errors are considered indicators for the same incident, respectively. Therefore, 4 failure incidents are detected while the production street is in the state *starting*. Errors in other states are ignored for the ground truth, because incidents that trigger errors, but do not actually concern the machine health cannot be distinguished from incidents that are in fact critical for the machine health.

Incidents Based on Alarms
According to the experts alarms in states other than *starting* are mostly triggered by manual actions as well. Therefore, they are ignored for the location of critical accidents. That leaves 695 of the total 1962 alarms, as can be seen in Table 4.5. The remaining alarms are filtered successively after the following criteria:

- $(t_{alarm} - t_{starting}) < 1800$ seconds:
  The mean duration without interruption of the producing state is approximately 35 minutes, while the median is only 67 seconds 4.4. To filter out alarms that occur due to bad synchronization in the starting phase of the machine, any alarms that occur in the first 30 minutes after the production street switches to the state *starting* are dismissed, accepting the downside of dismissing incidents in the first batch of data. Therefore, 426 alarms are are dropped, leaving 269 alarms.
- Semantic meaning of the associated message, which is displayed on the graphical user interface. Alarms regarding issues which are not relevant for the machine health like human safety for example, are dismissed. This alarms reflect open doors and light barriers. By this technique another 213 alarms are dismissed. The remaining messages can be seen in Table 4.6

The incidents extracted by analyzing the errors are combined with those that are reverse engineered by alarm analysis. Again, incidents which appear within 2 second are generalized as the same event. Hereby another reduction by 23 alarms is done. The steps above result in a ground truth of 33 events over the whole recording period that are considered critical for the machine health.

4.4 Data Preparation

4.4.1 Data Cleansing and Dimensionality Reduction

After the six iterations of analysis and discussions with domain experts of the device, the data is reduced to meaningful features by the following steps of data cleansing. The initially 764 tables contain 82 tables in various sizes and formats, which represent data that supports the operating system and the graphical user interface of the device. Except

Alarm: 49	MAIN DRIVE - INVERTER 1402T1-3
Alarm: 31	GAS PRESSURE TOP PLATE 1508S2
Alarm: 55	FAULT SUPPLY SERVO DRIVES
Alarm: 119	SAFETY FUNCTION MAIN DRIVE ACTIVE 1402T1-3
Alarm: 134	SAFETY FUNCTION FEEDING BELT ACTIVE
Alarm: 51	BATTER PUMP - INVERTER 1402T1-4
Alarm: 21	EXHAUST SYSTEM - PRESSURE SWITCH MINIMUM 1507S2
Alarm: 32	MIXTURE PRESSURE TOP 1508S5
Alarm: 34	FLAME MONITORING FRONT, TOP 1514A5
Alarm: 33	PRESSURE SWITCH GAS MAX. 1507S5
Alarm: 35	FLAME MONITORING FRONT, BOTTOM 1514A7
Alarm: 29	OVERTEMPERATURE

Table 4.6: Semantics of the relevant remaining alarms.

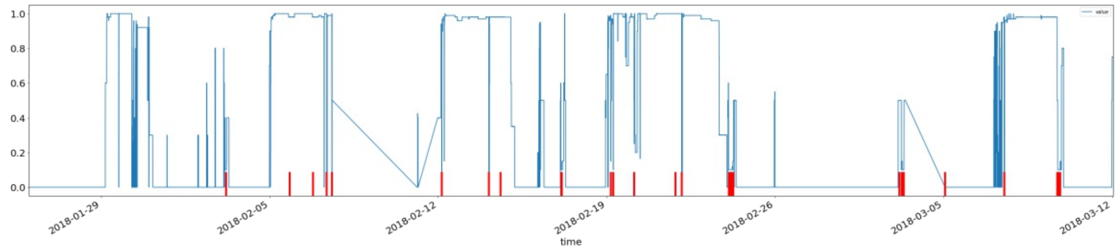


Figure 4.10: Final incidents which are considered critical for the machine health. The incidents are marked red. The the production speed is plotted in blue.

for one table, the included information is identified to be not relevant by discussions with domain experts. One remaining table contains information which maps the IDs of the tables to context information as stated in Section 4.3.4. The other tables are not used for further analysis.
682 tables of the initial data base feature the same two columns *timestamp, value*. The dimensionality of this data is reduced by the following criteria:

- **Statistical criteria:** empty tables, or features with $\sigma = 0$, are not considered for further analysis.
- **Expert feedback:** is identified by context analysis and with the help of expert knowledge. This semantic criteria involves data which holds redundant information or information which doesn't concern the machine health, as well as data which is identified to hold faulty information or dummy information for enabling future development.

Dimensionality Reduction by Statistical Criteria

Regarding the statistical analysis, special attention is directed to the standard deviation σ. For every device the number of relevant features n is determined as described in Equation 4.1:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

$$n = |A|, \text{ where } A = \{\sigma_i | \sigma_i > 0\} \tag{4.1}$$

Features with $\sigma = 0$ are dropped for further work, since they don't hold relevant information. By this strategy the originally 682 tables are reduced to 36 features, which equals a compression factor of approximately 19.

Parameter Reduction by Expert Feedback

Besides the rejection of features on the basis of statistical considerations, the data is split into two subsets. One holds information and measurements which are directly connected to the machine health status and will be used as training data set for machine learning methods. The other set contains meta information which is useful to reverse engineer the state of the machine and finally the ground truth. It will not be used as input for automated health detection.

As mentioned in Section 4.3.2, the statistics of the recording intervals is influenced strongly by sparse data. After the reduction of features by the described steps, 10 dimensions of measurements remain. This dimensions can be clustered to 4 topics as shown in figure 4.11.

The updated statistics on the sampling rates can be seen in Table 4.8. The maximum of the interval between two recordings stays the same, which indicates a long period without recordings where the device was standing idle. The median of the means and medians is reduced by approximately the factor 2570 and 40050. In the updated table, the median of the medians of the sampling rate reflects average sampling rate of approximately 2 seconds, which can be seen in intervals of the measurements when the machine is in a producing state. The larger mean is a result of the gaps where the device is in an idle state and no entries are written. Compared to the initial sampling rates in Table 4.1, the idle times of the sensors are reduced.

Table 4.7 shows the normalized statistics of 10 remaining measurements for the machine health after the reduction steps described in this section. The statistics include:

- The number of total data points in the table *num_entries*
- Mean and median of all measurements of the sensors
- The smallest and biggest measurement within the table *min_val*, *max_val*
- $0,25$ and $0,75$ quantiles
- Variance

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

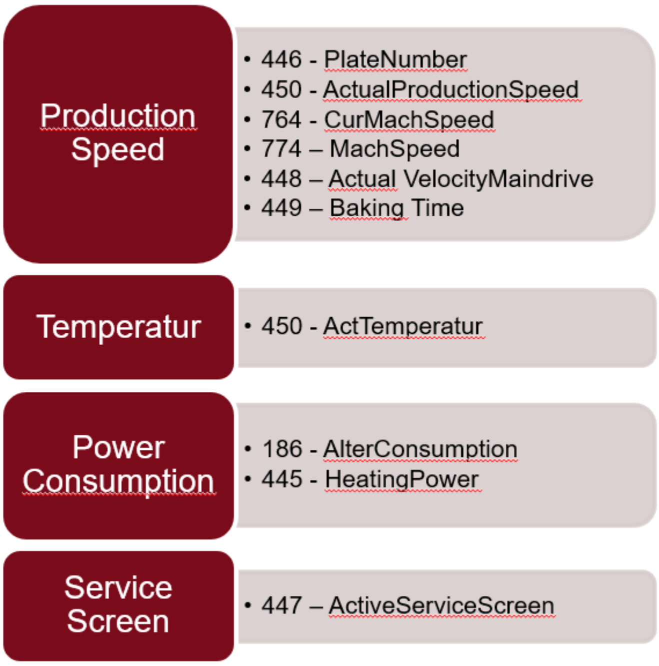
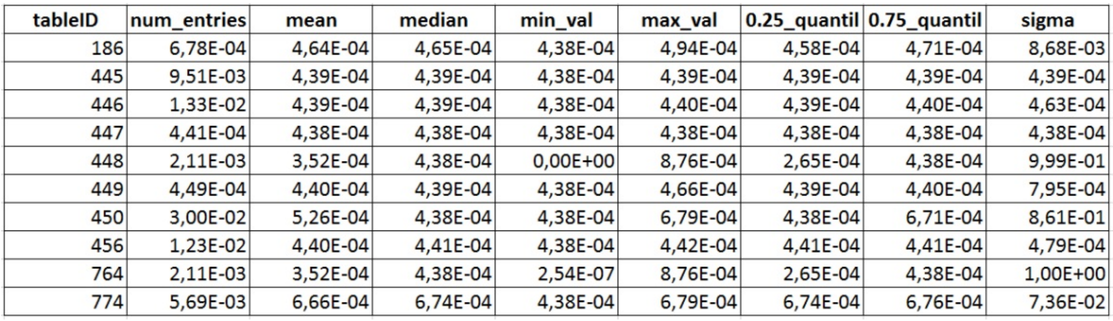


Figure 4.11: Remaining 10 dimensions after parameter reduction.



tableID	num_entries	mean	median	min_val	max_val	0.25_quantil	0.75_quantil	sigma
186	6,78E-04	4,64E-04	4,65E-04	4,38E-04	4,94E-04	4,58E-04	4,71E-04	8,68E-03
445	9,51E-03	4,39E-04	4,39E-04	4,38E-04	4,39E-04	4,39E-04	4,39E-04	4,39E-04
446	1,33E-02	4,39E-04	4,39E-04	4,38E-04	4,40E-04	4,39E-04	4,40E-04	4,63E-04
447	4,41E-04	4,38E-04	4,38E-04	4,38E-04	4,38E-04	4,38E-04	4,38E-04	4,38E-04
448	2,11E-03	3,52E-04	4,38E-04	0,00E+00	8,76E-04	2,65E-04	4,38E-04	9,99E-01
449	4,49E-04	4,40E-04	4,39E-04	4,38E-04	4,66E-04	4,39E-04	4,40E-04	7,95E-04
450	3,00E-02	5,26E-04	4,38E-04	4,38E-04	6,79E-04	4,38E-04	6,71E-04	8,61E-01
456	1,23E-02	4,40E-04	4,41E-04	4,38E-04	4,42E-04	4,41E-04	4,41E-04	4,79E-04
764	2,11E-03	3,52E-04	4,38E-04	2,54E-07	8,76E-04	2,65E-04	4,38E-04	1,00E+00
774	5,69E-03	6,66E-04	6,74E-04	4,38E-04	6,79E-04	6,74E-04	6,76E-04	7,36E-02

Table 4.7: Normalized statistics of the 10 remaining indicators of the machine health. The names of the sensors are obscured by random IDs.

Furthermore, basic data cleansing is applied to the remaining features. Therefore, arbitrary data formats are translated to standardized formats. For example the expression *'T#0ms'* is converted to zero of the unit milliseconds. Generally, all units of stored numbers are removed as well as brackets and other strings which surround the measured values. Numerical values are converted to python floats or integers depending on wether decimal digits are present. Strings which contain Boolean expressions are converted to standard python Boolean values.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

	smallest overall value (s)	**biggest overall value (s)**	**overall median (s)**
min(sampling interval)	$0,21$	2	$1,73$
max(sampling interval)	$3,62e+6$	$3,62e+6$	$3,62e+6$
mean(sampling interval)	$9,91$	$1,01e+5$	$114,57$
median(sampling interval)	2	396	2

Table 4.8: Updated statistics of the sampling interval of measurements after the data set is reduced by statistical criteria and expert feedback.

```
1   if the time stamp exists in the runtime:
        copy time stamp and value
2   else: map the time stamp by minimizing the
        distance to the next entry in the runtime
3   if more than one measurement of one sensor
    point to one time stamp in the runtime:
        average the value
4   if an entry in the runtime is empty after
    applying the rules above:
        fill it with the over all average of the sensor
```

Table 4.9: Rules for mapping measurements to the runtime.

4.4.2 Preprocessing of the Data for Machine Learning

To apply a machine learning method, there are still obstacles to overcome:

1. The machine is in a non producing state over 50% of the time. The behavior of sensors varies strongly between different states, which biases the prediction of small changes caused by degradation. The machine learning algorithms are influenced strongly by the heavy changes of the sensors.
2. Measurements are not continuous, but vary between approximately 1.5 seconds and 46 days.

For machine learning methods to work properly, the data has to be normalized to be independent from the obstacles above. Therefore, the following strategy is pursued.

First, a continuous time axis is created. It covers the whole time span of the device from the first time stamp to the end of the record. The sampling rate is normalized to 30 seconds. Second, each sensor is mapped to the continuous time line, which henceforth will be referred to as *runtime*. The mapping is based on the rules in Table 4.9.

Third, the alarms are concatenated to the runtime by mapping the occurrences to the nearest time stamp. Also, the time spans in which the machine is considered to be not in a producing state are dropped. The decision if the device is running or not is based on the ISA-88 state as mentioned in Section 4.3.6.

4.4.3 Feature Engineering

To evaluate the performance of the neural network described in Chapter 3, the data set is prepared for regression for comparison of the metrics. As stated in Chapter 1, machine health can be predicted by two different approaches: classification and regression. The data in this case study does not enable the differentiation between a true failure and incidents which result not necessarily in total failure. To be able to provide a warning 30 minutes before an incident occurs independent of whether it is a failure or another incident, additionally a classification analysis is evaluated.

Feature Engineering for Regression

Similar to the preparation steps in the first case study described in Section 3.4 a window of $n_{timesteps}$ is convoluted over the rows of measurements, which contain 10 values each. Therefore the content of the window is concatenated to vectors of the length $10 * n_{timesteps}$, creating vectors which contain the features of a time span of $n_{timesteps}$. The label for each vector is the remaining useful live of the n^{th} timestep in cycles, where for this data set one cycle equals 30 seconds.

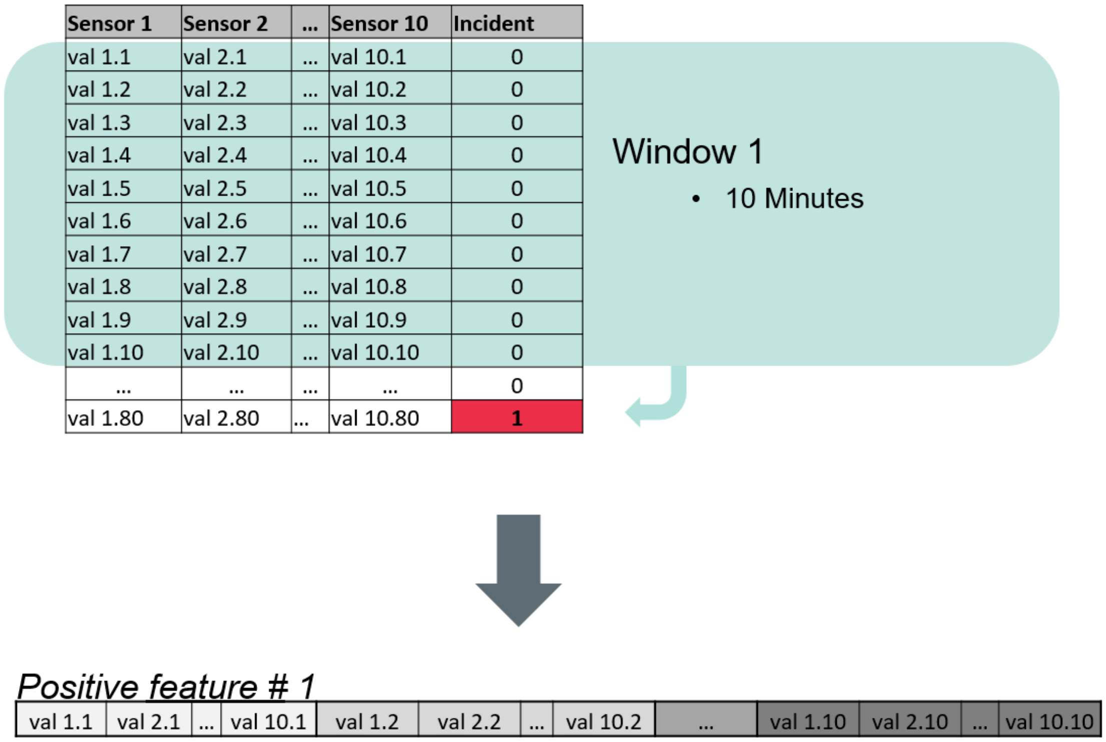
Feature Engineering for Classification

The properties of the classification are developed by discussing reasonable use in production. It was decided to grant an engineer 30 minutes of reaction time before a predicted event occurs. The properties result in the following features.

- A positive incident describes a point t in time, for which applies: in $t + 30$ minutes an anomaly occurs.
- A negative incident describes a point in time t, for which applies: in $t + 30$ minutes no anomaly occurs.

To create features, a window is convoluted over the runtime of the the 10 sensors which remain after the dimensionality reduction described in Section 4.4.1. The window size is varied in $1, 5, 10, 20$, which is equivalent to data of $30, 150, 300, 600$ seconds. The different window sizes are evaluated together with the hyper parameters of the algorithm with a grid search in the sections below. The values of each point in time in the windows are concatenated to a feature vector, representing the time frame, with a distance of 30 minutes to a positive or negative incident. Figure 4.12 shows the generation of a feature

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



Sensor 1	Sensor 2	...	Sensor 10	Incident
val 1.1	val 2.1	...	val 10.1	0
val 1.2	val 2.2	...	val 10.2	0
val 1.3	val 2.3	...	val 10.3	0
val 1.4	val 2.4	...	val 10.4	0
val 1.5	val 2.5	...	val 10.5	0
val 1.6	val 2.6	...	val 10.6	0
val 1.7	val 2.7	...	val 10.7	0
val 1.8	val 2.8	...	val 10.8	0
val 1.9	val 2.9	...	val 10.9	0
val 1.10	val 2.10	...	val 10.10	0
...	...	...	...	0
val 1.80	val 2.80	...	val 10.80	1

val 1.1	val 2.1	...	val 10.1	val 1.2	val 2.2	...	val 10.2	...	val 1.10	val 2.10	...	val 10.10

Figure 4.12: Process of generating a feature vector from a time frame of 10 minutes for a critical event, which occurs 30 minutes after the window.

vector for one positive event.

For the evaluation, the data is split by an 80 by 20 ratio, where the test set consists of 20% of the data. The split is done randomly under consideration of an aliquot number of positive incidents in each set, resulting in a train set of 69098 samples and a test set of 17275 samples.

4.5 Modeling

To compare the results to the first case study, the RUL is estimated by the random forest regression and the in the first case study described neural network. Additionally, the performance of a classification task is evaluated to further support the objective of giving a warning 30 minutes before an incident happens. By a grid search, the optimum of the hyper parameters of the random forest and the neural network proposed in the first case study are evaluated similarly as described in the first case study. The hyper parameter evaluation of the multi layer perceptron is described in Section 4.5.1

The following models are optimized by hyper parameter evaluation, trained and evaluated:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- Regression
    - Ramdom forest regression
    - Regression of the method described in the first case study
- Classification
    - Ramdom forest classification
    - Multi layer perceptron classification

4.5.1 Hyper Parameter Evaluation of the Multi Layer Perceptron Classification Algorithm

With a window size of 20, hyper parameters for the multi layer perceptron classifier are evaluated by a grid search, which 200 epochs for each trial, for the parameters below. This way the optimal configuration is estimated by a *constant* learning rate, hidden layer size of $100, 1$, batch sizes of 10 and the *logistic* activation function.

- Learning_rate: constant, invscaling, adaptive
- Hidden layer sizes: $(100, 1), (100, 2), (100, 3)$
- Batch sizes: $10, 50, 100, 200$
- Activation functions: logistic, relu

4.6 Evaluation

To evaluate to what extend the RUL can be predicted on the given data set, the regression task is evaluated on basis of the metrics described in Section 3.1.1. For the classification of predicting whether an incident will occur in 30 minutes or not, the evaluation metrics is a confusion matrix and further the accuracy and sensitivity.

4.6.1 Results of the RUL Detection

First the baseline is set by the random forest regression as in the first case study. Due to the difference in quality of the data the score degrades by a factor of roughly $3,3e62$ in comparison to the experiments on the benchmarc data set. The MAE and MSE degrade by the factor of 82 and 5102. R^2 only decreases by $0,04$. Using the neural network described in the first case study, instead of the random forest regression algorithm, the score increases by a factor of $9,5e53$. Also the MAE and MSE increase by a factor of 20 and $4,8$, R^2 by $0,27$. Comparing the performance of the state of the art neural network described in Chapter 3 between the benchmark data set and the sparse data set of the second case study, it can be seen that the score differs by $349911510,3$. In the first case study the MAE is smaller by a factor of roughly 8 and the MSE by a factor of roughly

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4084. The R^2 increases by $0,19$. To approximate the degradation of the data, the devices and features are reduced as described in Section 3.6.2. The comparison of the RUL regressions can be seen in Table 4.10. The score of the predictions for the second case study exceeds the score of the predictions for the first case study by $117374300,8$. The MAE and MSE of the predictions in the second case study are by $41,18$ and $1048976,41$ higher than those in the sparse data set of the first case study.

Description	**score**	**MAE**	**MSE**	**R^2**
First case study:				
Random Forest	22152,25	23,73	995,01	0,42
Stacked LSTM with deep autoencoder	468,73	11,88	257,98	0,84
Stacked LSTM with deep autoencoder with 1 device and 10 features	232537678,20	55,78	4681,59	-20,43
Second case study:				
Random forest regression	3,3e62	1957,8	5076501,9	0,38
Regression of the neural network described in Chapter 3	349911979,00	96,96	1053658,0	0,65

Table 4.10: Experimental results of the regression algorithms.

4.6.2 Results of the Automated Anomaly Detection Method

Table 4.11 shows the confusion matrix for the prediction of the test set. It can be seen that all samples are predicted to be negative. Since the test set contains only 7 positive events the accuracy is 9.99, but the sensitivity is 0.0.

Algorithm	**True negatives**	**True positives**	**False negatives**	**False positives**
Random Forrest	17268	0	7	0
Multi layer perceptron	17268	0	7	0

Table 4.11: Confusion matrix for the prediction of the test set of the data.

4.7 Discussion

In this section the results of the evaluation are compared and discussed and potential future work is pointed out.

Figure 4.14 shows the score of the random forest algorithm and the proposed neural network. The differing quality of the data sets results in highly varying results. The

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

biggest difference can be identified comparing the random forest algorithm which was used as baseline for both case studies. Looking at the neural network approach, a significant improvement can be seen in both case studies. A detailed view of the improvements can be seen in figure 4.13.

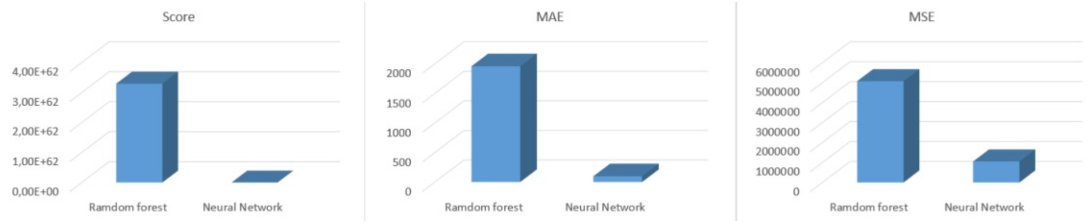


Figure 4.13: Performance comparison of the the neural network to the random forest method

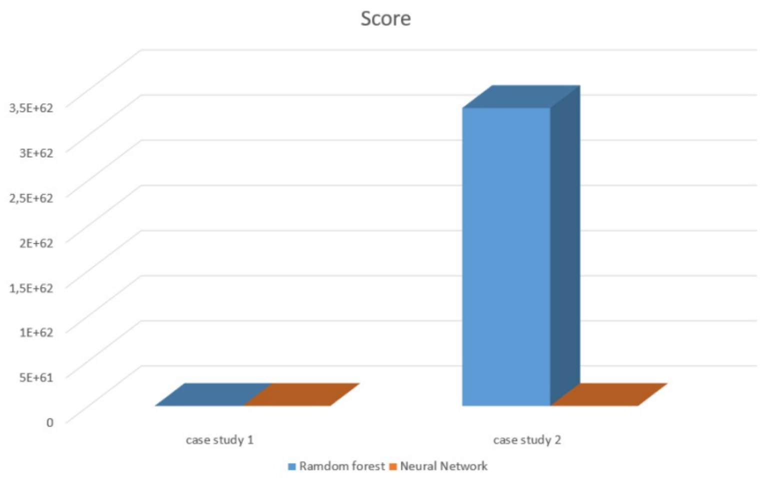


Figure 4.14: Comparison of the score of the first and the second case study

To make the results comparable, the data of the first case study is reduced to the same amount of devices and features available for the second case study. The comparison of the predictions by the neural network can be seen in figure 4.15. The results of the algorithm discussed in the fist case study degrades significantly with increasing sparsity of the data, as is discussion in section 3.7. Nevertheless, the score, MAE and MSE are still higher when applying the algorithm to the data of the food production line data set.

For the second case study the sampling rate was normalized to 30 seconds. With the mean absolute error of $96,96$ the prediction deviates from the true failure by roughly 49 minutes. Provided the fuzziness of this result, the business goal to provide 30 minutes of reaction time can not be met. Still from a business perspective value can be provided. By adding buffer time to the actual prediction and purposely skew the predictions to a point in time before the actual failure. That way outages can be reduced but regular maintenance

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

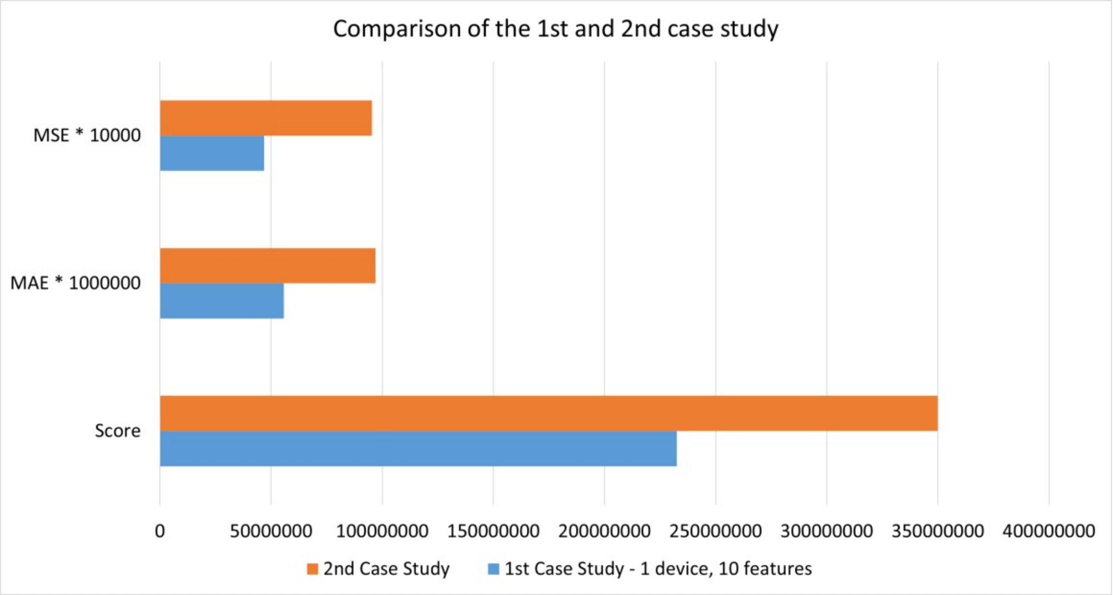


Figure 4.15: Comparison of the results achieved by the neural network in the first case study with the degraded data set and the second case study.

can still be avoided. This offers an entry point to a more flexible maintenance model, but improvements are still possible and necessary for future work. Starting points are described below.

Looking at the classification task, the applied machine learning methods are not able to identify a critical event within the given data as can be seen in Table 4.11. This result can be assembled to the following reasons:

First, the number of features used for learning and classification are sparse. At the dimensionality reduction described in Section 4.4.1, features which are not relevant for further processing due to statistical properties or their semantic content are eliminated. After the elimination 10 features remain. These 10 features partly depend on each other or describe the same underlying value. For example one feature is the batch counter and another feature measures the production speed, which directly relates to the batch counter. Therefore, the features can be clustered to 3 dimensions, which summarize the features that depend directly, as follows:

1. Speed
2. Temperature
3. Power consumption

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Furthermore, speed and power consumption features are values which are configured by the user. The only values that are measured by a sensor are the temperature of the heating unit and the batch counter. Although the influence of the other features to machine health prediction is not eliminated, they are not typical health indicators according to the literature discussed in Chapter 2.
Besides the low number of features, after the preprocessing steps and the creation of the *runtime* described in Section 4.4.2, the actual time span which is used for machine learning is approximately 700 hours, which corresponds to approximately 86350 data points. The life span of a production line, according to domain experts, is bigger than 10 years. Maintenance statistics were not available, but the probability of a failure or even breakdown within the captured data is evaluated as small. This raises the most important issue, which is the insufficient groundtruth.
Only 33 of the approximately 86350 data points are identified as critical events. These critical events are composed of a variety of different alarms and errors as described in Section 4.3.6. Therefore, the build up to the incidents cannot be assumed to be similar. In other words, the behavior of the features before an unknown number of the 33 incidents is probable to be variable. Due to the validation, a split of the data was conducted by a 80:20 ratio, leaving 26 randomly selected events for the training of the algorithms and 7 critical events for evaluation. Given the variety of events that can occur, it is possible that events are in the evaluation batch which are not in the trainings batch.

To overcome these shortcomings, two possible paths are identified:

- Improve the data
- Improve the methodology

The obvious implication for the lack of sufficient data is to improve the quality and quantity of the data. For supervised classification methods to learn the properties of a set of categories, further called clusters, three things are needed:

- clusters of data which are similar within the cluster but vary at least in one feature to other clusters
- depending on the variation of the data, the number of clusters and the used method: a minimum of data for the learning mechanisms to respond
- labels for the data

The first shortcoming is the overall quantity of the data. The most simple way to increase the amount of data is to increase the time span in which the data is measured. A more effective way is the extension of the number of monitored devices. There are already several devices monitored, but it is not possible to merge the data directly, due to the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

lack of harmonization. Some parameters cannot be harmonized due to environmental or customer specific circumstances. For example measurements of the temperature which vary heavily with different locations of the production line and seasonal changes. Also customization of features of the devices for customers influence the measurements. Examples are the recipe and shape of the processed food. But the standardization of machine states, an overall identification for sensors and a differentiation between measurements and control or interface data within the database would decrease the effort for merging the data of different devices drastically. Another way to increase data is to increase the number of measurements. Additionally to the existing measurements the following possibilities to add sensors are identified: Multiple measurements of critical parts of the production line increase the fragmentation of the monitoring and allow to predict the failure of specific parts. Different dimensions of measurements increase the probability to detect the degradation of parts. State of the art measurements include vibration, temperature, power consumption, revolution counting and domain specific wearing measurements like strain gauge. Experiments with degradation of features in the first case study indicate to use at least 18 features for 100 devices to be able to predict a failure with an mean absolute error of 30 minutes. [LWQM17] strongly recommend the usage of vibration sensors.

But an increased number of data points only increases the data quality if the corresponding labels are sufficient. Therefore, an appropriate strategy for assembling the ground truth is needed. A first step would be to increase the intelligibility of already captured incidents. This can be achieved directly by meaningful alarm and error messages. Furthermore, a designated separation between incidents regarding the machine health and other errors and warnings which concern human security, the product itself, or other issues would simplify the reverse engineering of critical incidents or failures. To be able to distinguish between different kinds of incidents, the segmentation can also reach further to document which part is affected or categorize the incident in an appropriate way. This strategy must still be equal at all devices, while respecting the subdivision between errors, alarms and warnings.

Besides the internal documentation of incidents, an addition is an external maintenance documentation. At the failure of a device, the state it is in is not stable. Therefore, an internal documentation can easily fail or be faulty. In that case a manual maintenance protocol can complement the missing data. The protocol can either be a book where the exact date and incident is documented, or a separate small system which for example only allows the maintenance worker to restart the production line if a summary of the incident is entered. Another approach would be to sell a full service contract in combination with the device, where maintenance is only conducted by known personnel which is instructed to document all actions properly. A more sophisticated method is to launch run to failure experiments with a production line. This would have several advantages. On one hand data can be collected and documented meticulously. The data captures the whole span of degradation up to the failure. At the same time, typical weak points of the device can be identified. This information can be used to place additional sensors in strategically

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

optimized places.

Besides improving the data, the second approach is to improve the methodology, which is also an interesting topic for future work. The iterative exploratory approach proved to be very useful. At each iteration, we were able to recover new information. The alternation between analysis and expert feedback supported and complemented the process in both directions: The understanding of the data for machine learning and the improvement of the data for the domain experts. A possible enhancement for the iterative process is to start earlier in the development process of a product, to identify possible improvements in the designing phase and thus reduce the cost for changes.

Regarding machine learning, it was shown that the used methods are not sufficient to predict failures with the available data. One approach to automatically simulate a higher quantity of data is to train generative neural networks to generate simulated incidents. Additionally more extensive experiments can be conducted to optimize network architectures which can handle sparse data.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 5

Conclusions

In this thesis two different settings are studied, in order to think about a suitable methodology for predicting the remaining useful life of a machine that can cope with the different preconditions that the settings implicate. In the first case study, the data is well defined and the groundtruth is known. In the second case study, a high level of uncertainty is present and the groundturth has to be reverse engineered to create the training and evaluation data set. Looking at the CRISP-DM methodology, the data preparation and modeling in the first case study can start immediately. The missing groundtruth request further steps of analyzing and disassembling to be able to reverse engineer incidents, which can further be marked for the training and evaluation of the data set. Another difference is the the data of the first case study is a run to failure data set. This means the incidents of failure can be assumed to be absolutely true and there is a periodic recording of measurements of the degradation that leads to a failure. That implies three different parts of information:

1. A number of measurements where no failure occurred
2. A number of incidents or failures which are known
3. Measurements of the degradation that lead to a failure

In contrast, in the second case study, the incidents are an assumption due to findings within the analysis of the data. Data where no incident is assumed automatically is suspected to include degradation that leads to the next assumed incident. Also the recording of the data is only partly periodic. To create a constant time series, it has to be interpolated and organized, and therefore modified. These distortions of the data potentially hinders the machine learning algorithm to perform at its optimum, as discussed in the comparison in Section 4.7.

The above differences between the case studies have an impact on the performance of the used machine learning algorithms, but are not considered by the algorithms itself. The methodology that leads to the usage of the machine learning algorithms however is influenced highly. Thinking of the whole process leading to the results, the potential of rising the degree of automation including both scenarios are identified in three strategies:

- Automation of the date preparation steps
- Refining the algorithm to be able to cope with variations that come with different scenarios like degree of sparsity of incidents
- Optimize the aggregation of the data to reduce the diversity between the scenarios

To simplify the analysis steps necessary in the second case study, parts of the engineering can be automated and conducted with their own intelligence. For the automated feature detection, a whole field of research exists [HA04]. Also for data amputation problems different approaches already exist and are refined [CFB$^+$03]. By joining these methods to a semi automated or automated pipeline, the efficiency of data preparation can be increased.

The focus on this thesis was to conduct experiments on two different scenarios with one neural network architecture. Although the results of the measured metrics differ, the difference is larger by a factor of approximately $2e52$ compared to the random forest algorithm which was assumed as baseline. Further refinement of the architecture of the neural network appears promising for future work. Especially noise reduction seems to be a promising topic for future work: With a one layered autoencoder the results of the first case study increased slightly as discussed in Chapter 3. Deep autoencoder weren't able to improve the final results. Further techniques used in state of the art methods include variational autoencoders [FvA14], which also would be an interesting next step for further investigations. Aside from that the noise reduction could also be improved in a next step by adding randomized, distributed noise to the input of an autoencoder. With this technique the smoothing of the encoder would increase, which potentially has a good effect for further processing.

Another way would be to properly record incidents from the beginning or create a run to failure data set in a test environment. But running a machine until its failure for the purpose of data acquisition is difficult in a commercial environment and according to experts not an accepted scenario. Therefore, for the future, data driven approaches connected to management methods which introduce a clean collection of measurements have great potential to converge the possibilities and results of real world scenarios similar to the second case study of this thesis.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

List of Figures

2.1 Selection of the categorization of AI based methods used in system health monitoring described by Kahn and Yairi [KY18]. 6
2.2 [KY18] This figure illustrates the structure of an autoencoder. Input (a) and Output (b) have the same number of dimensions, where the hidden layers show a reduced number of neurons. 8
2.3 Structure of a CNN. [AM17] 9
2.4 Comparison between a simple neural network (a) and the RNN structure (b). [KY18] 10
2.5 Standard process for extracting information from data, according to the 'Cross industry standard for data mining' [She00] 11

3.1 Boxplots of the C-MAPSS data set. 20
3.2 Selected sensor data of C-MAPSS data for device 1. The degradation of the device can be seen in the curves. 21
3.3 Generation of a feature vector by sliding a window of the size $n_{timesteps}$ over the rows of measurement. 22
3.4 Feature generation by overlapping windows. 25
3.5 Performance of the LSTM with one hidden layer before- and after the hyper-parameter evaluation. The R^2 and the MAE is scaled by a factor of 1000 and 10, respectively for better visibility of the graph. 33
3.6 Score of the stacked LSTM after noise reduction for the remaining cycles of the devices. 34
3.7 Predicted RUL on x axis versus the true RUL on the y achsis of the stacked LSTM before and after noise reduction by an autoencoder with 1 hidden layer of 18 dimensions. The green line symbolizes the most accurate prediction. 35
3.8 Delta between the true RUL and the predicted RUL of the stacked LSTM plotted along the axis of the true RUL before and after noise reduction by an autoencoder with 1 hidden layer of 18 dimensions. 35
3.9 Predicted RUL on x axis versus the true RUL on the y achsis of the algorithm with 12 features and 3 features. The green line symbolizes the most accurate prediction. 37

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.10 Degradation of the results with decreasing number of available features. To improve the visualization, the score is divided by 100 and the MAE is multiplied by 100. 38
3.11 Degradation of the results with decreasing number of available devices. To improve the visualization, the score is divided by 100, the MAE is multiplied by 10000 and the MSE is multiplied by 100. 38

4.1 Extract of one of the 682 tables in the format *timestamp, value*. 42
4.2 Batch counter of the device over the whole time period. The counter of a fixed number of pieces in the production is normalized to obscure the actual number of pieces that is produced. The orange bars mark weekends. . . 45
4.3 Saw-tooth textured measurement of the batch counter. 46
4.4 The 10 remaining features after conducting the dimensionality reduction. The data is normalized and the table names are obscured by random ids. The orange blocks mark weekends. 47
4.5 The starting sequence which can be found repeatedly in the data. 50
4.6 Correlation between internally used states and the ISA-88 states. To measure the simultaneous occurrences of the states, a time frame is slid over the ISA-88 states and the occurrences of the internal states within the frame are counted. The number of occurrences is emphasized by the size of the font. 51
4.7 Production speed and the production temperature of the heating unit overlapped with the ISA-88 states the production street is in. The normalized production speed is plotted as a function and the states *clearing*, *stopped* *starting* and *holding* are painted as green, orange, red and yellow blocks. . 52
4.8 Production speed (450) and the production temperature of the heating element (456). The orange bars mark weekends. Occurrences of alarms are marked with green lines and errors are marked in red. 53
4.9 Function of the production speed. The occurrence of errors is marked with red lines on the bottom of the plot. The ISA-88 states *clearing*, *stopped*, *starting* and *holding* are painted as green, orange, red and yellow blocks. 54
4.10 Final incidents which are considered critical for the machine health. The incidents are marked red. The the production speed is plotted in blue. . 56
4.11 Remaining 10 dimensions after parameter reduction. 58
4.12 Process of generating a feature vector from a time frame of 10 minutes for a critical event, which occurs 30 minutes after the window. 61
4.13 Performance comparison of the the neural network to the random forest method 64
4.14 Comparison of the score of the first and the second case study 64
4.15 Comparison of the results achieved by the neural network in the first case study with the degraded data set and the second case study. 65

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Tables

3.1 Statistics of the C-MAPSS data set. The first three rows are operational settings, followed by 21 rows of sensor measurements. 19
3.2 Grid search of the hyperparameters for the random forest regression algorithm and the parameters with the best result. 20
3.3 Performance of the random forest prediction algorithm after the hyperparameter evaluation. This results are produced by using the parameters of table 3.2 21
3.4 Best guesses of the remaining parameters while doing the evaluation for the number of cycles used for the feature vectors. 23
3.5 Metrics and the rankings for the evaluation of the number of cycles, which are packed to one vector to create the input for the LSTM. 24
3.6 Best guesses of the remaining parameters while doing the evaluation for the number of cycles that are overlapping. 24
3.7 Metrics and the rankings for the evaluation of the overlapping. 25
3.8 Best guesses of the remaining parameters while doing the grid search evaluation for the number of epochs and the batch size. 25
3.9 Metrics and the rankings for the grid search of the number of epochs and the batch size. 26
3.10 Best guesses of the remaining parameters while doing the evaluation for the optimization algorithm. 26
3.11 Metrics and the rankings for the evaluation of the optimization algorithm. 26
3.12 Best guesses of the remaining parameters while doing the evaluation for the loss function. 27
3.13 Metrics and the rankings for the evaluation of the loss function. 27
3.14 Best guesses of the remaining parameters while doing the evaluation for the number of neurons used in the hidden layer. 27
3.15 Metrics and rankings for the evaluation of the number of neurons used in the hidden layer. 28
3.16 Summary of the hyperparameters, which are assumed to build an optimum for the scenario evaluated in Section 3.5. 29
3.17 Comparison of the performance of stacked LSTM layers. 29
3.18 Comparison of different architectures of autoencoders with random forrest prediction. 30

3.19 Output of the 4 dimensional autoencoder used as noise reduction for the LSTM with 1 hidden layer. 30

3.20 Overview of the champions of the different network architectures. It is ordered by the complexity of the layers from simple to most complex. 31

3.21 Degradation of the results of the stacked LSTM with 1 layer autoencoder of 18 dimensions with reduced number of features for training. 31

3.22 Degradation of the results of the stacked LSTM with 1 layer autoencoder of 18 dimensions with reduced number of devices for training. 32

3.23 Parameters used for the first trial set by an educated best guess, as starting point for the parameter evaluation. 32

3.24 10 predictions with the highest score of the stacked LSTM before noise reduction. 33

3.25 Worst 10 predictions of the stacked LSTM after noise reduction by an autoencoder with 1 hidden layer of 18 dimensions. 36

3.26 Selection of metrics for the algorithms ranked best on the PHM'08 Challenge data based on the test set published by Ramasso et al. [RS14] and other publications. 36

4.1 Illustration of the boundary values of the sampling rates. The different rows show statistics over the data tables: minimum, maximum, mean and median. Columns summarize the statistics over all data tables: The smallest minimum of all tables, the biggest minimum of all tables and the average minimum of all tables, and so on. It can be seen that the interval between two samplings lies between $0,21$ seconds and $3,62e+6$ seconds. The median of the means is $2,93e+11$ and the median of medians is $8,01e+4$. 44

4.2 Number of occurrences of the states. 49

4.3 Connection between ISA-88 and internal states, which occur together within a time frame of 2 seconds over 100 times. 49

4.4 Mean and median duration of the ISA-88 states. 50

4.5 Number of errors and alarms in the particular states. 53

4.6 Semantics of the relevant remaining alarms. 56

4.7 Normalized statistics of the 10 remaining indicators of the machine health. The names of the sensors are obscured by random IDs. 58

4.8 Updated statistics of the sampling interval of measurements after the data set is reduced by statistical criteria and expert feedback. 59

4.9 Rules for mapping measurements to the runtime. 59

4.10 Experimental results of the regression algorithms. 63

4.11 Confusion matrix for the prediction of the test set of the data. 63

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Bibliography

[AAB$^+$15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[AG18] N. Amruthnath and T. Gupta. A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In *5th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 355–361, April 2018.

[AM17] Saleh Albelwi and Ausif Mahmood. A framework for designing the architectures of deep convolutional neural networks. *Entropy*, 19(6), 2017.

[AVR$^+$09] K. Abdennadher, P. Venet, G. Rojat, J. M. Retif, and C. Rosset. Kalman filter used for on line monitoring and predictive maintenance system of aluminium electrolytic capacitors in ups. In *IEEE Energy Conversion Congress and Exposition*, pages 3188–3193, Sept 2009.

[BMK$^+$20] Tarek Berghout, Leïla-Hayet Mouss, Ouahab Kadri, Lotfi Saïdi, and Mohamed Benbouzid. Aircraft engines remaining useful life prediction with an adaptive denoising online sequential extreme learning machine. *Engineering Applications of Artificial Intelligence*, 96:103936, 2020.

[BS16] Lorenzo Beretta and Alessandro Santaniello. Nearest neighbor imputation algorithms: a critical evaluation. *BMC medical informatics and decision making*, 16(3):74, 2016.

[BWES04] Carl S Byington, Matthew Watson, Doug Edwards, and Paul Stoelting. A model-based approach to prognostics and health management for flight control actuators. In *2004 IEEE Aerospace Conference Proceedings*, volume 6, pages 3551–3562. IEEE, 2004.

[BZL16] Giduthuri Sateesh Babu, Peilin Zhao, and Xiao-Li Li. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International conference on database systems for advanced applications*, pages 214–228. Springer, 2016.

[C$^+$15] François Chollet et al. Keras. `https://keras.io`, 2015.

[CFB$^+$03] Stefano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, and Maristella Matera. *Morgan Kaufmann series in data management systems: Designing data-intensive Web applications*. Morgan Kaufmann, 2003.

[CJL$^+$18] William Chan, Navdeep Jaitly, Quoc V Le, Oriol Vinyals, and Noam M Shazeer. Speech recognition with attention-based recurrent neural networks, June 5 2018. US Patent 9,990,918.

[FK18] Masakiyo Fujimoto and Hisashi Kawai. Comparative evaluations of various factored deep convolutional rnn architectures for noise robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4829–4833. IEEE, 2018.

[FvA14] Otto Fabius and Joost R van Amersfoort. Variational recurrent autoencoders. *arXiv preprint arXiv:1412.6581*, 2014.

[HA04] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.

[HHAL18] Yuhuang Hu, Adrian Huber, Jithendar Anumula, and Shih-Chii Liu. Overcoming the vanishing gradient problem in plain recurrent networks. *arXiv preprint arXiv:1801.06105*, 2018.

[JSV$^+$16] Olivier Janssens, Viktor Slavkovikj, Bram Vervisch, Kurt Stockman, Mia Loccufier, Steven Verstockt, Rik Van de Walle, and Sofie Van Hoecke. Convolutional neural network based fault detection for rotating machinery. *Journal of Sound and Vibration*, 377:331 – 345, 2016.

[KY18] Samir Khan and Takehisa Yairi. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241 – 265, 2018.

[LWQM17] Chen Lu, Zhen-Ya Wang, Wei-Li Qin, and Jian Ma. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

health state identification. *Signal Processing*, 130(C):377–388, January 2017.

[LWZ+14] Jay Lee, Fangji Wu, Wenyu Zhao, Masoud Ghaffari, Linxia Liao, and David Siegel. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42(1):314 – 334, 2014.

[LZZZ20] Han Li, Wei Zhao, Yuxi Zhang, and Enrico Zio. Remaining useful life prediction using multi-scale deep convolutional neural network. *Applied Soft Computing*, 89:106113, 2020.

[MBM15] Wim De Mulder, Steven Bethard, and Marie-Francine Moens. A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech and Language*, 30:61–98, 2015.

[MGM15] Yajie Miao, Mohammad Gowayyed, and Florian Metze. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 167–174. IEEE, 2015.

[Mob02] Keith Mobley. An introduction of predictive maintenance (second edition). *Elsevier*, 2002.

[MZ09] Ken R McNaught and Adam Zagorecki. Using dynamic bayesian networks for prognostic modelling to inform maintenance decision making. In *IEEE International Conference on Prognostics and Health Management*, pages 1155–1159, 2009.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[RL16] L. Ren and W. Lv. Remaining useful life estimation of rolling bearings based on sparse representation. In *7th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, pages 209–213, July 2016.

[RS14] Emmanuel Ramasso and Abhinav Saxena. Performance benchmarking and analysis of prognostic methods for cmapss datasets. *International Journal of Prognostics and Health Management*, 5:1–15, 11 2014.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[SGSE08] A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *International Conference on Prognostics and Health Management*, pages 1–9, Oct 2008.

[Sha86] G. L. Shaw. Donald hebb: The organization of behavior. In Günther Palm and Ad Aertsen, editors, *Brain Theory*, pages 231–233. Springer, 1986.

[She00] Colin Shearer. The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22, 2000.

[SMD17] Rahul Shrivastava, Hari Mahalingam, and N. N. Dutta. Application and evaluation of random forest classifier technique for fault detection in bioreactor operation. *Chemical Engineering Communications*, 204(5):591–598, 2017.

[SOVP12] E. Sutrisno, H. Oh, A. S. S. Vasan, and M. Pecht. Estimation of remaining useful life of ball bearings using data driven methodologies. In *IEEE Conference on Prognostics and Health Management*, pages 1–7, June 2012.

[SS08] Instrumentation Systems and Automation Society. *ISA-TR88.00.02: Machine and Unit States: an Implementation Example of ISA-88*. International Society of Automation, 2008.

[SSP$^+$15] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, June 2015.

[TMMZT12] Diego Alejandro Tobon-Mejia, Kamal Medjaher, Noureddine Zerhouni, and Gerard Tripot. A data-driven failure prognostics method based on mixture of gaussians hidden markov models. *IEEE Transactions on reliability*, 61(2):491–503, 2012.

[ULRS11] Paolo Umiliacchi, David Lane, Felice Romano, and A SpA. Predictive maintenance of railway subsystems using an ontology based modelling approach. In *Proceedings of 9th world conference on railway research*, pages 22–26, 2011.

[YDH08] Bo-Suk Yang, Xiao Di, and Tian Han. Random forests classifier for machine fault diagnosis. *Journal of Mechanical Science and Technology*, 22(9):1716–1725, Sep 2008.

[YL99] S.K. Yang and T.S. Liu. State estimation for predictive maintenance using kalman filter. In *Reliability Engineering and System Safety*, volume 66 (1), pages 29–39. Elsevier, 1999.

[YWL16] M. Yuan, Y. Wu, and L. Lin. Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In *IEEE International Conference on Aircraft Utility Systems (AUS)*, pages 135–140, Oct 2016.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub

[ZRFG17] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta. Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 88–95, June 2017.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.